International Conference on Humanoid Robots (HUMANOIDS), 2014

Online Multi-Camera Registration for Bimanual Workspace Trajectories

Neil T. Dantam Heni Ben Amor Henrik I. Christensen Mike Stilman

***Abstract*— We demonstrate that millimeter-level bimanual manipulation accuracy can be achieved without the static camera registration typically required for visual servoing. We register multiple cameras online, converging in seconds, by visually tracking features on the robot hands and filtering the result. Then, we compute and track continuous-velocity relative workspace trajectories for the end-effector. We demonstrate the approach using Schunk LWA4 and SDH manipulators and Logitech C920 cameras, showing accurate relative positioning for pen-capping and object hand-off tasks. Our filtering software is available under a permissive license.**[1]

I. Introduction

Visual feedback of hand movements provides rich information that can be used to correct for errors and improve manipulation accuracy. Recent evidence suggests that humans use visual feedback of the hand to guide reach and grasp tasks [15]. Continuously tracking and monitoring the state of the hand allows us to dynamically accommodate to internal and external perturbations, e.g., muscle impairments, thereby achieving a high degree of robustness during manipulation tasks.

In robotics, using visual feedback depends on a kinematic registration between the camera and the manipulators. Typically, this is viewed as a static task: registration is computed offline and assumed to be constant. In reality, camera registration changes during operation due to external perturbations, wear and tear, or even human repositioning. For example, during the recent DARPA Robotics Challenge trials, impacts from falls resulted in camera issues which significantly affected the robot behavior [10]. The pose registration process should be treated as a dynamic task in which the involved parameters are continuously updated. Such an *online* approach to pose registration is challenging, since it requires the constant visibility of a calibration reference and sufficient accuracy to perform manipulation tasks.

Bimanual manipulation requires accurate coordination of both end-effectors. To perform smooth and accurate bimanual manipulation, we propose an online estimation and control approach that combines (1) visual tracking of the manipulators, (2) co-estimation of poses for cameras and end-effectors using a special Euclidean group median and extended Kalman filter, and (3) continuous geometric interpolation on the special Euclidean group. Our key insight is to combine perception and control online, using the robot body frame as a reference. This work extends the single-camera and manipulator registration presented in [5] to multi-camera and multi-manipulator estimation, and it integrates the spherical blending approach of [6] to enable continuous motion of the manipulator in the workspace.

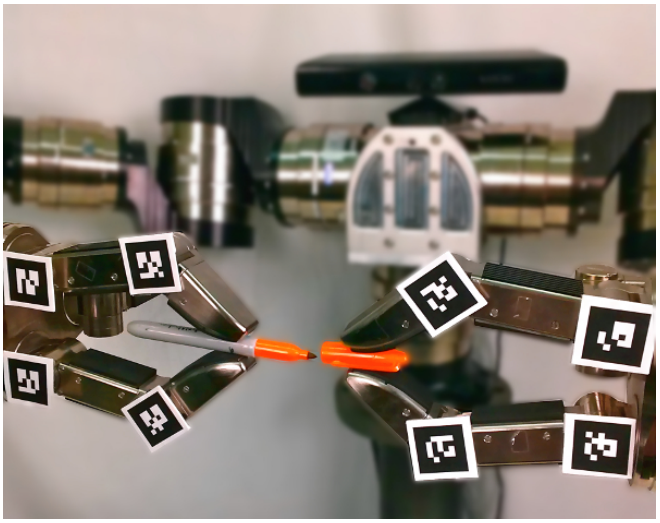

Fig. 1. Bimanual Schunk LWA4/SDH capping a pen using visual feedback from online camera registration and end-effector tracking.

II. Related Work

Typical camera registration methods collect a set of calibration data using an external reference object, compute the calibration, then proceed assuming the calibration is static. OpenCV determines camera registration from point correspondences, typically using a chessboard [12]. Pradeep, et. al, develop a camera and arm calibration approach based on bundle adjustment and demonstrate it on the PR2 robot [13]. This approach requires approximately 20 minutes to collect data and another 20 minutes for computation, a challenge for handling changing pose online.

Visual servo control incorporates camera feedback into robot motion control [1], [2]. The two main types of visual servoing are image-based visual servo control (IBVS), which operates on features in the 2D image, and position-based visual servo control, which operates on 3D parameters. Both of these methods assume a given camera registration. While IBVS is locally stable with regard to pose errors, under PBVS, even small pose errors can result in large tracking error [1]. Compared to both IBVS and PVBS, our method requires no initial camera registration, instead estimating the registration online. Additionally, compared to IBVS, we estimate the full kinematics of the camera and robot,

The authors are with the Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30332, USA. ntd@gatech.edu, hbenamor@cc.gatech.edu, hic@cc.gatech.edu

[1]software available at http://github.com/golems/reflex

and thus can directly follow workspace reference poses and trajectories, such as [6], rather than being limited to image-space reference points and trajectories.

