AAS 21-347

DIRECT VISUAL SLAM WITH A FILTER-BASED LIDAR MEASUREMENT INCORPORATION

Corey Marcus*, and Renato Zanetti†

In this paper we present a Direct Monocular SLAM system which is augmented with flash LIDAR images. The LIDAR measurements are incorporated with an Extended Kalman Filter and utilization of the Gamma and Inverse Gamma probability distribution functions. The system produces metric pose and map estimates. Monte Carlo methods are used to demonstrate that incorporating LIDAR measurements into the system provides significant performance improvements over a system without LIDAR in simulation. Experimental results using the EuRoC dataset are then presented which show improved system performance in real-time operation.

INTRODUCTION

Every time we enter a new room, our brains automatically construct a map of the room's geometry as well as our location within that geometry. This is a critical step in allowing us to control our motion or interact with our environment. There are many situations where a spacecraft might need to perform the exact same tasks autonomously. For example, a spacecraft performing an inspection of a target on-orbit must map the target while locating itself to avoid collision. In another scenario, a lander descending to the surface of the Moon or Mars may need to map the terrain in order to select a landing site. These situations require a solution for two related problems, estimating spacecraft spacecraft pose and mapping the local environment.

Separation of these two problems ignores the common errors and correlations between them, resulting in degraded performance. SLAM is effective because it acknowledges these correlations and seeks to minimize errors across both mapping and localization simultaneously. Simultaneous localization and mapping (SLAM)[1] is a powerful class of algorithms for these dual problems which couples their solutions. This paper modifies a direct monocular SLAM system to accept flash LIDAR images. Cameras and LIDAR systems have a strong heritage in spaceflight and a SLAM system based on them is of relevance to spaceflight problems concerning autonomous orbital proximity operations and/or landing.

The basic premise of SLAM is to simultaneously refine estimates of the environment around the vehicle and the vehicle's location within that space. Measurements of the environment are often relative to the sensor (*e.g.* camera images, inertial measurements), rather than absolute (*e.g.* GPS). This causes a coupling between estimates of sensor location and environment geometry. Poor estimates of one negatively affect the other. First proposed in 1988 by Cheeseman et al,[1] SLAM is not a new development. However, advances in computational power meant that SLAM with cameras was not implementable on a large scale or in real-time

*PhD Student, Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin.
†Assistant Professor, Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin.

until much more recently. Klein and Murray's Parallel Tracking and Mapping (PTAM)[2] represents one of the earliest and best known fully fledged SLAM architectures.

Most current bundle adjustment[3] visual SLAM systems fall into one of two categories, feature based or direct. Feature based methods form the bulk of modern SLAM algorithms. State of the art methodologies such as ORB-SLAM2[4] use a feature identification and matching algorithm such as ORB,[5] SURF,[6] or SIFT[7] to identify objects which are likely to correspond to the same physical features across multiple images. The features tend to correspond to things such as corners or changes in color on a surface. Feature based methods struggle with large textureless environments and discard the images after features have been matched. This approach utilizes a relatively small amount of the information contained within an image. Direct approaches seek to increase the amount of information extracted from an image by aligning the intensities of each pixel across multiple images. Every image is used "directly" in the SLAM algorithm. These types of algorithms tend to produce much denser maps than feature based approaches, as every pixel in an image could potentially become a map point. Until recently, direct SLAM was not real-time capable, but methodologies such as LSD-SLAM[8] have shown that it can be used to create semi-dense maps in real time with only a CPU. Direct SLAM also struggles with textureless environments, and textureless segments of the environment are often sparsely mapped.

SLAM with a monocular camera has one notable drawback, scale is not observable. Resolution of this problem necessitates additional sensors. Inertial sensors are frequently used to estimate the sensor translation between images, thus resolving scale. One such example of this utilization is shown by Mur et al.[9] LIDAR provides another convenient method for scale where measurements are used to estimate the depth of a feature[10] or pixels by Shin *et al*.[11] These works are the most similiar to the work produced within this paper. Shin concludes that direct SLAM is a better candidate for LIDAR integration because the association between sparse LIDAR measurements and sparse features can be tenuous. They integrate LIDAR measurements into their SLAM system through inclusion in their non-linear least squares optimization scheme. That is to say, they do not take a Bayesian approach to updating the map with information provided by LIDAR. Furthermore, they discard a large number of LIDAR measurements which align with low texture regions of the images. This paper differs from previous approaches by doing both of these things.

