Accurate 3D Lighthouse Localization of a Low-Power Crystal-Free Single-Chip Mote

Brian G. Kilberg, Student Member, IEEE, Felipe M. R. Campos, Student Member, IEEE, Filip Maksimovic, Thomas Watteyne, Senior Member, IEEE, Kristofer S. J. Pister

Abstract—We present a system for centimeter-precision 3 dimensional localization of a 2×3×0.3 mm³, 5 mg, wireless system-on-chip by utilizing a temporally-structured infrared illumination scheme generated by a set of base stations. This 3D localization system builds on previous work by adding a second lighthouse station to enable 3D localization and using the integrated wireless radio, making the localization system fully wireless. We demonstrate 3D tracking with mean absolute errors of 1.54 cm, 1.50 cm, and 5.1 cm for the X, Y, and Z dimensions. This is the first time such a lighthouse localization system has been able to localize a monolithic single-chip wireless system.

I. INTRODUCTION

Autonomous microsystems, such as microrobots, have limited payload capabilities, which require their processing and communication platforms to be highly miniaturized. The 2×3×0.3 mm³, 5 mg, Single Chip Micro Mote (SCμM) [1] is an example of a suitable system for microrobots, containing an ARM Cortex-M0 microprocessor and a 2.4 GHz radio compatible with IEEE 802.15.4 and Bluetooth Low Energy. Localization of these systems is important for them to perform intelligent tasks in the real world. One promising localization system for SCμM is “lighthouse” localization, a type of optical localization with millimeter-scale precision that uses horizontally and vertically rotating planar laser sweeps and omnidirectional synchronization pulses. These pulses allow an object to determine its azimuth and elevation angles relative to a lighthouse base station [2], [3].

The tracked object in this system requires an infrared-sensitive photodiode that detects the lighthouse station’s laser sweeps. In [4], we demonstrated that SCμM’s integrated optical receiver, originally intended for contact-free optical programming, can be repurposed to accurately detect the laser sweeps generated by an HTC Vive lighthouse base station1, which enables calculation of the mote’s azimuth and elevation angles relative to the lighthouse. This work further develops this application and demonstrates the full 3D tracking of a SCμM chip, while using its on-board radio to wirelessly communicate its position.

II. THE SINGLE CHIP MICROMOTE (SCμM)

The Single Chip Micro Mote (SCμM), shown in Fig. 2, is a fully monolithic wireless System-on-a-Chip developed for microbotic applications [1]. It contains a crystal-free 2.4 GHz radio that is compatible with the IEEE 802.15.4 standard and has limited Bluetooth Low Energy transmit capability, an ARM Cortex-M0 processor with 64 kB of program and data memory, and an integrated optical receiver for contact-free programming [4]. Intended for payload-constrained applications like micro-robotics, the 2×3×0.3 mm³ SCμM chip does not require any external connections except for a power source and an antenna. For example, SCμM has been demonstrated driving an electrostatic inchworm-powered microrobotic gripper [5]. Additionally, Chang et al. successfully demonstrated SCμM running a low-power time-synchronized network protocol (6TiSCH) and participating in a time-synchronized channel hopping mesh network [6].

SCμM features an integrated optical receiver, initially designed for contact-free optical programming. It comes with the necessary circuitry which, upon receiving a specific 32 bit bootload start symbol, copies the subsequent bitstream into memory, and resets the micro-controller. A special optical programming board is used to transfer a compiled binary from a computer onto SCμM. The active power of the optical receiver is 1.5 μW, compared to the active power of the entire system, which can be up to 2 mW [4].

1 https://www.vive.com/eu/accessory/base-station/
A. Base Principle

Lighthouse-based localization was initially developed for localizing constrained low-power electronic objects [2]. In recent years, it has been applied for precisely and quickly measuring the position and orientation of Virtual Reality (VR) headsets such as the HTC Vive [3]. The base principle is that a lighthouse base station (which is typically the size of a coffee cup) is equipped with a laser which sweeps a laser plane across space, first horizontally, then vertically. An object equipped with a photodiode timestamps the moments it detects the pulse from the horizontal and vertical sweeps. Knowing the speed of those sweeps, it can compute its azimuth and elevation angles, and hence knows it is located on a ray relative to the lighthouse base station.