Other recent work has explored online visual parameter identification. A single-camera and single-arm version of the approach in this paper was presented in [5]. [11] tracks a robot arm to identify encoder offsets. This method assumes a given camera registration, but is also tolerant to some registration error. In contrast, our work identifies the camera registration online. Though we do not explicitly consider encoder offsets, our method is empirically robust to offsets of even 30° (see Sect. V). [9] considers bimanual arm and object tracking with vision and tactile feedback. Though the hardware and implementation differ from work presented in this paper, we obtain similar accuracy using inexpensive webcams and without tactile sensing. [16] uses maps generated from a Simultaneous Localization and Mapping (SLAM) algorithm to calibrate a depth sensor. In our approach, unlike typical environments for SLAM, the object to which we are trying to register our camera – the manipulator – will necessarily be in motion.

III. Estimation: Poses of Cameras and Hands

We estimate poses of multiple cameras and manipulators by visually detecting the 3D pose of each manipulator. First, we detect texture features on the end-effector and fit a transform, providing an instantaneous estimate of camera and hand pose. To obtain sufficient accuracy for manipulation we then combine median and extended Kalman filtering of these poses.

To use the robot body as a reference for camera registration, we track the 3D pose of features on the end-effector. These 3D poses can be estimated with marker-based [14] and model-based approaches [3]. Marker-based approaches require attaching fiducials to known locations on the robot, such as the fingers. Model-based tracking, on the other hand, requires accurate polygon meshes of the tracked object. In our implementation, we use the ALVAR library [14] for marker-based tracking.

For computational reasons, we used the *dual quaternion* representation for the special Euclidean group $\mathcal{SE}(3)$. Compared to matrices, the dual quaternion has lower dimensionality and is more easily normalized, both advantages for our filtering implementation. Compared to the unit quaternion plus translation vector representation, dual quaternions are more convenient for algebraic manipulation because they are chained through multiplication. For Euclidean transformations, we use the conventional coordinate notation where the leading superscript denotes the parent frame and the following subscript denotes the child frame, i.e., ${}^a\mathcal{S}_b$ gives the origin of b relative to a. The transformation ${}^a\mathcal{S}_b$ followed by ${}^b\mathcal{S}_c$ is given as the dual quaternion multiplication ${}^a\mathcal{S}_b \otimes {}^b\mathcal{S}_c = {}^a\mathcal{S}_c$. We represent an orientation quaternion as ${}^a q_b$, a translation vector as ${}^a x_b$, a rotational velocity as ${}^a\omega_b$, and the combined translational and rotational velocity as ${}^a\dot{\chi}_b$.

A. Asynchronous Pose Co-Estimation

Each camera image provides pose measurements for visible end-effector features. To reduce estimation latency, we process and filter the measurements from each camera asynchronously as they arrive rather than collecting images from all cameras at a fixed timestep.

The kinematic chain through the manipulator, feature, and camera is defined as:

$$ {}^b\mathcal{S}_{w_i} \otimes {}^{w_i}\mathcal{S}_{w'_i} \otimes {}^{w'_i}\mathcal{S}_{f_p} = {}^b\mathcal{S}_{c_j} \otimes {}^{c_j}\mathcal{S}_{f_p} \tag{1} $$

where ${}^b\mathcal{S}_{w_i}$ is the encoder-measured pose of wrist i in body frame, ${}^{w_i}\mathcal{S}_{w'_i}$ is the estimated offset pose of wrist i, ${}^{w'_i}\mathcal{S}_{f_p}$ is the encoder-measured transform from wrist i to feature p on the hand, ${}^b\mathcal{S}_{c_j}$ is the estimated pose registration of camera j, and ${}^{c_j}\mathcal{S}_{f_p}$ is the visually-measured pose feature p in camera j. For a depiction of the setup see Fig. 3.

Based on (1), we produce measurements for wrist offset ${}^{w_i}\mathcal{S}_{w'_i}$ and camera registration ${}^b\mathcal{S}_{c_j}$:

$$ \widetilde{{}^{w_i}\mathcal{S}_{w'_i}} = ({}^b\mathcal{S}_{w_i})^{-1} \otimes \widehat{{}^b\mathcal{S}_{c_j}} \otimes {}^{c_j}\mathcal{S}_{f_p} \otimes ({}^{w'_i}\mathcal{S}_{f_p})^{-1} \tag{2} $$

$$ \widetilde{{}^b\mathcal{S}_{c_j}} = {}^b\mathcal{S}_{w_i} \otimes \widehat{{}^{w_i}\mathcal{S}_{w'_i}} \otimes {}^{w'_i}\mathcal{S}_{f_p} \otimes ({}^{c_j}\mathcal{S}_{f_p})^{-1} \tag{3} $$

where $\widetilde{{}^{w_i}\mathcal{S}_{w'_i}}$ is the wrist offset measurement from this image and feature, $\widetilde{{}^b\mathcal{S}_{c_j}}$ is the camera registration measurement, $\widehat{{}^{w_i}\mathcal{S}_{w'_i}}$ is the currently estimated wrist offset, and $\widehat{{}^b\mathcal{S}_{c_j}}$ is the currently estimated camera pose.

