Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2013, Article ID 374165, 15 pages
http://dx.doi.org/10.1155/2013/374165

Research Article

Monocular Vision SLAM for Indoor Aerial Vehicles

Koray Çelik and Arun K. Somani

Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50010, USA

Correspondence should be addressed to Koray Çelik; koray@iastate.edu

Received 14 October 2012; Accepted 23 January 2013

Academic Editor: Jorge Dias

This paper presents a novel indoor navigation and ranging strategy via monocular camera. By exploiting the architectural orthogonality of the indoor environments, we introduce a new method to estimate range and vehicle states from a monocular camera for vision-based SLAM. The navigation strategy assumes an indoor or indoor-like manmade environment whose layout is previously unknown, GPS-denied, representable via energy based feature points, and straight architectural lines. We experimentally validate the proposed algorithms on a fully self-contained microaerial vehicle (MAV) with sophisticated on-board image processing and SLAM capabilities. Building and enabling such a small aerial vehicle to fly in tight corridors is a significant technological challenge, especially in the absence of GPS signals and with limited sensing options. Experimental results show that the system is only limited by the capabilities of the camera and environmental entropy.

1. Introduction

The critical advantage of vision over active proximity sensors, such as laser range finders, is the information to weight ratio. Nevertheless, as the surroundings are captured indirectly through photometric effects, extracting absolute depth information from a single monocular image alone is an ill-posed problem. In this paper, we have aimed to address this problem with as minimal use of additional information as possible for the specific case of a rotorcraft MAV where size, weight, and power (SWaP) constraints are severe and investigate the feasibility of low-weight and low-power monocular vision-based navigation solution. Although we emphasize MAV use in this paper, our approach has been tested and proved perfectly compatible with ground based mobile robots, as well as wearable cameras such as helmet or tactical vest mounted device, and further, it can be used to augment the reliability of several other types of sensors. Considering the foreseeable future of intelligence, surveillance and reconnaissance missions will involve GPS-denied environments; portable vision-SLAM capabilities can pave the way for a GPS-free navigation systems.

Our approach is inspired by how intelligent animals such as cats and bats interpret depth via monocular visual cues such as relative height, texture gradient, and motion parallax [1] by subconsciously tracking dense elements such as foliage. We integrate this ranging technique with SLAM to achieve autonomous indoor navigation of an MAV.

1.1. Related Work on Vision-Based SLAM. Addressing the depth problem, the literature resorted to various methods such as the Scheimpflug principle, structure from motion, optical flow, and stereo vision. The use of moving lenses for monocular depth extraction [2] is not practical for SLAM, since this method cannot focus at multiple depths at once. The dependence of stereo vision on ocular separation [3] limits its useful range. And image patches obtained via optical flow sensors [4, 5] are too ambiguous for the landmark association procedure for SLAM. In sensing, efforts to retrieve depth information from a still image by using machine learning such as the Markov Random Field learning algorithm [6, 7] are shown to be effective. However, a-priori information about the environment must be obtained from a training set of images, which disqualifies them for an online-SLAM algorithm in an unknown environment. Structure from Motion (SFM) [3, 8, 9] may be suitable for the offline-SLAM problem. However, an automatic analysis of the recorded footage from a completed mission cannot scale to a consistent localization over arbitrarily long sequences in real time. Methods such

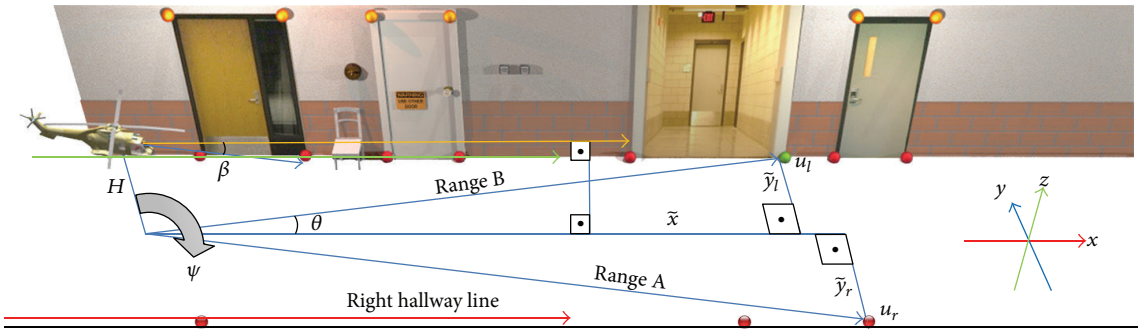

Figure 1: A three-dimensional representation of the corridor with respect to the MAV. Note that the width of the hallway is not provided to the algorithm and the MAV does not have any sensors that can detect walls.