To obtain a 3D localization, at least two base station are needed, as depicted in Fig. 1. These lighthouses are connected by a wire which allows them to synchronize their pulses to one another; the role of each lighthouse (Lighthouse 1 or Lighthouse 2) is configured by a slider on each lighthouse. Fig. 3 shows a chronogram of the activity of each lighthouse. The lighthouses are equipped with two types of light sources: a high-powered omnidirectional infrared LED (used for sending sync pulses), and two lasers pointed at either of two mirrors that rotate at 120 Hz, at a 90 degree angle. The mirrors are used to sweep through space either in the azimuth or elevation, causing SCuM to receive a laser pulse. The lighthouses alternate between sending sync pulses and sweeping one of their lasers. Lighthouse 1 always sends a sync pulse before Lighthouse 2. The duration of the sync pulse indicates what the lighthouse will do, per Table I: sweep its azimuth laser, sweep its elevation laser, or keep its lasers off. The duration before receiving the laser pulse encodes the angle to the lighthouse. After receiving an azimuth and elevation pulse, SCuM can compute on what measurement ray it is relative to the lighthouse that send the pulses. Because of measurement inaccuracies, the measurement rays from both lighthouse base stations most likely don’t intersect. The challenge is hence to compute the position of the point in 3D space which minimize the distance to both rays.

TABLE I
DURATION OF THE PULSE ACTIVITY OF AN HTC VIVE LIGHTHOUSE.

<table>
<thead>
<tr>
<th>Pulse Type</th>
<th>Duration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{A}$</td>
<td>62.5 µs</td>
<td>sync pulse announcing azimuth sweep</td>
</tr>
<tr>
<td>$T_{E}$</td>
<td>72.9 µs</td>
<td>sync pulse announcing elevation sweep</td>
</tr>
<tr>
<td>$T_{S}$</td>
<td>104.0 µs</td>
<td>sync pulse announcing skip (no sweep)</td>
</tr>
<tr>
<td>$T_{sweep}$</td>
<td>8.3 ms</td>
<td>full 180 degree sweep of the laser</td>
</tr>
</tbody>
</table>

B. Receiving Lighthouse Laser Pulses on SCuM

As described in [4], SCuM is able to detect the pulse from an HTC Vive Lighthouse using its optical receiver. We develop custom firmware\(^2\) to process the structured infrared light emitted by the base stations. This firmware uses interrupts to detect, measure, and decode the laser scan timings and sync pulse widths that are received by SCuM’s optical receiver. The output of the optical receiver is routed to a GPIO output pin, then back into the processor via multiple GPIO pins: one connected to an active high level interrupt and one connected to an active low level interrupt in order to implement an edge sensitive interrupt.

A nearby computer is equipped with an OpenMote, a popular IEEE 802.15.4-based platform [7]. The OpenMote is programmed to listen to a particular frequency, and report to the computer the frames it receives. Upon measuring the timings from the laser pulses, the SCuM chip reports those values to the computer over 2.4 GHz IEEE 802.15.4. The Python-based program running on the computer then uses the timings to calculate the relative azimuth and elevation angles between each lighthouse base station and the SCuM chip.

C. Lighthouse Projection Model

Our work in [4] demonstrates the ability of a single lighthouse base station to provide accurate relative angle measurements between itself and a SCuM mote. A single base station, however, doesn’t provide enough information for 3D localization of a single, freely moving mote; relative angle measurements from another base station with known relative pose (translation and rotation) from the original base station are required to triangulate 3D position. Additionally, triangulation in three dimensions proves a challenge since the two directional rays, generated by the relative angle measurements from each lighthouse, are unlikely to intersect in 3D space due to inherent noise in the measurements. As a result, an estimation method is required to find the triangulation solution that minimizes the error between the two rays (see Fig. 1).