B. $\mathcal{SE}(3)$ Median and Extended Kalman Filter

We apply median and extended Kalman filtering in the special Euclidean group $\mathcal{SE}(3)$ to the measurements for wrist offset $\widetilde{{}^{w_i}\mathcal{S}_{w'_i}}$ and camera registration $\widetilde{{}^b\mathcal{S}_{c_j}}$, similar to the approach in [5]. First, to reject outliers, we compute the median measurement over a sliding time window. Then, we use an extended Kalman Filter over time to compute optimal pose estimates under a Gaussian noise assumption.

To compute the median of orientation $\overline{q}$ over the sliding window, the structure of rotations in $\mathcal{SO}(3)$ offers a convenient distance metric between two orientations: the angle between them. Using this geometric interpretation, the median orientation $\overline{q}$ is the orientation with minimum angular distance to all other orientations.

$$ \overline{q} = \underset{q_i \in Q}{\arg\min} \sum_{j=0}^{n} |\ln(q_i^* \otimes q_j)| \tag{4} $$

The median translation $\overline{v}$ is the conventional geometric median, the translation with minimum Euclidean distance to all other translations:

$$ \overline{v} = \underset{v_i \in V}{\arg\min} \sum_{j=0}^{n} |v_i - v_j| \tag{5} $$

In the extended Kalman filter, we consider state x composed of a quaternion q, a translation vector v, and the translational and rotational velocities, $\dot{v}$ and ω:

$$ x = (q, v) = [q_x, q_y, q_z, q_w, v_x, v_y, v_z, \omega_x, \omega_y, \omega_z, \dot{v}_x, \dot{v}_y, \dot{v}_z] $$

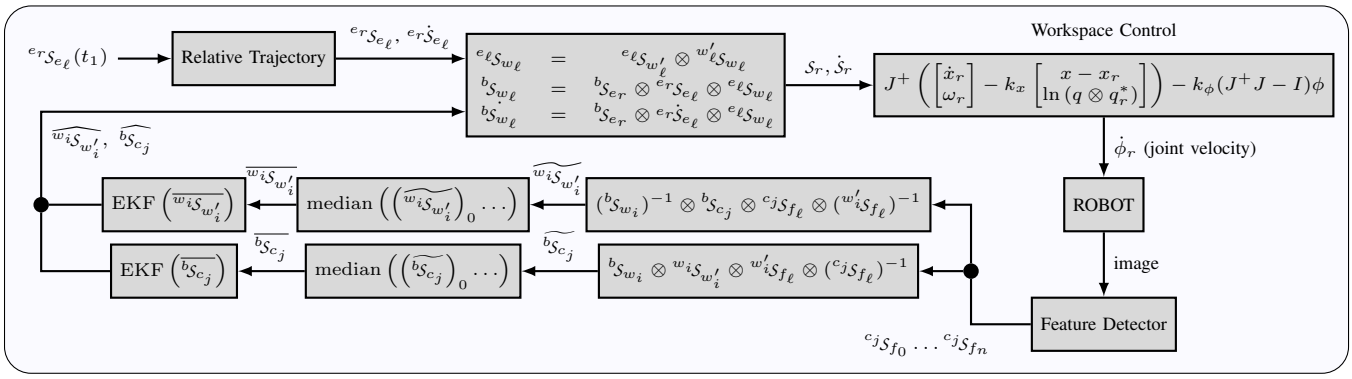

Fig. 2. Block diagram of the control system. 3D feature poses ${}^{c_j}\mathcal{S}_{f_p}$ are detected from visual data. Instantaneous wrist offsets $\widetilde{{}^{w_i}\mathcal{S}_{w'_i}}$ and camera registrations $\widetilde{{}^{b}\mathcal{S}_{c_j}}$ are computed. Then the median of these poses is taken over a sliding window and subsequently Kalman-filtered. The filtered poses are used to track a relative left-right workspace trajectory, and the Jacobian damped-least squares gives the reference joint velocities $\dot{\phi}_r$.