as monoSLAM [10, 11] which depend on movement for depth estimation and offer a relative recovered scale may not provide reliable object avoidance for an agile MAV in an indoor environment. A rotorcraft MAV needs to bank to move the camera sideways, a movement severely limited in a hallway for helicopter dynamics; it has to be able to perform depth measurement from a still, or nearly-still platform.

In SLAM, Extended Kalman Filter based approaches with full covariance have a limitation for the size of a manageable map in real time, considering the quadratic nature of the algorithm versus computational resources of an MAV. Global localization techniques such as Condensation SLAM [12] require a full map to be provided to the robot a-priori. Azimuth learning based techniques such as Cognitive SLAM [13] are parametric, and locations are centered on the robot which naturally becomes incompatible with ambiguous landmarks—such as the landmarks our MAV has to work with. Image registration based methods, such as [14], propose a different formulation of the vision-based SLAM problem based on motion, structure, and illumination parameters without first having to find feature correspondences. For a real-time implementation, however, a local optimization procedure is required, and there is a possibility of getting trapped in a local minimum. Further, without merging regions with a similar structure, the method becomes computationally intensive for an MAV. Structure extraction methods [15] have some limitations, since an incorrect incorporation of points into higher level features will have an adverse effect on consistency. Further, these systems depend on a successful selection of thresholds.

1.2. Comparison with Prior Work and Organization. This paper addresses the above shortcomings using an unmodified consumer-grade monocular web camera. By exploiting the architectural orthogonality of the indoor and urban outdoor environments, we introduce a novel method for monocular vision-based SLAM by computing absolute range and bearing information without using active ranging sensors. More thorough algorithm formulations and newer experimental results with a unique indoor-flying helicopter are discussed in this paper than in our prior conference articles [16–19]. Section 2 explains the procedures for perception of world geometry as pre-requisites for SLAM, such as range measurement methods, as well as performance evaluations of proposed methods. While a visual turn-sensing algorithm is introduced in Section 3, SLAM formulations are provided in Section 4. Results of experimental validation as well as a description of the MAV hardware platform are presented in Section 5. Figure 2 can be used as a guide to sections as well as to the process flow of our proposed method.

2. Problem and Algorithm Formulation

We propose a novel method to estimate the absolute depth of features using a monocular camera as a sole means of navigation. The camera is mounted on the platform with a slight downward tilt. Landmarks are assumed to be stationary. Moving targets are also detected, however, they are not considered as landmarks and therefore ignored by the map. Altitude is measured in real time via the on-board ultrasonic altimeter on our MAV, or in the case of a ground robot it can be provided to the system via various methods depending on where the camera is installed. It is acceptable that the camera translates or tilts with respect to the robot, such as, mounted on a robotic arm, as long as the mount is properly encoded to indicate altitude. We validate our results with a time-varying altitude. The ground is assumed to be relatively flat (no more than 5 degrees of inclination within a 10-meter perimeter). Our algorithm has capability to adapt to inclines if the camera tilt can be controlled; we have equipped some of our test platforms with this capability.

2.1. Landmark Extraction Step I: Feature Extraction. A landmark in the SLAM context is a conspicuous, distinguishing landscape feature marking a location. A minimal landmark can consist of two measurements with respect to robot position: range and bearing. Our landmark extraction strategy is a three step automatic process. All three-steps are performed on a frame, I_t, before moving onto the next frame, $I_t + 1$. The first step involves finding prominent parts of I_t that tend to be more attractive than other parts in terms of texture, dissimilarity, and convergence. These parts tend to be immune to rotation, scale, illumination, and image noise, and we refer to them as features, which have the form $f_n(u, v)$. We utilize two algorithms for this procedure. For flying