LSD-SLAM

Our work expands upon the framework provided by Large Scale Direct SLAM (LSD-SLAM)[8] by including LIDAR images in addition to the camera, or red-green-blue (RGB), images used by LSD-SLAM. LSD-SLAM is an open-source direct monocular SLAM algorithm. Direct SLAM algorithms work without the feature matching algorithms found in many other SLAM systems such as ORB-SLAM2.[4] Instead, the depths of individual pixels themselves are estimated directly. LSD-SLAM divides the global map into a series of local maps which are anchored on selected images called keyframes. LSD-SLAM chooses to estimate the *inverse* depth and associated variance for a subset of pixels in each keyframe. Because scale is unobservable for monocular camera based systems, the map produced by LSD-SLAM is non-metric. LSD-SLAM scales its maps such that the average inverse depth estimate is one. This promotes well behaved linearization of the functions governing variance propagation and allows the creation of a metric controlling keyframe creation which is independent of the local environment's true scale. Since LIDAR measurements are metric, they can be used to estimate the true scale of the environment and create metric maps.

LSD-SLAM was chosen for the base of our system primarily for the density of maps created. A dense map is

beneficial for LIDAR measurement incorporation as it increases the likelihood that a LIDAR measurement directly impinges upon a map point.

LIDAR MEASUREMENTS

Each LIDAR measurement, d_i, is assumed to come from a sensor which is collocated with the monocular camera. Thus, the calibration of these measurements is defined according to an azimuth, θ_i, and elevation, ϕ_i, angle specified in the camera's coordinate system. These angles are diagrammed in Figure 1.

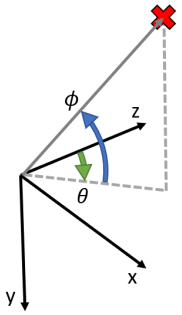

Figure 1: A diagram of the coordinate system defining a LIDAR measurement (red "X"). Note the definition of the blue elevation angle, ϕ, and the green azimuth angle, θ.

The measurement can be expressed as a three dimensional vector, P_i^l, in the camera coordinate frame according to Equation (1).

$$P^l = d_i \begin{bmatrix} \sin(\theta_i)\cos(\phi_i) \\ -\sin(\phi_i) \\ \cos(\phi_i)\cos(\theta_i) \end{bmatrix} \tag{1}$$

LSD-SLAM MODIFICATIONS

LIDAR measurements are used in changes to two portions of LSD-SLAM. First, a LIDAR image is used to initialize the map on system start-up. Second, LIDAR measurements are used to estimate scale and refine the map in the mapping thread. The mapping thread is outlined below in Figure 2.

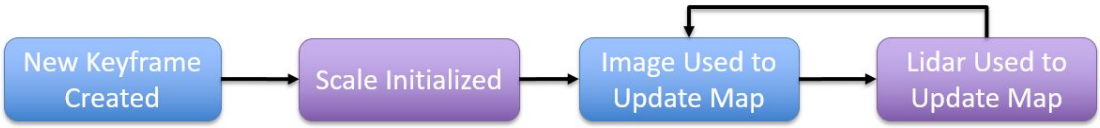

Figure 2: An overview of the SLAM system's mapping thread. The elements *New Keyframe Created* and *Image Used to Update Map* are found within LSD-SLAM while the other two have been created as part of this work.

The mapping thread begins when a new keyframe is initialized. Next, LIDAR measurements are used to estimate the keyframe scale. From this point until a new keyframe is created, subsequent LIDAR and camera images are used to refine the map. The LIDAR based modifications are outlined below.

Map Initialization

LSD-SLAM chooses to initialize the map randomly with a large variance. Our system initializes map points with LIDAR measurements when possible, and randomly when no LIDAR measurement is present. For the first RGB and LIDAR image pair, every pixel matching a LIDAR measurement is initialized to a depth prescribed by that measurement. The remainder are initialized randomly. This process is non-trivial as LIDAR measurements are of range while map elements are inverse depth estimates and scaled to have an average of one. Note that the camera and LIDAR system are assumed to be collocated, thus LIDAR measurements can always be associated with image pixels when the RGB and LIDAR images are collected at the same time.

The LIDAR provides range measurements, d_i, which have been corrupted with zero-mean measurement noise, ω_i. The variance of this noise is defined as σ_ω^2. This range measurement is matched with a pixel using the camera calibration matrix.[12] We would like to initialize the inverse depth estimate of this pixel according to this measurement. In order to provide a closed form solution, we assume each d_i is sampled from an Inverse Gamma probability distribution function. The simulation outlined in the following sections actually uses an ω_i which is distributed according to a Gaussian with zero-mean and variance σ_ω^2, but the approximation of the measurement as an Inverse Gamma is good for Gaussians with a large positive mean and small variance. These assumptions are well suited to our LIDAR system where the range measurements tend to be several orders of magnitude larger than σ_ω^2 and are always positive.