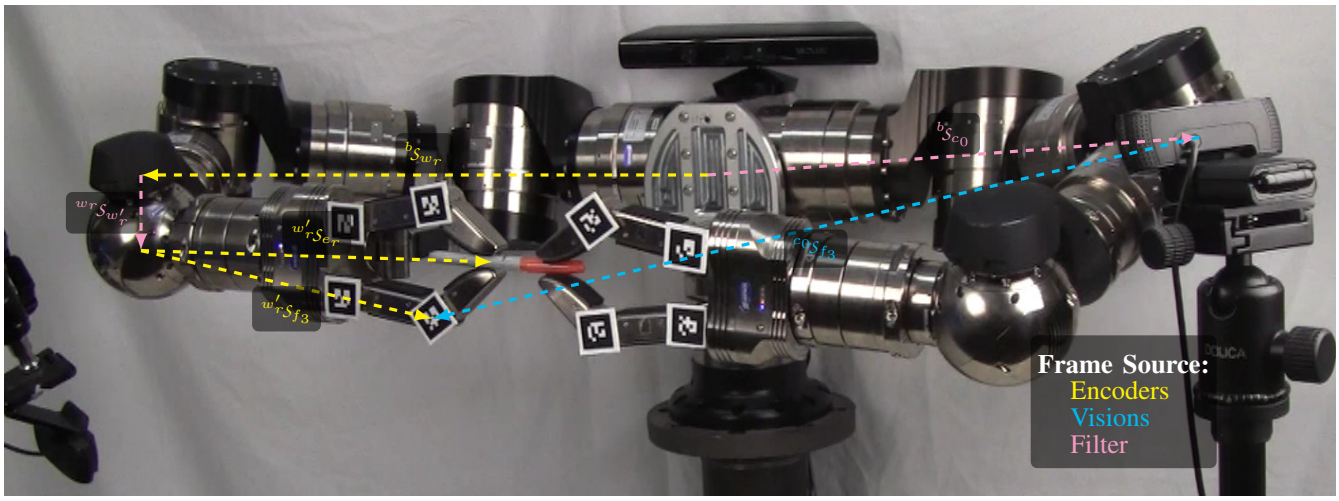

Fig. 3. Setup for a dual-arm and dual-camera system. The kinematic frames are shown for one of the arms and cameras.

The measurement z is the median pose from the sliding window:

$$z = (\overline{q}, \overline{v}) = [q_x, q_y, q_z, q_w, v_x, v_y, v_z]$$

The general EKF prediction step for time k is:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1}) \tag{6}$$

$$F_{k-1} = \left.\frac{\partial f}{\partial x}\right|_{\hat{x}_{k-1|k-1}} \tag{7}$$

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q_{k-1} \tag{8}$$

where $\hat{x}$ is the estimated state, $f(x)$ is the process model, F is the Jacobian of f, P is the state covariance matrix, and Q is the process noise model.

The process model integrates the translational and rotational velocity, staying in the $\mathcal{SE}(3)$ manifold using the dual quaternion exponential of the twist Ω:

$$\Omega(\omega, \dot{v}, v) = \Big(\omega,\ v \times \omega + \dot{v}\Big)$$

$$f(x) = \exp\left(\frac{\Delta t}{2}\Omega\right) \otimes (q,\ v) \tag{9}$$

Now, we find the process Jacobian F. The translation portion is a diagonal matrix of the translational velocity. For the orientation portion, we find the quaternion derivative $\dot{q}$ from the rotational velocity:

$$\dot{q} = \frac{1}{2}\omega \otimes q \tag{10}$$

This quaternion multiplication can be converted into the following matrix multiplication:

$$\frac{1}{2}\omega \otimes q = \frac{1}{2} M_r(q)\, \omega$$

$$M_r(q) = \begin{bmatrix} q_w & q_z & -q_y \\ -q_z & q_w & q_x \\ q_y & -q_x & q_w \\ -q_x & -q_y & -q_z \end{bmatrix} \tag{11}$$

Note that we omit the w column of the typical quaternion multiplication matrix because the w element of rotational velocity ω is zero.

This gives the following process 13×13 Jacobian F:

$$F = \begin{bmatrix} I_{4\times4} & 0 & \frac{1}{2}\Delta t M_r\left(q\right) & 0 \\ 0 & I_{3\times3} & 0 & \Delta t I_{3\times3} \\ 0 & 0 & I_{3\times3} & 0 \\ 0 & 0 & 0 & I_{3\times3} \end{bmatrix} \tag{12}$$

Now we consider the EKF correction step. The general form is:

$$\hat{z}_k = h(\hat{x}_{k|k-1}) \tag{13}$$
$$H_k = \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_{k|k-1}} \tag{14}$$
$$y_k = v(z_k, \hat{z}) \tag{15}$$
$$S_k = H_k P_{k|k-1} H_k^T + R_k \tag{16}$$
$$H_k P_{k|k-1} = S_k K_k^T \tag{17}$$
$$\hat{x}_{k|k} = p(\hat{x}_{k|k-1}, K_k y_k) \tag{18}$$
$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{19}$$

where z is the measurement, h is the measurement model, H is the Jacobian of h, $\hat{z}$ is the estimated measurement, R is the measurement noise model, and K is the Kalman gain, v is a function to compute measurement residual, and p is a function to compute the state update.