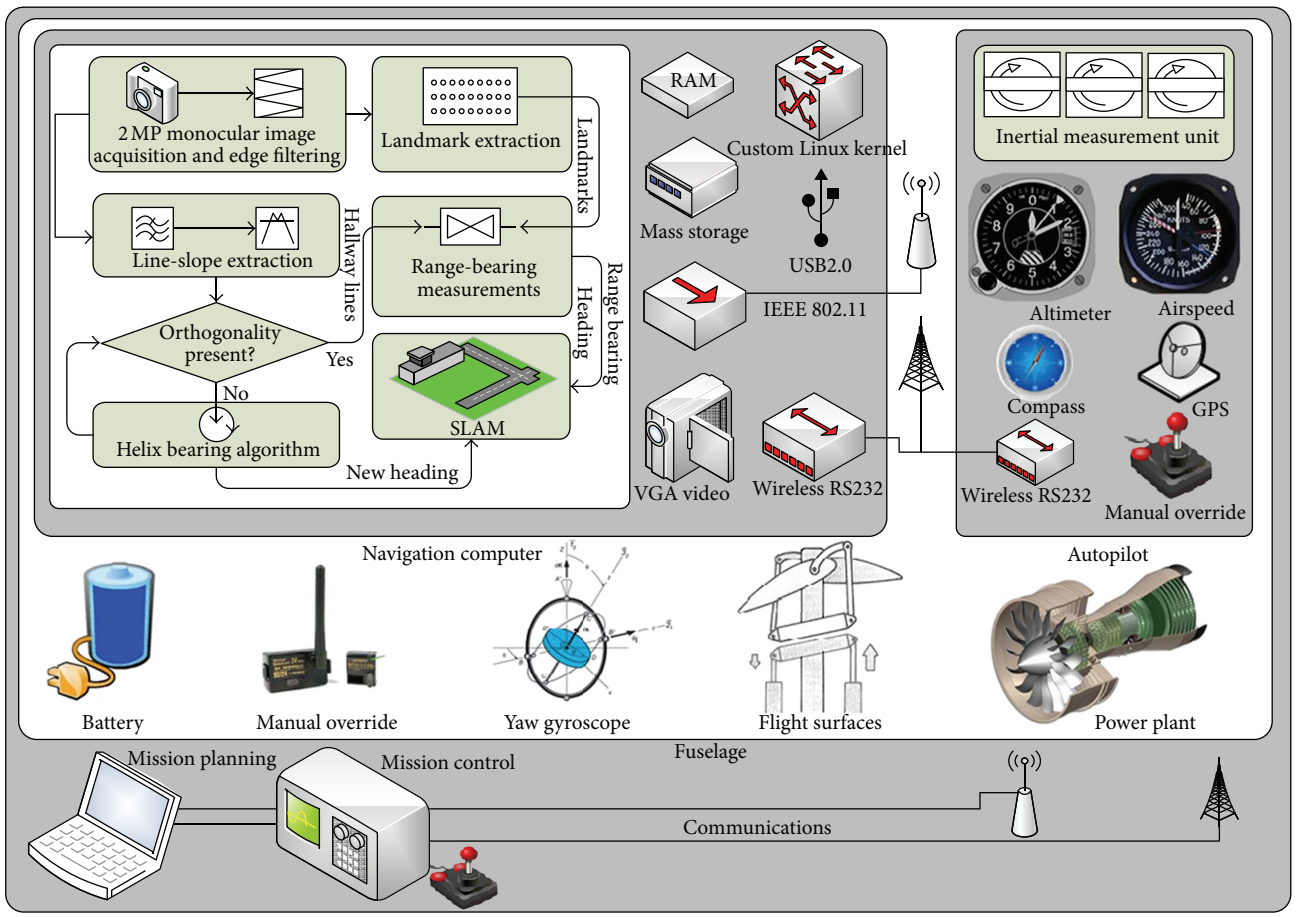

FIGURE 2: Block diagram illustrating the operational steps of the monocular vision navigation and ranging at high level, and its relations with the flight systems. The scheme is directly applicable to other mobile platforms.

platforms, considering the limited computational resources available, we prefer the the algorithm proposed by Shi and Tomasi [20] in which sections of I with large eigenvalues are extracted into a set Ψ such that $\Psi = f_1, f_2, \ldots, f_n$. Although there is virtually no limit for n, it is impossible at this time in the procedure to make an educated distinction between a useless feature for the map (i.e., one that cannot be used for ranging and bearing), and a potential landmark (i.e., one that provides reliable range and bearing information and thus can be included in the map). For ground based platforms, we prefer the SURF algorithm (Figure 3) due to the directionality its detected features offer [21]. Directional features are particularly useful where the platform dynamics are diverse, such as human body, or MAV applications in gusty environments; directional features are more robust in terms of associating them with architectural lines, where instead of a single distance threshold, the direction of feature itself also becomes a metric. It is also useful when ceilings are used where lines are usually segmented and more difficult to detect. This being an expensive algorithm, we consider faster implementations such as ASURF.