The Inverse Gamma distribution is defined by the shape, a_i, and scale, b_i, parameters. These can be found as function of d_i and σ_ω^2 in Equations (2) and (3).

$$a_i = 2 + \frac{d_i^2}{\sigma_\omega^2} \tag{2}$$

$$b_i = d_i \left(1 + \frac{d_i^2}{\sigma_\omega^2}\right) \tag{3}$$

We have a range measurement but are interested in a depth measurement, thus we must scale the Inverse Gamma distribution by a factor of $\cos(\theta_i)\cos(\phi_i)$ from Equation (1). When scaling an Inverse Gamma distribution, the scale parameter must also be scaled. We term the new scale parameter $\bar{b}_i$ and find it according to, $\bar{b}_i = \cos(\theta_i)\cos(\phi_i)b_i$. We have a new random variable, $x_i \sim \mathcal{G}^{-1}\left(a_i, \bar{b}_i\right)$, representing the depth of the map point.

We are interested in the distribution of the inverse depth, $\rho_i = 1/x_i$. ρ_i is distributed according to a Gamma probability distribution function and the relationship between Gamma and Inverse Gamma distributions is known and analytic. The distribution of ρ_i is defined by the same parameters as x_i; however, the scale parameter, b_i, is now termed an inverse scale parameter. Thus, $\rho_i \sim \mathcal{G}\left(a_i, \bar{b}_i\right)$.

The inverse depths are all scaled such that their mean is equal to one. This requires a new scale factor, $k = \text{mean}(\frac{1}{x_i})$, and a redefinition of $\rho_i = \frac{1}{kx_i}$. The inverse scale parameter must be multiplied by k. The distribution of the inverse depth estimate under these assumptions is now given by $\rho_i \sim \mathcal{G}\left(a_i, k\bar{b}_i\right)$. The

mean, $\hat{\mu}_i$, and variance, $\hat{\sigma}_i^2$, of ρ_i are found according to the following:

$$\hat{\mu}_i = \frac{a_i}{k\bar{b}_i} = \frac{2 + \frac{d_i^2}{\omega^2}}{k\cos(\theta_i)\cos(\phi_i)d_i\left(1 + \frac{d_i^2}{\sigma_\omega^2}\right)} \tag{4}$$

$$\hat{\sigma}_i^2 = \frac{a_i}{\left(k\bar{b}_i\right)^2} = \frac{2 + \frac{d_i^2}{\omega^2}}{\left(k\cos(\theta_i)\cos(\phi_i)d_i\left(1 + \frac{d_i^2}{\sigma_\omega^2}\right)\right)^2} \tag{5}$$

Scale Initialization

When the camera has rotated and translated sufficiently far from the origin of the local map, a new keyframe is created. Points in the current map are propagated into the new one in order to initialize. The LIDAR image associated with the keyframe is used to estimate the scale between the non-metric map and the metric environment, s. This scale is defined such that a non-metric quantity y_1 could be converted to a metric representation, y_2 with $y_2 = sy_1$.

Each LIDAR range measurement, d_i, has a corresponding azimuth and elevation. This information is used to map d_i into a 3D vector, P_i^l, expressed in the camera's coordinate system, Equation (1). P_i^l is matched with a pixel in the keyframe using the camera calibration matrix. If this pixel has a valid inverse depth estimate ρ_i and associated variance σ_i^2, the estimate, estimate variance, and the depth element of the 3D vector, $P_i^l(3)$, are used to form a measurement of the scale, β_i, according to Equation (6).

$$\beta_i = \rho_i P_i^l(3) \tag{6}$$

A maximum likelihood estimator is employed to estimate the scale, $\hat{s}$. The likelihood function is taken as a Gaussian sum, where each element of the sum is distributed according to $\mathcal{N}\left(\beta_i, \sigma_i^2\right)$. Thus, $\hat{s}$ is given according Equation (7). An approximation of $\hat{s}$ is found by first taking a Least-Squares estimate of s. The domain surrounding this Least-Squares approximation is discretized and the Gaussian sum is evaluated at each discrete point. The point with the highest likelihood is accepted as $\hat{s}$ and the variance of the Least-Squares estimate is taken as the maximum liklihood variance.

$$\hat{s} = \arg\max_s \sum_i \mathcal{N}\left(s; \beta_i, \sigma_i^2\right) \tag{7}$$

This method is preferred over a gradient based optimization scheme due to the large number of local maximums in the Gaussian sum.

Map Update With Lidar

Each new LIDAR image after the keyframe is used to update the map. The coordinate frame of the senor during measurement is termed the observer frame. Again, each LIDAR range measurement, d_i, is used to form a 3D vector, P_i^l, in the camera frame. The cameras coordinate system has been rotated, R_{o2m},

and translated, t^m_{m2o}, away from the map's coordinate system by the movement of the sensor. P^l_i can be expressed in the non-metric map frame according to Equation (8).

$$P^m_i = \frac{1}{\hat{s}} R_{o2m} P^l_i - t^m_{m2o} \tag{8}$$

P^m_i can then be associated with a pixel in the keyframe using the camera calibration matrix. If this pixel has a valid inverse depth estimate, the measurement will used to perform an Extended Kalman Filter (EKF) update. If it does not, an inverse depth estimate will be initialized as detailed in the following section.