We compute the EKF residuals and state updates using relative quaternions to remain in $\mathcal{SE}(3)$ without needing additional normalization. The observation $h(x)$ is a pose estimate:

$$\begin{aligned} h(x) &= (q, v) \\ H &= I_{7\times7} \end{aligned} \tag{20}$$

We compute the measurement residual based on the relative rotation between the measured and estimated pose:

$$\begin{aligned} v(z, \hat{z}) &= (y_q, y_v) \\ y_q &= \ln\left(z_q \otimes \hat{z}_q^*\right) \otimes q \\ y_v &= z_v - \hat{z}_v \end{aligned} \tag{21}$$

where y_q is the orientation part of the residual and y_v the translation part. Note that $\ln\left(z_q \otimes \hat{z}_q^*\right)$ corresponds to a velocity in the direction of the relative transform between the actual and expected pose measurement and that we can consider y_q as a quaternion derivative. Then, the update function will integrate the pose portion of y, again using the exponential of the twist. First, we find the twist corresponding to the product of the Kalman gain K and the measurement residual y:

$$\begin{aligned} (Ky)_\phi &= (Ky)_q \otimes q^* \\ \Omega(Ky, v) &= ((Ky)_\phi, v \times (Ky)_\phi + (Ky)_v) \end{aligned} \tag{22}$$

Then, we integrate estimated pose using the exponential of this twist:

$$(x_{(q,v)})_{k|k} = \exp\left(\frac{\Delta t}{2}\Omega\right) \otimes (q, v) \tag{23}$$

Finally, the velocity component of innovation y is scaled and added:

$$(x_{\omega,\dot{v}})_{k|k} = x_{\omega,\dot{v}} + (Ky)_{\omega,\dot{v}} \tag{24}$$

IV. Control: Continuous Workspace Trajectories

To perform smooth, bimanual motion, we compute a relative workspace trajectory between the two manipulators, transform the relative pose and velocity of the trajectory to the body frame, then compute joint velocities using the Jacobian damped least squares pseudoinverse.

We compute a relative trajectory for the two end-effectors using the spherical parabolic blends described in [6]. This provides a straight-line, constant-axis, and continuous-velocity workspace path for the end-effector by blending subsequent spherical linear interpolation segments. Given a list of relative left-right waypoint poses and times, ${}^{e_r}\mathcal{S}_{e_\ell}(t_0), \ldots, {}^{e_r}\mathcal{S}_{e_\ell}(t_n)$, we compute the reference left-right pose and velocity as a function of time: ${}^{e_r}\mathcal{S}_{e_\ell}(t)$, ${}^{e_r}\dot{\chi}_{e_\ell}(t)$.

From the relative reference pose ${}^{e_r}\mathcal{S}_{e_\ell}$ and velocity ${}^{e_r}\dot{\chi}_{e_l}$ between the left and right end-effectors, we control the left arm in workspace, by first converting the relative pose and velocity to the body frame, then computing the Jacobian damped-least-squares inverse kinematics solution.

The left-arm wrist pose ${}^{b}\mathcal{S}_{w_\ell}$ follows directly from the kinematic chain through the right arm:

$$\begin{aligned} {}^{b}\mathcal{S}_{w_\ell} &= {}^{b}\mathcal{S}_{e_r} \otimes {}^{e_r}\mathcal{S}_{e_\ell} \otimes {}^{e_\ell}\mathcal{S}_{w_\ell} \\ {}^{e_\ell}\mathcal{S}_{w_\ell} &= {}^{e_\ell}\mathcal{S}_{w'_\ell} \otimes {}^{w'_\ell}\mathcal{S}_{w_\ell} \end{aligned} \tag{25}$$

Next, we compute the body-frame feedforward reference velocity, ${}^{a}\dot{\chi}_{b}$. Since there is only one changing frame, ${}^{e_r}\mathcal{S}_{e_\ell}$, we could find the corresponding body frame motion by rotating the velocity. However, the typical computation is notationally cumbersome [4, p140].[2] Instead, we find an elegant and more general solution by merely taking the derivative of the pose:

$$\begin{aligned} & {}^{b}\mathcal{S}_{w_\ell} = {}^{b}\mathcal{S}_{e_r} \otimes {}^{e_r}\mathcal{S}_{e_\ell} \otimes {}^{e_\ell}\mathcal{S}_{w_\ell} \\ \Rightarrow\ & {}^{b}\dot{\mathcal{S}}_{w_\ell} = \cancelto{0}{{}^{b}\dot{\mathcal{S}}_{e_r}} \otimes ({}^{e_r}\mathcal{S}_{e_\ell} \otimes {}^{e_\ell}\mathcal{S}_{w_\ell}) + {}^{b}\mathcal{S}_{e_r} \otimes \frac{d}{dt}\left({}^{e_r}\mathcal{S}_{e_\ell} \otimes {}^{e_\ell}\mathcal{S}_{w_\ell}\right) \\ \Rightarrow\ & {}^{b}\dot{\mathcal{S}}_{w_\ell} = {}^{b}\mathcal{S}_{e_r} \otimes \left({}^{e_r}\mathcal{S}_{e_\ell} \otimes \cancelto{0}{{}^{e_\ell}\dot{\mathcal{S}}_{w_\ell}} + {}^{e_r}\dot{\mathcal{S}}_{e_\ell} \otimes {}^{e_\ell}\mathcal{S}_{w_\ell}\right) \\ \Rightarrow\ & {}^{b}\dot{\mathcal{S}}_{w_\ell} = {}^{b}\mathcal{S}_{e_r} \otimes {}^{e_r}\dot{\mathcal{S}}_{e_\ell} \otimes {}^{e_\ell}\mathcal{S}_{w_\ell} \end{aligned} \tag{26}$$

