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. It 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.

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
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. Dansereau 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, $J$ – to map camera spatial velocity to the optical flow of points in the scene

$$\dot{p} = J(p, P; K)\nu,$$  \hspace{1cm} (1)

where $cP \in \mathbb{R}^3$ is the coordinate of a world point in the camera reference frame, $p \in \mathbb{R}^2$ is its image plane projection, $K \in \mathbb{R}^{3 \times 3}$ is the camera intrinsic matrix, $\nu = [v; \omega] \in \mathbb{R}^6$ is the camera’s spatial velocity in the camera reference frame, which is the concatenation of the camera’s translational velocity $v = [v_x, v_y, v_z]^T$ and rotational velocity $\omega = [\omega_x, \omega_y, \omega_z]^T$ in the camera reference frame.

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

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

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

$$J(p, cP; K)\nu = \lambda(p^* - p^\#),$$  \hspace{1cm} (2)

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

$$\begin{bmatrix}
J(p_1, cP_1; K) & \nu & = & \lambda (p_1^* - p_{1^\#}) \\
\vdots & \vdots & & \vdots \\
J(p_N, cP_N; K) & \nu & = & \lambda (p_N^* - p_{N^\#})
\end{bmatrix}$$ \hspace{1cm} (3)

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

$$\nu = -\lambda \begin{bmatrix} J_1 & \vdots & J_N \end{bmatrix}^+ \begin{bmatrix} p_1^* - p_{1^\#} \\
\vdots \\
p_N^* - p_{N^\#} \end{bmatrix},$$ \hspace{1cm} (4)

where $J^+$ represents the pseudo-inverse of $J$. 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 $\phi = [s, t, u, v, 1]^T$ 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, $cP = [P_x, P_y, P_z]^T$ follow a pair of linear relationships [11], [12], as shown in Fig. 3

$$\begin{bmatrix} u \\ v \end{bmatrix} = \left( \frac{D}{P_z} \right) \begin{bmatrix} P_x - s \\ P_y - t \end{bmatrix},$$ \hspace{1cm} (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 $\mathcal{L}(s, t, u, v) \in \mathbb{R}^2$. 

$$\begin{bmatrix} u \\ v \end{bmatrix} = \left( \frac{D}{P_z} \right) \begin{bmatrix} P_x - s \\ P_y - t \end{bmatrix},$$ \hspace{1cm} (5)
TABLE I

COMPARISON OF CAMERA SYSTEMS’ CAPABILITIES AND TOLERANCES FOR VISUAL SERVOING

<table>
<thead>
<tr>
<th>Systems</th>
<th>Perspectives</th>
<th>Field of View</th>
<th>Baseline</th>
<th>Baseline Direction</th>
<th>Aperture Problem</th>
<th>Occlusion Tolerance</th>
<th>Specular Tolerance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional Cameras</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mono</td>
<td>1</td>
<td>wide</td>
<td>zero</td>
<td>none</td>
<td>significant</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Stereo</td>
<td>2</td>
<td>wide</td>
<td>wide</td>
<td>single</td>
<td>moderate</td>
<td>weak</td>
<td>no</td>
</tr>
<tr>
<td>Trinocular</td>
<td>3</td>
<td>wide</td>
<td>wide</td>
<td>three</td>
<td>moderate</td>
<td>moderate</td>
<td>no</td>
</tr>
<tr>
<td>Multiple cameras</td>
<td>n</td>
<td>wide</td>
<td>wide</td>
<td>multiple</td>
<td>minor</td>
<td>moderate</td>
<td>no</td>
</tr>
<tr>
<td>Light Field Cameras</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Array</td>
<td>(n^2)</td>
<td>wide</td>
<td>wide</td>
<td>multiple</td>
<td>minor</td>
<td>strong</td>
<td>yes</td>
</tr>
<tr>
<td>MLA (^a)</td>
<td>(n^2)</td>
<td>wide</td>
<td>narrow</td>
<td>multiple</td>
<td>minor</td>
<td>strong</td>
<td>yes</td>
</tr>
<tr>
<td>MirrorCam (^b)</td>
<td>(n^2)</td>
<td>narrow</td>
<td>wide</td>
<td>multiple</td>
<td>minor</td>
<td>strong</td>
<td>yes</td>
</tr>
</tbody>
</table>

\(^a\) Based on \(n^2\) pixels per lenslet

\(^b\) Based on \(n^2\) mirrors

---

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

\[
\begin{bmatrix}
  u_0 \\
  v_0 \\
  w
\end{bmatrix}
= \begin{bmatrix}
  u \\
  v
\end{bmatrix}_{s,t=0}
= \left(\frac{D}{P_z}\right)
\begin{bmatrix}
  P_x \\
  P_y
\end{bmatrix}.
\]  

(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 \(cP\). 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

\[
\begin{align*}
  \dot{u}_0 &= \frac{D(\dot{P}_y P_z - P_z \dot{P}_y) / P_z^2}{P_z}, \\
  \dot{v}_0 &= \frac{D(\dot{P}_x P_z - P_z \dot{P}_x) / P_z^2}{P_z}, \\
  \dot{w} &= \frac{D \dot{P}_z}{P_z^2},
\end{align*}
\]  

(9)  

(10)  

(11)

