Abstract—We present an approach for monocular pose estimation and feedback for multi-robot trajectory control. RGB LEDs are mounted on each robot in asymmetric configurations. The LEDs are detected by a single camera with robustness to a variety of lighting conditions. The centroids of the LEDs are used to estimate robot pose through the P3P algorithm and Gauss-Newton minimization. Subsequently, the pose estimates are inputs into an Extended Kalman Filter (EKF) to localize each robot in the global frame. From the EKF estimate, feedback control is applied. Trajectory results with a two robot team are included.

I. INTRODUCTION

Manually searching rubble for trapped survivors in urban disasters is both extremely challenging and hazardous to human rescuers. Multi-robot teams consisting of small robots show promise in their ability to navigate the narrow spaces found in pancaked buildings [1]. This enables the multi-robot team to perform Simultaneous Localization and Mapping (SLAM) in previously unreachable environments.

Within the context of small robots, heterogeneous multi-robot teams have exhibited distinct advantages over more common homogeneous teams. Specifically, we consider a team consisting of a single computationally capable observer robot equipped with a camera and multiple more mobile but less computationally capable picket robots. Unobservable hazards can be detected through the sacrifice of less expensive and computationally capable picket robots [1]. This low cost and specialization approach enables the picket robots to quickly survey terrain while the observer remains safe and maintains the SLAM problem.

Localization is a central problem in cooperative multi-robot teams. Accurate localization, consisting of pose and orientation, is essential for completing complex tasks effectively and is fundamental to the SLAM problem. RGB LEDs serve as a low cost, weight, and power solution for accurate relative localization in unstructured and variable light environments with a monocular camera [2]. Trajectory control builds on accurate localization to enable coordinated motions. This can be accomplished with simple feedback control, addressing the dead zones of motor commands, and a basic motion model of the robot.

In this paper we present a LED based system that is capable of taking in images from a stationary camera and outputting motor commands while maintaining localization estimates of each picket robot. This approach can be directly extended to heterogeneous multi-robot teams by substituting the stationary camera with a mobile observer robot and by adding an observer state to the EKF.

II. RELATED WORK

Several previous approaches exist for using a monocular camera to do pose estimation in six degrees of freedom (6DOF) for robotic systems. The standard approach demonstrated in [2] [3] uses feature detection, perspective 3 point (P3P), and Gauss Newton minimization pipeline. Infrared and RGB LEDs are a common feature [2] [3] as they are low cost, low power, and easily identifiable in a wide range of lighting conditions. The P3P algorithm [4] [5] along with prior estimates enables the estimation of the 6DOF pose when there is a known correspondences between features. Gauss Newton minimization is able to refine the 6DOF pose estimates with a rigid body assumption [6]. The byproduct of the minimization is a 6DOF pose covariance which is advantageous to filtering based localization approaches. A P3P and RANSAC approach is common for applications involving structure from motion (SfM) or estimating the pose of a camera [7][8]. RANSAC approaches are not necessary because LEDs detection has a low number of false detections and total features.

Expanding on the relative pose estimate with covariance, localization filters have been used to improve state estimates. Generalized to systems with moving relative pose sensors, a distributed EKF approach is introduced in [9] for 3-DOF estimation. Odometry sensors are used to propagate the state of each robot. The EKF update is performed when the robots meet and a relative pose measurement is recorded. Relative pose measurements in an EKF framework are expanded on in [10] for more generalized systems. The constraints of relative pose measurements were relaxed to contain any combination of bearing, distance, or orientation. Similar to EKF based approaches, particle filters have been explored in [11][12][13] and show gains in accuracy at the cost of computational resources. A more recent approach [14] uses range-only sensors with a team of aerial vehicles for simultaneous localization and mapping (SLAM) and builds on the limited sensor approach of [15]. Alternatively, graph based approaches have been used to address the multi-robot localization problem [16][17] as alternatives to filter based approaches.

A variety of approaches exist for trajectory control. A basic controller for a monocular camera system is presented in [18]. Communication latency is important for multi-robot teams with a centralized observer that communicates with several robots. The delay between sensor measurement and motor commands increases with the number of robots due to the growing amount of computation required. A notion of time delay in trajectory control is introduced in [19].
which addresses this problem. Stability and path accuracy is further addressed in [20]. Fuzzy adaptive controllers have demonstrated improvements for wheeled and treaded robots for systems without a slippage model. The issue of dead zones is addressed in [22]. The approach does not require each of the robots’ dead zones to be manually calibrated, which is a significant advantage when considering large teams of robots.