where $\cancelto{0}{\mathcal{S}}$ indicates that S cancels to zero, and we assume the right arm and left fingers are stationary ($0 = {}^{b}\dot{\mathcal{S}}_{e_r} = {}^{e_\ell}\dot{\mathcal{S}}_{w_\ell}$). Relative motion with both arms moving could be computed by including the nonzero derivative ${}^{b}\dot{\mathcal{S}}_{e_r}$ in the computation. We can use the product rule for this derivation because dual quaternion poses are chained through multiplication. Using the quaternion plus translation vector representation,

[2] The complexity of the velocity transformation notation in [4, p140] stems from its representation using Gibbs's vector calculus which decouples the quaternion multiplication into separate dot and cross products. Hamilton's and Study's classical quaternion and dual quaternion notation is simpler and more elegant for this kinematic computation. A similar computation is also possible using transformation matrices and their derivatives, but these matrices are more difficult to normalize than quaternions, increasing numerical error.

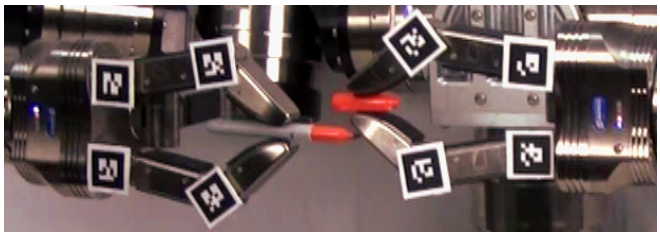

Fig. 4. Manipulation error using only encoders for position feedback. Without using visual feedback, there is a 15mm relative positioning error between the two end-effectors.

chaining is not a multiplication, so an equivalent derivation would be more complex.

Velocity and the dual quaternion derivative are related as follows:

$$\frac{d\mathbb{R}(S)}{dt} = \frac{1}{2}\omega \otimes \mathbb{R}(S)$$
$$\frac{d\mathbb{D}(S)}{dt} = \frac{1}{2}\left(\dot{x} \otimes \mathbb{R}(S) + x \otimes \frac{d\mathbb{R}(S)}{dt}\right) \quad (27)$$

where $\mathbb{R}(S)$ is the real part of S, $\mathbb{D}(S)$ is the dual part of S, ω is rotational velocity, and x is translation.

Finally, we compute reference joint velocities using the Jacobian damped least squares with a nullspace projection to keep joints near the zero position:

$$\dot{\phi}_r = J^+\left(\begin{bmatrix}\dot{x}_r \\ \omega_r\end{bmatrix} - k_x \begin{bmatrix}x - x_r \\ \ln\left(q \otimes q_r^*\right)\end{bmatrix}\right) - k_\phi(J^+J - I)\phi \quad (28)$$

where x is the actual translation, q is the actual orientation quaternion, x_r is the reference translation, q_r is the reference orientation quaternion, ω is the actual rotational velocity, ω_r is the reference rotational velocity, k_x is the workspace position error gain, k_ϕ is the null-space projection gain, and ϕ is the configuration. We then use joint-level velocity control to track the reference joint velocities $\dot{\phi}_r$. A block diagram depicting the components of the control system and their interplay can be found in Fig. 2.

V. Experiments

We implement this approach on a pair of Schunk LWA4 manipulators with SDH end-effectors, and use a pair of Logitech C920 webcams to track the robot and objects. Our estimation and control software is implemented as a distributed system using the Ach real-time communication library [7]. The Schunk LWA4 has seven degrees of freedom and uses harmonic drives, which enable repeatable positioning *precision* of ± 0.15mm [8]. However, absolute positioning *accuracy* is subject to encoder offset calibration and link rigidity. In practice, we achieve ± 15mm accuracy when using only the joint encoders for feedback, as can be seen in Fig. 4. The Logitech C920 provides a resolution of 1920x1080 at 15 frames per second. To measure ground-truth distances, we use a ruler and meter-stick.

To test the relative positioning accuracy of our implementation, we servo the end-effectors to a reference zero relative alignment, Fig. 5, and then measure the actual relative error between the two end-effectors. We conduct this test using only encoder feedback, then with visual feedback. We also repeat the test injecting encoder error of 15° at the initial shoulder joint, 30° at the shoulder, and 15° at both the shoulder and elbow. The results of this test are summarized in Table I.