For the case in which an inverse depth estimate, $\hat{\rho}_i$, already exists, it must first be mapped into a predicted measurement in the observer frame. The mapping of $\hat{\rho}_i$ into a $\bar{d}_i$ is given by Equation (9)

$$\bar{d}_i = \hat{s} \sqrt{\frac{1}{\left(\hat{\rho}_i \hat{P}^m_i(3)\right)^2} - \frac{2}{\hat{\rho}_i \hat{P}^m_i(3)} \left(\hat{P}^m_i \cdot t^m_{m2o}\right) + (t^m_{m2o} \cdot t^m_{m2o})} \tag{9}$$

Where $\hat{P}^m_i$ is the unit vector along P^m_i and $(a \cdot b)$ indicates the inner product of vectors a and b.

In practice, direct estimation of ρ_i with an EKF or UKF leads to a smug and biased filter. This is due to the non-linearity of the inverse operation. Noting that any inverse depth in the map will be greater than zero, we assume that the inverse depth has a probability density function given according to a Gamma distribution, $\mathcal{G}(a_i, b_i)$. Shape parameter a_i and inverse scale b_i can be found as analytic expressions of the distribution's mean and variance, ρ_i and σ^2_i. These are given in Equations (10) and (11).

$$a_i = \frac{\rho^2_i}{\sigma^2_i} \tag{10}$$

$$b_i = \frac{\rho_i}{\sigma^2_i} \tag{11}$$

If the random variable ρ_i is distributed according to $\mathcal{G}(a_i, b_i)$, its inverse, $\delta_i = 1/\rho_i$, is distributed according to an Inverse Gamma, $\mathcal{G}^{-1}(a_i, b_i)$. Here a_i remains the shape parameter, but b_i is known as the scale parameter. The mean, μ_i and variance, Σ_i of δ_i are known and analytic, Equations (12) and (13).

$$\mu_i = \frac{b_i}{a_i - 1} \tag{12}$$

$$\Sigma_i = \frac{b^2_i}{(a_i - 1)^2 (a_i - 2)} \tag{13}$$

Estimating $\rho_i \sim \mathcal{G}(a_i, b_i)$ is equivalent to estimating $\delta_i \sim \mathcal{G}^{-1}(a_i, b_i)$. This leads to a new formation for the predicted measurement given in Equation (14).

$$\bar{d}_i = \hat{s}\sqrt{\frac{\delta_i^2}{\hat{u}_m(3)^2} - \frac{2\delta_i}{\hat{u}_m(3)}\left(\hat{P}_i^m \cdot t_{m2o}^m\right) + (t_{m2o}^m \cdot t_{m2o}^m)} \tag{14}$$

We estimate δ_i with an EKF, so the partial derivative of Equation (14) with respect to δ_i is required to find a first order approximation of the variance of $\bar{d}_i$, Equation (15).

$$\frac{\partial \bar{d}_i}{\partial \delta_i} = \frac{\hat{s}\left(\frac{2\delta_i}{\hat{u}_m(3)^2} - \frac{2}{\hat{u}_m(3)}\left(\hat{P}_i^m \cdot t_{m2o}^m\right)\right)}{\sqrt{\frac{\delta_i^2}{\hat{u}_m(3)^2} - \frac{2\delta_i}{\hat{u}_m(3)}\left(\hat{P}_i^m \cdot t_{m2o}^m\right) + (t_{m2o}^m \cdot t_{m2o}^m)}} \tag{15}$$

Now, an update to the mean and variance of δ_i can be found with the EKF equations.

$$\mu_i^+ = \mu_i^- + \frac{\left[\frac{\partial \bar{d}_i}{\partial \delta_i}|_{\mu_i^-}\right]^2 \Sigma_i^-}{\left[\frac{\partial \bar{d}_i}{\partial \delta_i}|_{\mu_i^-}\right]^2 \Sigma_i^- + \sigma_\omega^2}(d_i - \bar{d}_i) \tag{16}$$

$$\Sigma_i^+ = \Sigma_i^- - \frac{\left(\left[\frac{\partial \bar{d}_i}{\partial \delta_i}|_{\mu_i^-}\right]^2 \Sigma_i\right)^2}{\left[\frac{\partial \bar{d}_i}{\partial \delta_i}|_{\mu_i^-}\right]^2 \Sigma_i + \sigma_\omega^2} \tag{17}$$

The updated mean, μ_i^+, and variance, Σ_i^-, are then transformed back into Inverse Gamma shape and scale parameters. These are converted to shape and inverse scale parameters of a Gamma distribution representing the inverse depth, and then finally into the new values for the mean and variance of the estimate of inverse depth ρ_i.

We have found that estimating ρ_i indirectly through estimation of δ_i with an EKF and Equations (16) and (17) leads to an estimator with better performance than one which uses an EKF to directly estimate ρ_i. An overview of the process is outlined in Algorithm 1.