In following steps, we describe how to extract a sparse set of reliable landmarks from a populated set of questionable features.

2.2. Landmark Extraction Step II: Line and Slope Extraction. Conceptually, landmarks exist in the 3D inertial frame and they are distinctive, whereas features in $\Psi = f_1, f_2, \ldots, f_n$ exist on a 2D image plane, and they contain ambiguity. In other words, our knowledge of their range and bearing information with respect to the camera is uniformly distributed across I_t. Considering the limited mobility of our platform in the particular environment, parallax among the features is very limited. Thus, we attempt to correlate the contents of Ψ with the real world via their relationship with the perspective lines.

On a well-lit, well-contrasting, noncluttered hallway, perspective lines are obvious. Practical hallways have random objects that segment or even falsely mimic these lines. Moreover, on a monocular camera, objects are aliased with distance making it more difficult to find consistent ends of perspective lines as they tend to be considerably far from the camera. For these reasons, the construction of those lines should be an adaptive approach.

We begin the adaptive procedure by edge filtering the image, I, through a discrete differentiation operator with more weight on the horizontal convolution, such as

$$I'_x = F_h * I, \qquad I'_y = F_v * I, \tag{1}$$

where $*$ denotes the convolution operator, and F is a 3×3 kernel for horizontal and vertical derivative approximations. I'_x and I'_y are combined with weights whose ratio determines the range of angles through which edges will be filtered. This in effect returns a binary image plane, I', with potential edges that are more horizontal than vertical. It is possible to reverse this effect to detect other edges of interest, such as ceiling lines, or door frames. At this point, edges will disintegrate the more vertical they get (see Figure 3 for an illustration). Application of the Hough Transform to I' will return all possible lines, automatically excluding discrete point sets, out of which it is possible to sort out lines with a finite slope $\phi \neq 0$ and curvature $\kappa = 0$. This is a significantly expensive operation (i.e., considering the limited computational resources of an MAV) to perform on a real-time video feed since the transform has to run over the entire frame, including the redundant parts.

To improve the overall performance in terms of efficiency, we have investigated replacing Hough Transform with an algorithm that only runs on parts of I' that contain data. This approach begins by dividing I' into square blocks, $B_{x,y}$. Optimal block size is the smallest block that can still capture the texture elements in I'. Camera resolution and filtering methods used to obtain I' affect the resulting texture element structure. The blocks are sorted to bring the highest number of data points with the lowest entropy (2) first, as this is a block most likely to contain lines. Blocks that are empty, or have a few scattered points in them, are excluded from further analysis. Entropy is the characteristic of an image patch that makes it more ambiguous, by means of disorder in a closed system. This assumes that disorder is more probable than order, and thereby, lower disorder has higher likelihood of containing an architectural feature, such as a line. Entropy can be expressed as

$$-\sum_{x,y} B_{x,y} \log B_{x,y}. \tag{2}$$

The set of *candidate* blocks resulting at this point are to be searched for lines. Although a block B_n is a binary matrix, it can be thought as a coordinate system which contains a set of points (i.e., pixels) with (x, y) coordinates such that positive x is right, and positive y is down. Since we are more interested in lines that are more horizontal than vertical, it is safe to assume that the errors in the y values outweigh those in the x values. Equation for a ground line is in the form $y = mx + b$, and the deviations of data points in the block from this *line* are $d_i = y_i - (mx_i + b)$. Therefore, the most likely *line* is the one that is composed of data points that minimize the deviation such that $d_i^2 = (y_i - mx_i - b)^2$. Using determinants, the deviation can be obtained as in (3)

$$d_i = \begin{vmatrix} \sum \left(x_i^2\right) & \sum x_i \\ \sum x_i & i \end{vmatrix}, \qquad m \times d_i = \begin{vmatrix} \sum \left(x_i \cdot y_i\right) & \sum x_i \\ \sum y_i & i \end{vmatrix},$$

$$b \times d_i = \begin{vmatrix} \sum \left(x_i^2\right) & \sum \left(x_i \cdot y_i\right) \\ \sum x_i & \sum y_i \end{vmatrix}. \tag{3}$$