Triangulation is a well-studied problem commonly arising in computer vision with multi-perspective cameras [8]. In fact, modeling each lighthouse base station as a camera allows for both calibration of relative poses of each lighthouse in addition to the use of triangulation methods like the Direct

\(^2\) available at https://github.com/PisterLab/scum-test-code.
Fig. 3. Chronogram of the pulses received by the SC\(\mu\)M chip from the two lighthouses. The lighthouses are synchronized over a wired interface. They alternate between periods when sending omnidirectional sync pulses, and periods when sending laser pulses. Lighthouse 1 always sends a sync pulse before Lighthouse 2. The duration of the sync pulse indicates what the lighthouse will do, per Table I. The duration before receiving the laser pulse encodes the angle to the lighthouse. In this illustration, SC\(\mu\)M measures 120 degrees azimuth / 70 degrees elevation from Lighthouse 1, 110 degrees azimuth / 150 degrees elevation from Lighthouse 2.

Linear Transform (DLT) for 3D localization [8], [9]. The mathematical model that describes the transformation between a point in 3D global coordinates to a 2D point on the image plane of a pinhole camera is described by (1).

\[
X_{image} = \begin{bmatrix} x_{img} \\ y_{img} \\ 1 \end{bmatrix} = PX_{global}
\]

Eq. (2) is the definition of the camera projection matrix, which projects the 3D object onto the 2D camera’s image plane. \(K\) is the matrix representing the intrinsic camera parameters, which are the focal length \(f_x, f_y\) (the distance between the focal plane and the pin hole) and the principal point offset \((c_x, c_y)\), which is the offset between the center of the image plane and the pinhole. \(K\) is expressed in (3).

\[
P = K[R|t]
\]

\[
K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}
\]

The pose matrix \([R|t]\), expressed in (4), represents the transformation to rotate and translate points from the global frame to the camera frame. The rotation from global to camera frame is represented by \(R\), which is composed of \(r_{ij}\). The translation from the global frame to the camera frame, in camera frame coordinates, is represented by \(t_i\).

\[
[R|t] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}
\]

The global point being projected onto the camera’s image plane \(X_{global}\) is expressed in (5). It contains the global point in augmented homogeneous representation [8].

\[
X_{global} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}
\]

In our triangulation implementation, we model the lighthouse base stations as a pinhole camera, where the center of rotation of the lighthouse laser scans is the pinhole and the intrinsic matrix is \(I\). The azimuth and elevation measurements (range 0-\(\pi\), with \(\frac{\pi}{2}\) corresponding to the laser perpendicular to the image plane) are projected at unit distance onto this plane using (6). Fig. 4 illustrates this model.

\[
X_{image} = \begin{bmatrix} x_{img} \\ y_{img} \\ 1 \end{bmatrix} = \begin{bmatrix} \tan(\theta_{azimuth} - \frac{\pi}{2}) \\ \tan(\theta_{elevation} - \frac{\pi}{2}) \\ 1 \end{bmatrix}
\]

D. Triangulation

We use the Direct Linear Transform (DLT) [8] to triangulate the position of the chip using the relative angle measurements from each lighthouse base station. In the DLT, a system of equations, derived from (7) is set up using the camera projection matrices of each base station from the unknown global point. This system is shown in (8), where \(p^j\) is the \(j^{th}\) column of each base station’s camera projection matrix. The projection matrix and image coordinates of the second lighthouse base station are denoted \(p^j\) and \(x', y'\). This system of equations is solved using least squares\(^3\). The most recent of each azimuth and elevation measurements are used in the triangulation algorithm. Unfortunately this can lead to inaccuracies if one of these measurements are missed as an out-of-date measurement is then used in its place.

\[
PX = \begin{bmatrix} x_{image} \\ y_{image} \\ 1 \end{bmatrix}^T
\]