III. METHOD

Our computer vision and control approach assumes a previously calibrated camera, accurate static LED relative locations (from a motion capture system or precise measurements), and a specified trajectory. The overview of our approach is shown in Fig. 1. RGB LED detection is used to find the location of LEDs in the camera image. Correspondence and pose optimization takes the location of the LEDs and estimates the pose in the camera frame for each robot. The EKF tracker fuses commanded motions to produce filtered estimates in the global frame for feedback control and to enhance the robustness of previous steps. Finally, the feedback control is tasked with maintaining accurate trajectories. Each step in the overall system approach is described in the following sections.

A. RGB LED Detection

![Fig. 2: RGB LED Detection overview](image)

The RGB LED detection pipeline is summarized in Fig. 2. To reduce the number of false detections, the camera is set to a low exposure. From the raw RGB image, we generate a grayscale image and extract the saturation which is defined as:

$$\text{saturation} = \max(\text{RGB}) - \min(\text{RGB}) \in [0, 1]$$

A minimum saturation, minimum grayscale value, and maximum grayscale value are used to find colored halos surrounding the bright centers of the LEDs. A Difference of Gaussian (DoG) filter and threshold are applied to find dots in the image. The grayscale image is directly thresholded to find brightness (bright) in the image. The intersection of dots and bright form the cores of the LEDs. Applying connected component analysis (CCA) to the cores defines regions. The use of previous pose estimates is an optional parameter to filter out potentially false detections in the regions step. The regions and colored steps are combined with area, eccentricity and color thresholds to find LEDs of appropriate colors and to estimate the centroids (x,y) in the camera image. LEDs’ colors are unique to each robot which enables the LEDs to be quickly clustered.

B. Correspondence and Pose Optimization

After clustering LEDs for each robot, we use the approach and implementation of [2]. Pose correspondence is computed with the P3P algorithm [4][5] on combinations of three LEDs within each robot’s individual cluster. Previous pose estimates are used to increase the speed of the search and handle potential degenerate cases due to occlusion.

The pose optimization is performed with an exponential map and Gauss-Newton minimization algorithm. The iterative approach minimizes reprojection error and starts with an initial estimate from the P3P algorithm [2]:

$$P^* = \arg \min_P \sum_{\langle l,d \rangle \in C} ||\pi(l, P - d)||^2$$

where l is the set of LED configurations, d is the set of LED detection, C is the LED correspondences, and \(\pi\) projects an LED from \(\mathbb{R}^3\) into \(\mathbb{R}^2\) (camera image).

The pose estimate covariance (\(Q_m\)) is calculated with the Jacobian (\(J\)) from the Gauss-Newton minimization [2]:

$$Q_m = (J^T \Sigma^{-1} J)^{-1}$$

where \(\Sigma = I_{2 \times 2} \text{ pixels}^2\)

The pose estimate with covariance is then passed into the localization filter.

C. EKF Tracker

An Extended Kalman Filter smooths the pose estimates from the camera for each robot. The state of the system is updated after each camera pose estimate and propagates based on the time step between camera pose estimates.

1) State Vector: The state vector for each robot is:

$$x = [B_G q^T, G_p^T, B_v^T]^T$$

where \(B_G q^T\) is the unit quaternion representation of the rotation from the global frame (G) to the body frame (B), \(G_p\) is the position of the body in the global frame and \(B_v\) is the velocity in the body frame.
The corresponding error state for each robot is:

\[ \hat{x} = [\delta \theta^T, \dot{\hat{p}}^T, v^T]^T \]

where \( \delta \theta \) is the minimal representation from the error quaternion \( \delta \hat{q} \approx [\frac{1}{2} \delta \theta^T, 1]^T \). The remaining states use an additive error model (e.g. \( \hat{p} = p - \dot{p} \)).

2) Motor Propagation: The state of the robot is propagated with the commanded linear \((\dot{B}v)\), angular velocity in the body frame \((\dot{B}\omega)\) and the time step \((\Delta t)\). Using Euler integration the discrete time motion model and state transition matrix of the system is:

\[
F = \begin{bmatrix}
\begin{bmatrix}
B \dot{q}_k \\
G \dot{p}_k \\
B \dot{v}_k
\end{bmatrix} & \begin{bmatrix}
\Delta t / 2 \cdot \dot{B} \omega \\
\Delta t \cdot G \dot{p}_{k-1} \\
\Delta t \cdot B \dot{v}_k
\end{bmatrix}
\end{bmatrix}
\begin{bmatrix}
I_{3x3} \\
0_{3x3} \\
I_{3x3}
\end{bmatrix}
\begin{bmatrix}
0_{3x3} \\
0_{3x3} \\
I_{3x3}
\end{bmatrix}
\]

where \( \otimes \) represents quaternion composition, \( q_0 \) is the unit quaternion, and

\[
[\omega \times] = \begin{bmatrix}
0 & -\omega_z & \omega_y \\
\omega_z & 0 & -\omega_x \\
-\omega_y & \omega_x & 0
\end{bmatrix}, \Omega(\omega) = \begin{bmatrix}
-\omega \times & \omega^T \\
-\omega^T & 0
\end{bmatrix}
\]