Map Point Initialization

Often a LIDAR measurement will be mapped into the keyframe and there will not be a valid inverse depth estimate at that pixel. In this case, the measurement forms the initialization of the inverse depth estimate. The inverse depth estimate is initialized with process similar to the one used to initialize the SLAM system. This time, the new inverse depth estimates are scaled with the scale estimate found during keyframe creation, as opposed to the scaling such that their mean is one as is done in SLAM initialization.

As before, we have a range measurement, d_i, which has been corrupted with zero-mean noise, ω. The variance of this noise is defined as σ_ω^2. This range measurement has been matched with a pixel whose

inverse depth estimate we seek to initialize. We assume the distribution of our measurement is an Inverse Gamma, $\mathcal{G}^{-1}(a_i, b_i)$. The parameters a_i and b_i are defined as a function of the measurement and its variance according to Equations (2) and (3).

The measurement can be converted to a 3D vector, P_i^l, with Equation (1) and mapped into the coordinate system of the keyframe with Equation (8). This set of operations is formalized in Equation (18).

$$P_i^m = \frac{d_i}{\hat{s}} R_{o2m} P_i^l - t_{m2o}^m \tag{18}$$

We are interested in the third element (depth) of P_i^m. $P_i^m(3)$ has a distribution expressed as a transformation of the Inverse Gamma distribution of d_i with an offset of $-t_{m2o}^m(3)$. Thus its mean and variance can be found explictly as a function of d_i and σ_ω^2 as shown in Equations (19) and (20).

$$\mu_i = \frac{d_i}{\hat{s}} \left(R_{o2m}(3,:) \cdot P_i^l \right) - t_{m2o}^m(3) \tag{19}$$

$$\sigma_i^2 = \frac{\sigma_\omega^2}{\hat{s}^2} \left(R_{o2m}(3,:) \cdot P_i^l \right)^2 \tag{20}$$

Where $R_{o2m}(3,:)$ is the 3[rd] row of R_{o2m} and $(a \cdot b)$ is the inner product of a and b.

$P_i^m(3)$ is not distributed according to an Inverse Gamma for non-zero $t_{m2o}^m(3)$, however, we will assume that an Inverse Gamma is a decent approximation. This approximation is valid so long as μ_i is sufficiently large and σ_i^2 is sufficiently small. We can find the a_i and b_i parameters of this distribution according to Equations (2) and (3).

We now assume our inverse depth estimate is distributed according to a Gamma with the same a_i and b_i parameters. The mean and variance of the inverse depth estimate can then be initialized according to Equations (21) and (22).

$$\hat{\rho}_i = \frac{a_i}{b_i} = \frac{2 + \frac{\mu_i^2}{\sigma_i^2}}{\mu_i \left(1 + \frac{\mu_i^2}{\sigma_i^2} \right)} \tag{21}$$

$$\hat{\sigma}_i^2 = \frac{a_i}{b_i^2} = \frac{2 + \frac{\mu_i^2}{\sigma_i^2}}{\mu_i^2 \left(1 + \frac{\mu_i^2}{\sigma_i^2} \right)^2} \tag{22}$$

The entire map update process is outlined below in Algorithm 1.

Algorithm 1 Map Update From LIDAR

for $i = 0; i < N; i++$ **do**
  Transform d_i into the map frame and match it with a pixel
  if The pixel has a valid inverse depth estimate **then**
    Assume the inverse depth estimate is of a Gamma distribution
    Assume the range measurement is sampled from an Inverse Gamma distribution
    Find the inverse depth estimate's shape and inverse scale parameters
    Use these parameters to find the mean and variance of the depth estimate
    Transform the depth estimate into a predicted measurement using the linearized transformation
    Perform an EKF update of the depth estimate using the LIDAR measurement
    Transform the updated Inverse Gamma depth back into a new Gamma inverse depth estimate
  else
    Assume the range measurement error is distributed according to an Inverse Gamma
    Assume the inverse depth estimate we will initialize is distributed according to a Gamma
    Map the range measurement to a measurement of depth in the keyframe, find its mean and variance
    Find the mean and variance of the new inverse depth estimate
  end if
end for

Map Smoothing

The map is smoothed using the LSD-SLAM smoothing algorithm after all EKF updates are performed and map points are initialized from the LIDAR measurements. Each inverse depth estimate is reassigned according to the average of the surrounding estimates weighted by the respective inverse estimate variances. Since the EKF update tends to significantly reduce the variance of an estimate, the surrounding inverse depths are pulled in the direction of the update by this process. Thus, the information from the LIDAR is able to update map points which are not directly measured. Smoothing can be justified from a theoretical perspective as the depths of adjacent pixels are highly correlated. Thus a LIDAR measurement of one pixel does provide information on the adjacent ones.

The algorithm used to perform the smoothing is identical to the one found in the unmodified LSD-SLAM.