\(^3\) available at https://github.com/PisterLab/scuM_lighthouse_localization.
E. Calibration of Lighthouse Projection Matrices

Using the Direct Linear Transform (DLT) to triangulate the 3D position of $\text{SC}_\mu\text{M}$ in the lighthouse-camera model requires knowing both the intrinsic and extrinsic parameters of the two lighthouse base stations, corresponding to the matrices $K$ and $[R|t]$, respectively. Since we know $K$ to be the identity matrix $I$, the calibration task amounts to estimating the extrinsic parameters for each lighthouse in the lighthouse-camera model. In computer vision, this is known as the Perspective-$n$-Point problem (or just P$n$P) [10], whose aim is to determine a camera’s pose (position and orientation) given its intrinsic parameters and a set of $n \geq 3$ correspondences between global 3D points and their 2D projections onto the image plane [11].

For the purpose of this experiment, we build a sizeable set of 3D-2D correspondences: we isolate approx. 1000 sample points of the $\text{SC}_\mu\text{M}$ angle data projected onto the unit distance image plane, and their corresponding ground truth 3D position data. We use the nonlinear Levenberg-Marquardt iterative optimization algorithm to estimate a solution to the P$n$P problem [11]. The Levenberg-Marquardt algorithm attempts to find a solution $[\hat{R}|\hat{t}]$ to (9) by minimizing the residual sum of squared distances between the observed projections $[x_i^\text{img}, y_i^\text{img}, 1]^T$ and the projected ground truth points $[\hat{R}|\hat{t}][x \ y \ z \ 1]^T$.

\[
\begin{bmatrix}
  x_1^\text{img} & x_2^\text{img} & \cdots & x_n^\text{img} \\
  y_1^\text{img} & y_2^\text{img} & \cdots & y_n^\text{img} \\
  1 & 1 & \cdots & 1 
\end{bmatrix}
= [\hat{R}|\hat{t}]
\begin{bmatrix}
  x_1 & x_2 & \cdots & x_n \\
  y_1 & y_2 & \cdots & y_n \\
  z_1 & z_2 & \cdots & z_n \\
  1 & 1 & \cdots & 1 
\end{bmatrix}
\tag{9}
\]

To implement our calibration procedure, we used OpenCV’s solvePnP/Ransac method with the CV_ITERATIVE flag [12]. This uses the random sample consensus (RANSAC) iterative method on top of the standard OpenCV Levenberg-Marquardt optimization P$n$P solver to make the projection robust to outliers in $\text{SC}_\mu\text{M}$’s angle measurements. We apply the calibration to each lighthouse to recover their relative poses. Fig. 5 illustrates the results of using our calibration procedure by applying the pose of Lighthouse 1 that minimizes the reprojection error.

IV. RESULTS

We evaluate the accuracy of the $\text{SC}_\mu\text{M}$ lighthouse tracking system using the OptiTrack motion capture system. We move the $\text{SC}_\mu\text{M}$ chip along a trajectory, while tracking it with both the lighthouse system and the OptiTrack (we use the same methodology as in [4]). We make sure to synchronize the OptiTrack’s infrared exposure pulses with the lighthouse sync pulses to avoid interference.

Fig. 6 shows the azimuth and elevation angles measured during this experiment by each lighthouse, compared to the ground truth angles. Fig. 7 shows the error of the azimuth and elevation tracking; the interquartile range of the measurements is under 1 degree for three of the four angles. The RMS tracking error statistics are skewed by the presence of a few outlier errors. These outliers are observed to be significantly more frequent when the OptiTrack system is active, indicating that the OptiTrack’s infrared cameras are likely interfering with the lighthouse signal despite the synchronization protocol. When the outliers are removed in post-processing (error >10 degrees), the RMS tracking error is 0.63 deg, 0.37 deg, 0.60 deg, and 3.87 deg. Fig. 8 shows the 3D triangulated tracking data for the experiment, compared to ground truth. The mean absolute error for 3D tracking is 1.54 cm, 1.50 cm, and 5.1 cm for the X, Y, and Z dimensions.