(1)

The process noise covariance \((Q_p)\) was estimated by driving the robots on a motion capture system across different commanded velocities.

An Unscented Kalman Filter (UKF) or a Particle Filter (PF) could be considered to avoid the Jacobian. Due to the simplicity of the motion model and the direct observation of the pose component of the state, an EKF was chosen.

3) Camera Update: The state is updated after each pose estimate:

\[ z = [B \hat{q}^T, C \hat{p}^T]^T \]

where the transform from the global to camera frame is expressed by:

\[ [C \hat{q}^T, B \hat{p}^T]^T \]

A residual, \( r \), and a measurement Jacobian, \( H \), are required for the EKF update. The relationship between the residual and measurement Jacobian is given by:

\[ r = z - \hat{z} \simeq H \hat{z} + n \]

where \( n \) is noise. In order to calculate a residual, \( r \), an estimate of the camera measurements, \( \hat{z} \), is computed.

With respect to the camera frame, the predicted measurement is:

\[ \hat{z} = \begin{bmatrix}
B \hat{q} \otimes C \hat{q} \\
C \hat{R}(C \hat{p} - G \hat{p}_c)
\end{bmatrix} \]

The corresponding residual and measurement Jacobian are:

\[ r = \begin{bmatrix}
2 \cdot \pi (B \hat{q} \otimes (B \hat{q} \otimes \hat{q})^{-1}) \\
C \hat{R} 0_{3x3} 0_{3x3} 0_{3x3}
\end{bmatrix} \]

where \( \pi \) is defined as \( \pi([q_x, q_y, q_z, \theta]^T)^T = [q_x, q_y, q_z]^T \) and use a small angle approximation for the orientation difference between \( z \) and \( \hat{z} \).

The standard covariance update, Kalman gain, and correction are calculated with the measurement noise \((Q_m)\) from Gauss-Newton minimization. The non-quaternion states use standard additive correction (e.g. \( q_k+1 = q_k + \Delta q \)). Quaternion multiplication is converted to quaternion corrections. Quaternion multiplication is applied to the quaternion states and corrections (e.g. \( q_{k+1}[k+1] = q_{k+1}[k] \otimes q_k \)).

D. Feedback Control

We apply a simple feedback control law to drive each robot through a trajectory consisting of waypoints. Each robot follows a differential drive model consisting of a linear and angular velocity command. These commands serve as inputs to the on-board low-level velocity controllers that drive the robot’s motors.

The angular velocity command \( \omega_c \) is set by a Proportional-Derivative (PD) controller on heading error where the desired heading is the angle from the estimate of the robot’s x,y position \((\hat{p}_x, \hat{p}_y)\) to the target waypoints x,y position \((\hat{p}_x, \hat{p}_y)\). Since the motors have a deadband range in which low commands fail to turn the motors, we add a term based on the sign of the motor command.

\[
\theta_{err} = \theta_{des} - \theta
\]

\[
\theta_{des} = \text{atan2} (\hat{p}_y - \hat{p}_y, \hat{p}_x - \hat{p}_x)
\]

\[
\omega_c = k_p \theta_{err} + k_d \theta + \text{sgn}(k_p \theta_{err} + k_d \theta)
\]

The forwards velocity controller’s command \( v_c \) advances the robot at a nominal rate \( v \) at zero heading error and decreases the desired velocity as magnitude of the heading error increases. This keeps the robot from swerving off course while maintaining a smooth transition from one desired heading to another during waypoint switches.

\[
v_c = v \cdot \max(\cos(\theta_{err}), 0)
\]

The robot’s body-frame lateral and longitudinal distances from the target waypoint are thresholded. Thresholding is more lenient in the lateral direction due to the non-holonomic differential drive constraint. Once both distances are sufficiently close, the waypoint is considered achieved and the robot moves onto the next waypoint in the specified trajectory.
IV. EXPERIMENTAL RESULTS