Since our range measurement methods depend on these lines, the overall line-slope accuracy is affected by the reliability in detecting and measuring the hallway lines (or road lines, sidewalk lines, depending on context). The high measurement noise in slopes has adverse effects on SLAM and should be minimized to prevent inflating the uncertainty in $L_1 = \tan \phi_1$ and $L_2 = \tan \phi_2$ or the infinity point (P_x, P_y). To reduce this noise, lines are cross-validated for the longest collinearity via pixel neighborhood based line extraction, in which the results obtained rely only on a local analysis. Their coherence is further improved using a postprocessing step via exploiting the texture gradient. With an assumption of the orthogonality of the environment, lines from the ground edges are extracted. Note that this is also applicable to ceiling lines. Although ground lines (and ceiling lines, if applicable) are virtually parallel in the real world, on the image plane they intersect. The horizontal coordinate of this intersection point is later used as a heading guide for the MAV, as illustrated in Figure 5. Features that happen to coincide with these lines are potential landmark candidates. When this step is complete, a set of features cross-validated with the perspective lines, Ψ', which is a subset of Ψ with the nonuseful features removed, is passed to the third step.

2.3. Landmark Extraction Step III: Range Measurement by the Infinity-Point Method. This step accurately measures the absolute distance to features in Ψ' by integrating local patches of the ground information into a global surface reference frame. This new method significantly differs from optical flows in that the depth measurement does not require a successive history of images.

Our strategy here assumes that the height of the camera from the ground, H, is known a priori (see Figure 1); MAV provides real-time altitude information to the camera. We also assume that the camera is initially pointed at the general direction of the far end of the corridor. This later assumption is not a requirement; if the camera is pointed at a wall, the system will switch to visual steering mode and attempt to recover camera path without mapping until hallway structure becomes available.

The camera is tilted down (or up, depending on preference) with an angle β to facilitate continuous capture of feature movement across perspective lines. The infinity point, (P_x, P_y), is an imaginary concept where the projections of the two parallel perspective lines appear to intersect on the image plane. Since this intersection point is, in theory, infinitely far from the camera, it should present no parallax in response to the translations of the camera. It does, however, effectively represent the yaw and the pitch of the camera (note the crosshair in Figure 5). Assume that the end points of the perspective lines are $E_{H1} = (l, d, -H)^T$ and $E_{H2} = (l, d - w, -H)^T$ where l is length and w is the width of the hallway, d is the horizontal displacement of the camera from the left wall, and H is the MAV altitude (see Figure 4 for a visual description). The Euler rotation matrix to convert