SLAM SIMULATION

We have created a full environmental simulation tool which produces RGB and LIDAR images for the SLAM algorithm. This tool allows us to better understand the efficacy of our SLAM system when compared to experimental data by allowing the true environment and system pose to be known.

The simulation produces 480×640 RGB images, Figure 3, as well as 50×50 LIDAR images, Figure 4, which simulate the performance of a flash LIDAR. The RGB images are generated according to a pinhole camera model.[12] Lidar range measurments are created by measuring the range until intersection along a vector specified by an azimuth and elevation. The focal points of the camera and LIDAR are assumed to be aligned, and each camera image corresponding to one LIDAR image taken at the exact same time.

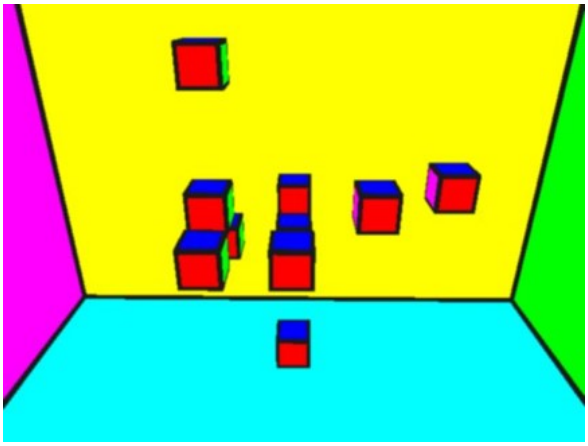

Figure 3: A RGB Image produced by the SLAM simulation.

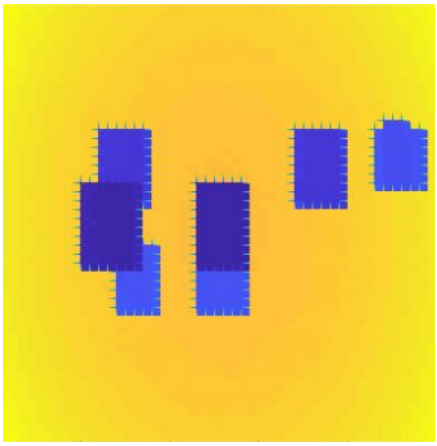

Figure 4: A LIDAR image produced by the simulation (colorized to demonstrate depth).

RESULTS

Simulation

We have performed a Monte Carlo analysis using the simulation described above. First we consider the system's tracking performance with and without LIDAR EKF updates. We used 100 Monte Carlo runs for each case. Lidar is used in both cases to estimate the scale of the non-metric map. Figure 5 shows the absolute error in position estimation with respect to the keyframe. On average the absolute position error is lower when LIDAR measurements are included. Figure 6 shows the "XYZ" error Euler Angles in attitude estimation with respect to the keyframe. Again, the error is on average lower when LIDAR measurements are included.

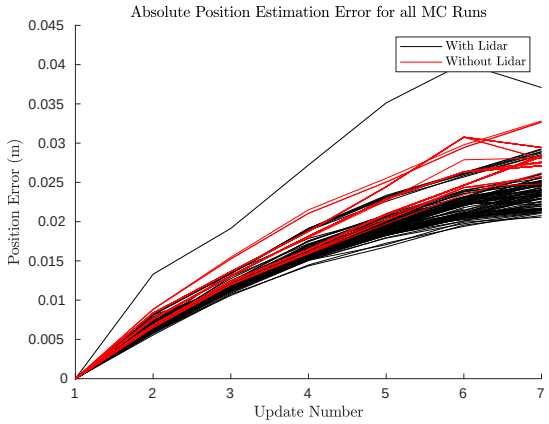

Figure 5: Absolute position error for all MC runs.

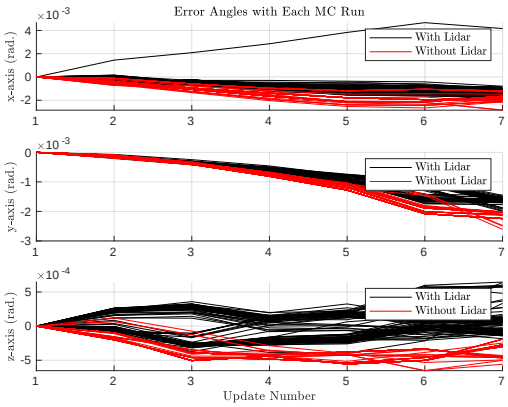

Figure 6: Error Euler Angles for all MC runs.

Next, we consider the system's mapping performance with LIDAR EKF updates. In Figure 7 we analyze

the map points directly updated by the EKF. In each update step of each Monte Carlo run, the difference in absolute error before and after the EKF is calculated. These differences are averaged and displayed in the upper portion of the figure. In general, the average difference is positive. This implies that the absolute mapping error before the update is higher than after and the EKF in general improves mapping performance. The lower portion tracks the number of points updated.