In this section we present a subset of the total experimental results for the Zumy robotic platform. To see the videos presented in conjunction with the live demo at the poster session, please view: [https://www.youtube.com/channel/UCCt9udU6xNjDMD8C0xLhw](https://www.youtube.com/channel/UCCt9udU6xNjDMD8C0xLhw)

A. Zumy Robotic Platform

To test the algorithmic methods for this work, we applied the monocular trajectory control technique to two low-cost mobile robots. The Zumy robotic platform\(^1\) is a decimeter-scale tracked robot designed to be able to support a full Linux computing system for convenient software development, networking, and accelerated vision processing. The Zumy can carry a Microsoft Lifecam 3000 webcam and an InvenSense MPU-6050 MEMS IMU. It also supports WiFi wireless communication to a host computer. This allows the Robot Operating System (ROS)\(^2\) to transmit control and sensor information between the team of robots. This robot is also designed to be easily built from commercially available off-the-shelf (COTS) parts for a total cost of $350. For the purpose of this project, we fitted each Zumy with an LED “hat” and removed the webcam as shown in Fig. 4.

We used the Microsoft Lifecam 3000, which was removed from the Zumy. At the beginning of each experiment, we calibrate the world frame by setting one robot at the desired origin. We take the inverse transform of this robot relative to the camera as our constant camera transform.

Fig. 4: LED “Hat” Zumy (left side) & Normal Zumy with OptiTrack Markers (right side)

B. RGB LED Detection and Pose Estimation

A sample frame of RGB LED Detection is shown in Fig. 5. The original image is processed to find LEDs. In this frame example, all LEDs are found with no false detections. The pose estimate shows the optimized locations of the LEDs with red circles, the rigid body frame with the blue rectangles, and the rotation coordinate frame with the axes.

C. EKF and Trajectory Control

The results for a single Zumy performing a circle 4 times consecutively is shown in Fig. 6, 7. The circle trajectory was performed on a flat surface. From Fig. 7, it is demonstrated that our EKF reduces uncertainty in the depth estimation as the height (\(p_z\)) is more constant. Improvements in the XY plane would require a more effective motion model and a more rigorous process noise covariance model. Overall, the robot consistently follows the circle trajectory despite limitations in the motion model and deadband of the motors.

Further demonstrations such as the robot following a moving camera (waypoint always set as the origin) and multi-robot trajectory videos can be found on the Youtube Channel linked above.

![Fig. 5: Original image (top left), LED Detection (bottom left), Pose Estimate Zumy-1 (top right), Pose Estimate Zumy-2 (bottom right)](image)

**Fig. 6:** Pose estimate and EKF estimate in the XY plane for a circle trajectory. The circle was performed a total of 4 times by the Zumy.

V. CONCLUSIONS

In this paper, we demonstrated the successful integration of RGB LED detection using a monocular camera, pose

\(^1\)https://wiki.eecs.berkeley.edu/biomimetics/Main/Zumy
\(^2\)www.ros.org
estimation through the P3P algorithm and Gauss-Newton minimization, an EKF to smooth the estimates, and a PD controller on the robots’ heading. System integration, including data transfer and wireless communication, was done entirely using ROS. In our current framework, we are able to communicate at 30Hz and achieve real-time control. One issue that we faced was that if we wanted to either visualize more images on the screen or if we wanted to record bagfiles with multiple image topics, we introduced lag into our system. This lag had a significant negative effect on our detection and pose estimation, which made real-time control infeasible.

Aside from reducing lag in ROS, we can also save computational power in the LED detection. Currently, we downsample and search the entire camera image at each iteration. Instead, we can introduce a region of interest in which the previous LED detections can give us an estimate of image where the next search should be performed. In this way, we eliminate the need to search the full image at each iteration. We also attain better resolution because we no longer need to downsample the image, providing better localization of our robot at further distances from the camera.

From adjusting the parameters of our LED detection to changing our control gains, future performance improvements can be made with general parameter tuning. We can elicit tighter control by implementing better deadzone compensation for our motors, and we can achieve better EKF results with a better motion model covariance calibration. Furthermore, we can use active LED modulation for more robust LED detection and correspondence. Looking toward the overall project goals mentioned in the introduction, a logical next step would be to put the camera on a moving platform. With our work, we have laid down a good foundation, but much work remains in the area of using multi-robot teams consisting of small and cheap robots to explore previously unreachable environments.

ACKNOWLEDGMENT

The authors would like to thank the Biomimetic Milisystems Lab, specifically Austin Buchan, Andrew Chen, Andrew Pullin, and James Lam Yi for their contributions to the Zumy platform development.

REFERENCES