We characterize the operating range of the $\text{SC}_\mu\text{M}$ lighthouse over distance from lighthouse and incidence angle of the $\text{SC}_\mu\text{M}$ mote relative to the lighthouse pulses. Specifically, we measure the percent of lighthouse measurements received at various distances from a lighthouse base station (Fig. 10), with the $\text{SC}_\mu\text{M}$ mote directly in front of the base station. The maximum effective range of the localization system is 1 m. We measure the effect of the angle of incidence on reliability by rotating the $\text{SC}_\mu\text{M}$ mote in a stationary position directly in front of two adjacent lighthouse base stations (Fig. 11). The maximum angle the $\text{SC}_\mu\text{M}$ mote could be rotated with respect to the lighthouses is 70°. As $\text{SC}_\mu\text{M}$’s optical receiver is designed for optical programming and not lighthouse localization, a future redesign of the receiver system should increase the detection range of the lighthouse signal.

V. FUTURE WORK

While these results are promising, several problems limit this system’s utility. First, intermittent outliers with significant
Fig. 6. Experimental azimuth and elevation measurements of a SC\(_\mu\)M mote compared to motion capture ground truth. The lighthouse-relative ground truth azimuth and elevation angles were determined by using the calibrated lighthouse poses to project the 3-dimensional world-frame ground truth trajectory to 2-dimensional lighthouse-relative azimuth and elevation angles.

Fig. 7. Violin plot showing distribution of azimuth and elevation angle error. One outlying data point that is 80 degrees from the mean of the elevation B data is not shown. Data points outside of these percentiles are not shown. The majority of error points are within a tight distribution, with a few outliers far from the mean. The RMS tracking error, with outliers (error >10 degrees) removed, is 0.63 deg, 0.37 deg, 0.60 deg, and 3.87 deg for azimuth 1, elevation 1, azimuth 2, and elevation 2, respectively.

Fig. 8. Triangulation of lighthouse data to reconstruct the trajectory of the SC\(_\mu\)M chip throughout the experiment. Large deviations in tracking are caused by missed measurements. In our system, a missed measurement gets replaced by the latest measurement. This allows for the system to always display a position, but causes the reconstruction algorithm to use out-of-date measurements, resulting in inaccurate positioning. The subset of data used for lighthouse calibration is excluded from this data.

Fig. 9. Triangulation error of lighthouse tracking over time. The RMS tracking error has significant temporal deviation throughout the experiment, characterized by periods of time with minimal error and periods of time with significant error. Periods of significant error are likely correlated with line-of-sight occlusion, infrared interference, or RF interference events.

error (>10 degrees) are present. Second, this system relies on line-of-sight and is vulnerable to occlusion of the lighthouse base stations. Third, the diminished lighthouse reception rate over distance and incidence angle can intermittently reduce the triangulation rate significantly. Finally, out-of-date measurements can cause inaccuracies in the triangulation algorithm, which relies on four measurements that do not occur simultaneously.

These problems could be solved by fusing the lighthouse measurements with inertial measurement unit (IMU) data. Sensor fusion algorithms like the Extended Kalman Filter (EKF) could use statistical measures such as the Mahalanobis
from direct individual lighthouse angle measurements instead of triangulation from four temporally-disparate measurements. Currently, the triangulation algorithm is calculated on a computer that is receiving SC$_M$M packets with lighthouse data. This is not scalable to large numbers of SC$_M$M chips, so it would be beneficial to implement the triangulation on-board the SC$_M$M chip itself. Further work must be done to determine the computational load of this triangulation method.

VI. CONCLUSION

We have demonstrated the 3D localization of a 1.8 mm$^3$ crystal-free wireless SoC using SC$_M$M’s integrated optical receiver and commercially available virtual reality tracking hardware. This system does not require any extra components on the SC$_M$M side, thus preserving its miniature form factor. Applications in microrobotics and personal telemetry could be improved by this localizcation system.