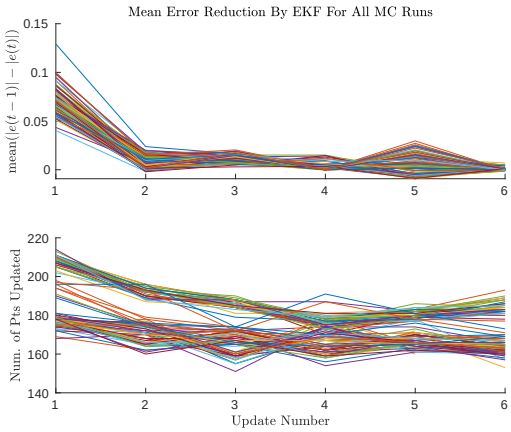

Figure 7: LIDAR error reduction for all MC runs based on EKF updated points only.

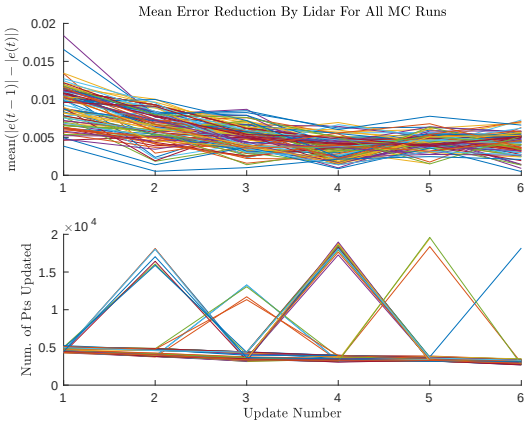

Figure 8: LIDAR error reduction for all MC runs including EKF and smoothing.

Figure 8 shows the same average difference metric as Figure 7. In this figure, the metric is based on all map points affected by the EKF update and smoothing process. Again, the average difference is in general positive which implies the EKF and smoothing process reduce mapping error. We can see based on the number of points updated, that large portions of the map are affected by LIDAR measurements. In some cases approximately fifteen thousand map points are updated.

EuRoC Dataset

The European Robotics Challenge (EuRoC) dataset is a high quality dataset intended for use with SLAM and other related algorithms.[13] It contains stereo camera, IMU, and 3D positioning (Vicon) measurements of a UAV flying through a small indoor space. Also included is all calibration information, a high density point cloud of the room's geometry, and a best estimate of the UAV's true pose found from applying a smoothing process to the IMU and 3D positioning measurements. The dataset contains multiple runs through multiple environments. A representative image of the environment chosen is shown in Figure 9

The dataset's pointcloud and estimates of the UAV's pose were used to create simulated flash LIDAR images. Each LIDAR range measurement was corrupted with Gaussian noise. These flash LIDAR images and the camera images from one side of the stereo camera system were fed into the SLAM system at 20 Hz on a *Dell Optiplex 7060* workstation. Two runs were performed. In both runs, the LIDAR images were used to estimate the true scale of the SLAM map. In only one run were the LIDAR measurements used to update the map. Figure 10 shows screenshots of the LSD-SLAM user interface with and without using LIDAR range measurements to update the map. The pixels with depth estimates are colored to represent their range. As in the simulation, map density is increased through using LIDAR measurements to initialize map points.

Figure 9: A representative image of the Vicon Room environment used in the EuRoC dataset.

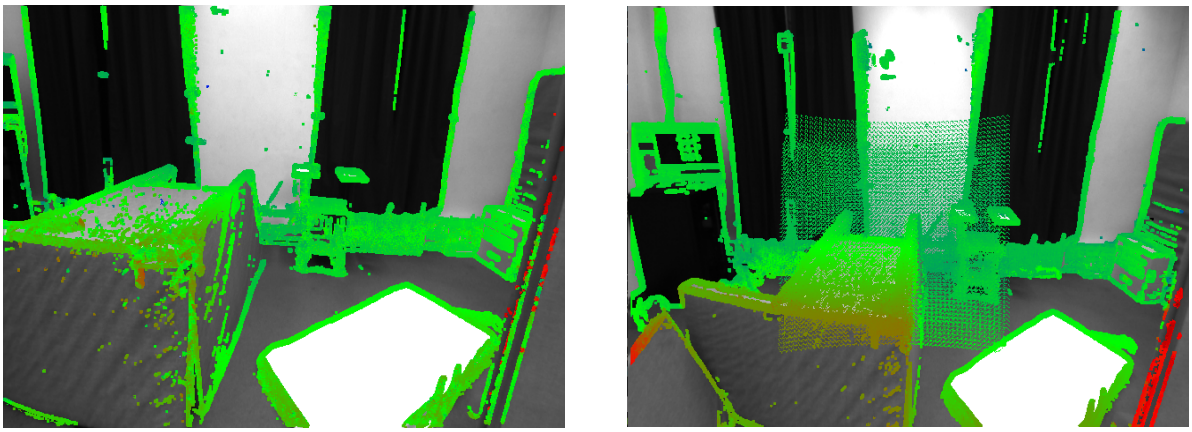

Figure 10: The LSD-SLAM user interface without and with LIDAR map updates (left and right, respectively). Note depth estimates are largely absent from portions of the image without significant texture when LIDAR measurements are not used to update the map. The user interface when LIDAR map updates are performed shows that these regions are able to be estimated using LIDAR measurements.