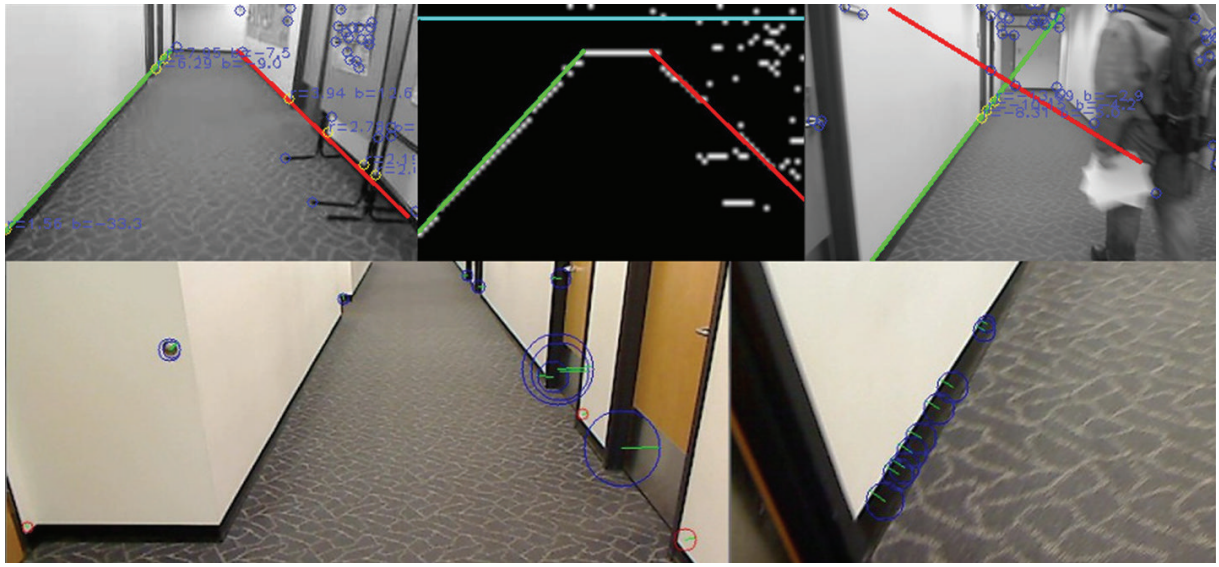

Figure 3: Initial stages after filtering for line extraction, in which the line segments are being formed. Note that the horizontal lines across the image denote the artificial horizon for the MAV; these are not architectural detections, but the on-screen display provided by the MAV. This procedure is robust to transient disturbances such as people walking by or trees occluding the architecture.

from the camera frame to the hallway frame is given in (4):

$$A = \begin{bmatrix} c\psi c\beta & c\beta s\psi & -s\beta \\ c\psi s\phi s\beta - c\phi s\psi & c\phi c\psi + s\phi s\psi s\beta & c\beta s\phi \\ s\phi s\psi + c\phi c\psi s\beta & c\phi s\psi s\beta - c\psi s\phi & c\phi c\beta \end{bmatrix}, \quad (4)$$

where c and s are abbreviations for cos and sin functions, respectively. The vehicle yaw angle is denoted by ψ, the pitch by β, and the roll by ϕ. Since the roll angle is controlled by the onboard autopilot system, it can be set to be zero.

The points E_{H1} and E_{H2} are transformed into the camera frame via multiplication with the transpose of A in (4):

$$E_{C1} = A^T \cdot (l, d, -H)^T, \qquad E_{C2} = A^T \cdot (l, d - w, -H)^T. \quad (5)$$

This 3D system is then transformed into the 2D image plane via

$$u = \frac{yf}{x}, \qquad v = \frac{zf}{x}, \quad (6)$$

where u is the pixel horizontal position from center (right is positive), v is the pixel vertical position from center (up is positive), and f is the focal length (3.7 mm for the particular camera we have used). The end points of the perspective lines have now transformed from E_{H1} and E_{H2} to $(Px_1, Py_1)^T$ and $(Px_2, Py_2)^T$, respectively. An infinitely long hallway can be represented by

$$\lim_{l\to\infty} Px_1 = \lim_{l\to\infty} Px_2 = f\tan\psi,$$
$$\lim_{l\to\infty} Py_1 = \lim_{l\to\infty} Py_2 = -\frac{f\tan\beta}{\cos\psi} \quad (7)$$

which is conceptually the same as extending the perspective lines to infinity. The fact that $Px_1 = Px_2$ and $Py_1 = Py_2$ indicates that the intersection of the lines in the image plane is the end of such an infinitely long hallway. Solving the resulting equations for ψ and β yields the camera yaw and pitch, respectively,

$$\psi = \tan^{-1}\left(\frac{P_x}{f}\right), \qquad \beta = -\tan^{-1}\left(\frac{P_y\cos\psi}{f}\right). \quad (8)$$

A generic form of the transformation from the pixel position, (u, v) to (x, y, z), can be derived in a similar fashion [3]. The equations for u and v also provide general coordinates in the camera frame as $(z_c f/v, uz_c/v, z_c)$ where z_c is the z position of the object in the camera frame. Multiplying with (4) transforms the hallway frame coordinates (x, y, z) into functions of u, v, and z_c. Solving the new z equation for z_c and substituting into the equations for x and y yields

$$\tilde{x} = \left(\frac{(a_{12}u + a_{13}v + a_{11}f)}{(a_{32}u + a_{33}v + a_{31}f)}\right)z,$$
$$\tilde{y} = \left(\frac{(a_{22}u + a_{23}v + a_{21}f)}{(a_{32}u + a_{33}v + a_{31}f)}\right)z, \quad (9)$$

where a_{ij} denotes the elements of the matrix in (4). See Figure 1 for the descriptions of $\tilde{x}$ and $\tilde{y}$.