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 \(W\) through a continuous-domain image Jacobian

\[
\dot{W} = J_{W} \nu.
\]  

(8)

Differentiation of (6) and (7) yields

\[
\begin{align*}
  \dot{u}_0 &= \frac{D(\dot{P}_y P_z - P_z \dot{P}_y) / P_z^2}{P_z}, \\
  \dot{v}_0 &= \frac{D(\dot{P}_x P_z - P_z \dot{P}_x) / P_z^2}{P_z}, \\
  \dot{w} &= \frac{D \dot{P}_z}{P_z^2},
\end{align*}
\]  

(9)  

(10)  

(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
\[
\dot{P} = -(\omega \times P) - v,
\]
yielding three components \( \dot{P} \) expressed in terms of \( \dot{P} \) and \( v \). Substituting these expressions into (9)–(11) allows us to factor out the continuous-domain Jacobian
\[
J_c = \begin{bmatrix}
0 & -w_0 & w_0 & 0 & -D_2 & -D - \frac{1}{2}
0 & w_0 & -D + \frac{1}{2} & 0 & -\frac{1}{2} - w_0 & -v_0
0 & 0 & -w_0 & w_0 & 0 & 0
\end{bmatrix}.
\]

While conventional image Jacobians require an estimate of depth, we note that \( J_c \) 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 Jacobians would have similar depth dependencies to \( J_c \). Multiple views make parallax, and thus depth, observable in rotations about the \( x \)- and \( y \)-axes for the LF camera array. Additionally, \( J_c \) 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 \( M \) as the discrete position and slope \( M = \begin{bmatrix} k_0 & l_0 & m_x & m_y \end{bmatrix}^T \), where \( \begin{bmatrix} k_0 & l_0 \end{bmatrix} \) 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 \( H \) relating a ray \( \phi \) to the corresponding sample in the LF \( n = [i, j, k, l]^T \) as in
\[
\phi = Hn,
\]
where in general \( H \) is of the form
\[
H = \begin{bmatrix}
h_{11} & 0 & h_{13} & 0 & h_{15}
0 & h_{21} & 0 & h_{23} & h_{25}
h_{31} & 0 & h_{33} & 0 & h_{35}
h_{32} & h_{42} & 0 & h_{44} & h_{45}
0 & 0 & 0 & 1 & 1
\end{bmatrix},
\]
and the matrix \( H \) 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 \( h_{13} \) 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 \( k \), \( h_{15} = -h_{11}(N_k/2 + 1/2) \) 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
\[
m_x = \frac{h_{11}h_{44}}{h_{22}h_{33}}m_y.
\]

We wish to express the image Jacobian of (8) in the discrete domain,
\[
M = [k_0, l_0, m_x]^T = J_o \nu,
\]
where the observation is expressed relative to the LF centre, \( k_0 = k_0 + h_{35}/h_{33}, l_0 = l_0 + h_{45}/h_{44} \).

From (14), we can relate the discrete and continuous-domain observations as
\[
u_0 = h_{33}k_0, \quad v_0 = h_{44}l_0, \quad w = \frac{h_{33}}{h_{11}} m_x = \frac{h_{44}}{h_{22}} m_y,
\]
from which it is trivial to express the derivatives of the discrete observation in terms of the continuous variables:
\[
\begin{align*}
\dot{k}_0 &= h_{33}\dot{u}_0, \quad \dot{l}_0 = h_{44}\dot{v}_0, \quad \dot{m}_x = \frac{h_{11}}{h_{33}} \dot{w}, \quad \dot{m}_y = \frac{h_{22}}{h_{44}} \dot{w}.
\end{align*}
\]

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
\[
J_o = \begin{bmatrix}
m_{x1} & 0 & h_{33} & h_{44} & h_{35} & h_{45}
m_{y22} & h_{33} & h_{44} & h_{33} & 0 & 0
0 & h_{44} & h_{44} & 0 & 0 & h_{45}
0 & 0 & h_{33} & 0 & 0 & h_{35}
\end{bmatrix}.
\]

### 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 $N_{\text{min}}$ 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 $N_{\text{min}}$ images is accepted as a 4D feature, and a mean slope estimate is formed based on all passing sub-images. $N_{\text{min}}$ 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 $\nu$, which is subsequently multiplied by a gain $\lambda$, as in (4).

Although velocity control is formulated in (4), the un-optimized 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 $\nu$ into a homogeneous transform $\bar{\tau}$ 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 image features match.

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.

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 $M$, 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 $\lambda$ 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 $3 \times 3$ 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 $\lambda$ 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.
For the experiments, we first moved the robot arm and camera were controlled using the architecture outlined in Fig. 4. For the experiments, we first moved the robot arm to the end of a Kinova MICO arm robot, shown in Fig. 1a. The robot arm is moved forwards to reach the goal view. Motion towards the goal, the camera retreats backwards, before retreat issues. Instead of taking the straight-forward screw regular IBVS, this formulation of LF-IBVS exhibited camera motion profiles relative to the goal, which converge to zero, and the camera velocity profiles, which converge to zero, and the image Jacobian condition number. Note the LF-IBVS outperforms 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.

Fig. 6a shows the view of the central camera, and the image feature paths as the camera array servo 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 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.
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 $\lambda$ 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, 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...
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


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.