Figure 11 shows the localization performance for the SLAM system in each case alongside the true trajectory as provided by the dataset. LIDAR updates of the map result in localization estimates with lower drift in the position estimates. Drift in these estimates is a problem found in many SLAM systems without absolute measurements of position. Furthermore, LIDAR updates result in a system with all around more robust localization performance. Tracking is lost in the system without LIDAR updates at approximately 90

seconds into the flight due to rapid UAV movement. The system with LIDAR updates is able to continue providing localization estimates through this segment of the flight.

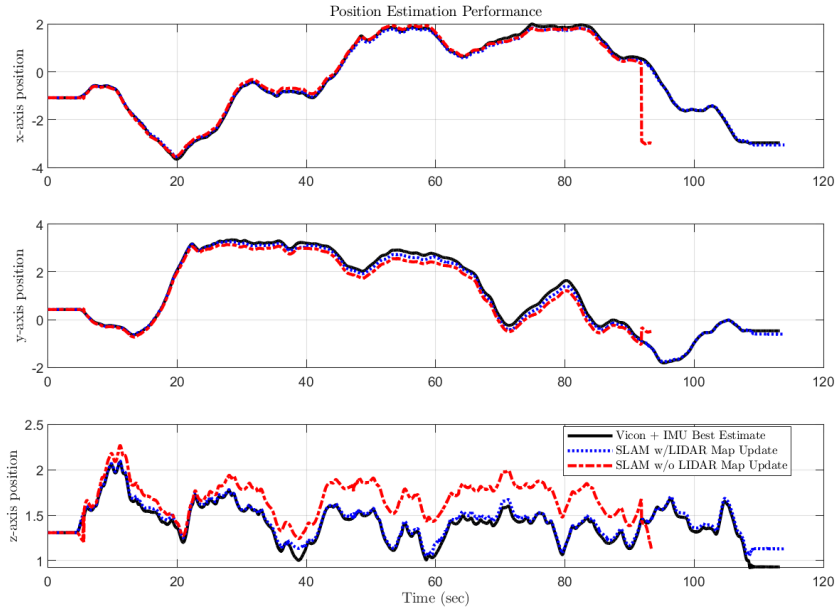

Figure 11: Real-time localization performance with and without LIDAR map updates. Note the increased localization drift and loss of tracking when LIDAR is not used to update the map. Even with LIDAR updates, tracking performance is poor in the last 10 seconds of flight as the UAV lands. This is due to the UAV approaching the texture-less landing surface in a manner such that the surface fills the majority of the camera's field of view.

CONCLUSION

We have demonstrated an effective and novel algorithm for SLAM using monocular cameras and flash LIDAR. Our system produces a measurable improvement over methods using a monocular camera alone. First, LIDAR measurements make the scale of the SLAM map observable. Second, LIDAR measurements are incorporated into the map with an EKF and smoothing algorithm, directly improving mapping performance. Third, the improved map results in improved pose estimation and more robust tracking. The combination of Monte Carlo simulation and real-time operation from real data provides additional confidence that this system would perform well on a real spacecraft.

ACKNOWLEDGEMENTS

This work was supported by a NASA Space Technology Graduate Research Opportunity.

REFERENCES

[1] P. Cheeseman, R. Smith, and M. Self, "A stochastic map for uncertain spatial relationships," *4th International Symposium on Robotic Research*, 1987, pp. 467–474.

[2] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, IEEE Computer Society, 2007, pp. 1–10.

[3] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," *International workshop on vision algorithms*, Springer, 1999, pp. 298–372.

[4] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, Vol. 33, No. 5, 2017, pp. 1255–1262.

[5] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *2011 International conference on computer vision*, Ieee, 2011, pp. 2564–2571.

[6] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *European conference on computer vision*, Springer, 2006, pp. 404–417.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, Vol. 60, No. 2, 2004, pp. 91–110.

[8] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," *European conference on computer vision*, Springer, 2014, pp. 834–849.

[9] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular SLAM with map reuse," *IEEE Robotics and Automation Letters*, Vol. 2, No. 2, 2017, pp. 796–803.

[10] J. Graeter, A. Wilczynski, and M. Lauer, "Limo: Lidar-monocular visual odometry," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 7872–7879.

[11] Y.-S. Shin, Y. S. Park, and A. Kim, "Direct visual SLAM using sparse depth for camera-lidar system," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 1–8.

[12] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[13] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research*, Vol. 35, No. 10, 2016, pp. 1157–1163.