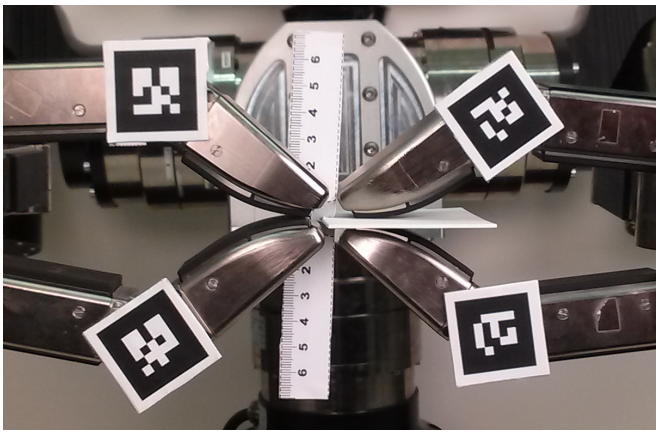

Fig. 5. Testing relative positioning accuracy by aligning the end-effectors. Incorporating visual feedback and online registration reduces manipulation error from 15mm to $\approx$ 2mm.

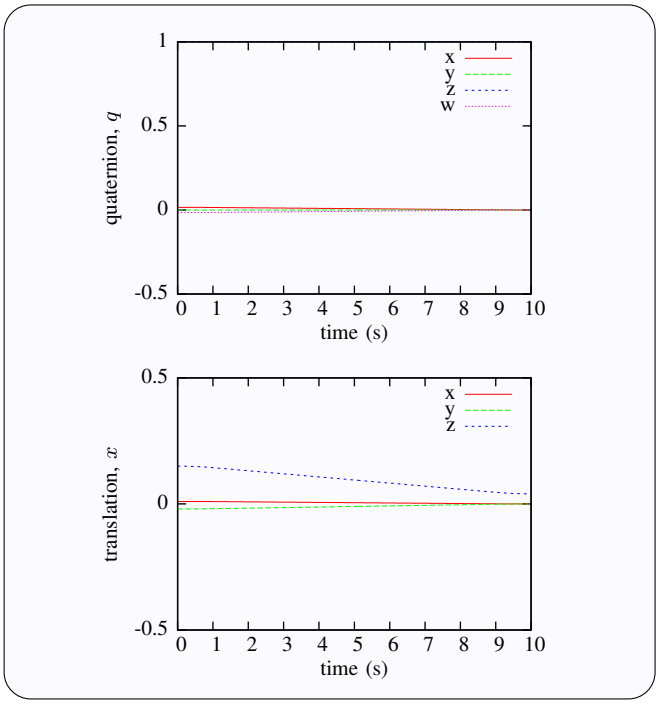

Fig. 6. Relative trajectory of ${}^{e_r}\mathcal{S}_{e_\ell}$ between left and right end-effectors for pen-capping. The trajectory has constant acceleration, constant velocity, and constant deceleration segments.

In addition, we use this method to perform the pen-capping task shown in Fig. 3 and the object hand off task shown in Fig. 7. The relative trajectory of ${}^{e_r}\mathcal{S}_{e_\ell}$ for the pen-capping task is plotted in Fig. 6

VI. Discussion

The results of Sect. V show that this method achieves bimanual positioning accuracy of a few millimeters without static camera registration and even with significant (30°) error in the joint encoders.

TABLE I

POSITIONING TEST RESULTS (mm)

	Mean		Std. Dev.	
	encoder	**visual**	encoder	**visual**
No Offset	16.5	2.2	0.5	0.94
shoulder: 15°	155	2.8	0.6	0.78
shoulder: 30°	280	1.3	0	0.95
shoulder & elbow: 15°	240	0.95	0	1.1

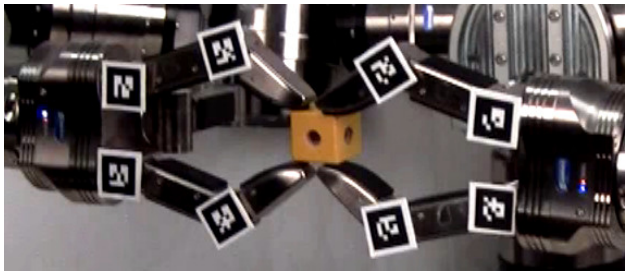

Fig. 7. An object hand-off task.

There are a variety of error sources that we address in this system. For the kinematics, error from encoder offsets in the arm, imprecise link lengths, and flexing of links all contribute inaccurate kinematic pose estimates. For perception, error from inaccurate camera intrinsics, imprecise fiducial sizes, offsets in object models, and noise in the image all contribute to error in visual pose estimates. To achieve accurate manipulation, we must account for these potential sources of error.

The position of the tracked features on the robot has an important effect on error correction. Kinematic errors between the robot body origin and the tracked features, e.g., due to flex or encoder offsets, are incorporated into the camera registration and handled through the servo loop. Error between the observed features and the end-effector cannot be corrected. Thus, it is better to track features as close to the end-effector as possible. Consequently, we placed the fiducial markers on the fingers of the SDH end-effector.

One source of error for manipulation that we do not address is error in grasping. Because we track only the robot hand, any error in the relative pose between the hand and grasped object is not corrected. In reality, when grasping an object, the *object* itself becomes the robot's end-effector. Thus, to accurately manipulate in-hand objects, it would be better to track the objects themselves. Since a grasped object is likely to be partially occluded, model-based tracking such as [3], which is robust to occlusions, is a potential approach.

A crucial additional consideration in manipulation is force and tactile sensing. Using visual feedback without force and tactile sensing already reduces the error to a few millimeters and allows the robot to perform tasks such as pen capping and object hand-off. However, considering the generated contact forces during the manipulation would further improve performance and allow even more accurate operation, in particular during the post-contact phase. This is a key area for improvement in this approach.

VII. CONCLUSION

We have presented an online method to identify multiple camera and manipulator poses and track continuous relative trajectories for bimanual manipulation tasks. This is useful for the typical case where camera registration is not static but changes due to model error, disturbances, or wear and tear. The key point is to track both manipulators, and follow a trajectory based on the visually estimated relative 3D pose between the end-effectors. By combining median and Kalman filtering, we are able to achieve millimeter-level manipulation accuracy. We have shown in our experiments that online registration can be used to improve positioning accuracy during bimanual manipulation tasks where successful operation depends on relative end-effector pose.

This method uses feedback only from joint encoders and visual tracking of the robot hand. Further improvements could be made by including force and tactile sensing and by visually tracking in-hand objects.

REFERENCES

[1] François Chaumette and Seth Hutchinson. Visual servo control, part I: Basic approaches. *Robotics and Automation Magazine*, 13(4):82–90, 2006.

[2] François Chaumette and Seth Hutchinson. Visual servo control, part II: Advanced approaches. *Robotics and Automation Magazine*, 14(1):109–118, 2007.

[3] Changhyun Choi and Henrik I Christensen. Robust 3d visual tracking using particle filtering on the special euclidean group: A combined approach of keypoint and edge features. *The International Journal of Robotics Research*, 31(4):498–519, 2012.

[4] J. Craig. *Introduction to Robotics: Mechanics and Control*. Pearson, 3rd edition, 2005.

[5] N. Dantam, H. Ben Amor, H. Christensen, and M. Stilman. Online camera registration for robot manipulation (presented). In *International Symposium on Experimental Robotics*, 2014.

[6] N. Dantam and M. Stilman. Spherical parabolic blends for robot workspace trajectories (accepted). In *International Conference on Intelligent Robots and Systems*, 2014.

[7] Neil Dantam, Daniel Lofaro, Ayonga Hereid, Paul Oh, Aaron Ames, and Mike Stilman. Multiprocess communication and control software for humanoid robots (accepted). *Robotics and Automation Magazine*, 2014.

[8] Schunk GmbH. Dextrous lightweight arm LWA 4D, technical data. http://mobile.schunk-microsite.com/en/produkte/produkte/dextrous-lightweight-arm-lwa-4d.html.

[9] Paul Hebert, Nicolas Hudson, Jeremy Ma, and Joel W Burdick. Dual arm estimation for coordinated bimanual manipulation. In *Intl. Conf. on Robotics and Automation*, pages 120–125. IEEE, 2013.

[10] Sungmoon Joo and Michael Grey. DRC-Hubo retrospective, January 2014. Personal Communication.

[11] Matthew Klingensmith, Thomas Galluzzo, Christopher Dellin, Moslem Kazemi, J. Andrew (Drew) Bagnell, and Nancy Pollard. Closed-loop servoing using real-time markerless arm tracking. In *Intl. Conf. on Robotics and Automation (Humanoids Workshop)*, May 2013.

[12] *OpenCV API Reference*. http://docs.opencv.org/master/modules/refman.html.

[13] Vijay Pradeep, Kurt Konolige, and Eric Berger. Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach. In *Experimental Robotics*, pages 211–225. Springer, 2014.

[14] Kari Rainio and Alain Boyer. *ALVAR – A Library for Virtual and Augmented Reality User's Manual*. VTT Augmented Reality Team, December 2013.

[15] Jeffrey A. Saunders and David C. Knill. Humans use continuous visual feedback from the hand to control both the direction and distance of pointing movements. *Experimental Brain Research*, 162(4):458–473, 2005.

[16] A. Teichman, S. Miller, and S. Thrun. Unsupervised intrinsic calibration of depth sensors via slam. In *Robotics: Science and Systems (RSS)*, 2013.