For objects likely to be on the floor, the height of the camera above the ground is the z position of the object. Also, if the platform roll can be measured, or assumed negligible, then the combination of the infinity point with the height can be used to obtain the range to any object on the floor of the hallway. This same concept applies to objects which are likely to be on the same wall or the ceiling. By exploiting the geometry of the corners present in the corridor, our

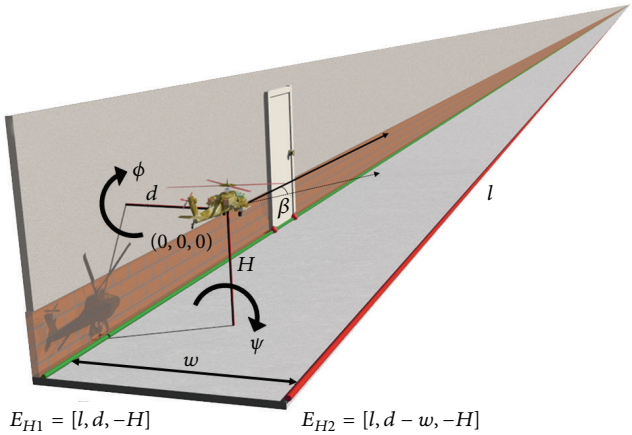

FIGURE 4: A visual description the environment as perceived by the infinity-point method.

method computes the absolute range and bearing of the features, effectively turning them into landmarks needed for the SLAM formulation. See Figure 5 which illustrates the final appearance of the ranging algorithm.

The graph in Figure 6 illustrates the disagreement between the line-perspectives and the infinity-point method (Section 2.3) in an experiment in which both algorithms executed simultaneously on the same video feed. With the particular camera we used in the experiments (Logitech C905), the infinity-point method yielded a 93% accuracy. These numbers are functions of camera resolution, camera noise, and the consequent line extraction noise. Therefore, disagreements not exceeding 0.5 meters are in the favor of it with respect to accuracy. Disagreements from the ground truth include all transient measurement errors such as camera shake, or occasional introduction of moving objects that deceptively mimic the environment and other anomalies. The divergence between the two ranges that is visible between samples 20 and 40 in Figure 6 is caused by a hallway line anomaly from the line extraction process, independent of ranging. In this particular case, both the hallway lines have shifted, causing the infinity point to move left. Horizontal translations of the infinity point have a minimal effect on the measurement performance of the infinity-point method, being one of its main advantages. Refer to Figure 7 for the demonstration of the performance of these algorithms in a wide variety of environments.

The bias between the two measurements shown in Figure 6 is due to shifts in camera calibration parameters in between different experiments. Certain environmental factors have dramatic effects on lens precision, such as acceleration, corrosive atmosphere, acoustic noise, fluid contamination, low pressure, vibration ballistic shock, electromagnetic radiation, temperature, and humidity. Most of those conditions readily occur on an MAV (and most other platforms, including human body) due to parts rotating at high speeds, powerful air currents, static electricity, radio interference, and so on. Autocalibration concept is wide and beyond the scope of this paper. We present a novel mathematical procedure that addresses the issue of maintaining monocular camera calibration automatically in hostile environments in another paper of ours and we encourage the reader to refer to it [22].

3. Helix Bearing Algorithm

When the MAV approaches a turn, an exit, a T-section, or a dead-end, both ground lines tend to disappear simultaneously. Consequently, range and heading measurement methods cease to function. A set of features might still be detected, and the MAV can make a confident estimate of their spatial pose. However, in the absence of depth information, a one-dimensional probability density over the depth is represented by a two-dimensional particle distribution.

In this section, we propose a turn-sensing algorithm to estimate ψ in the absence of orthogonality cues. This situation automatically triggers the turn-exploration mode in the MAV. A yaw rotation of the body frame is initiated until another passage is found. The challenge is to estimate ψ accurately enough to update the SLAM map correctly. This procedure combines machine vision with the data matching and dynamic estimation problem. For instance, if the MAV approaches a left-turn after exploring one leg of an "L" shaped hallway, turns left 90 degrees, and continues through the next leg, the map is expected to display two hallways joined at a 90-degree angle. Similarly, a 180-degree turn before finding another hallway would indicate a dead end. This way, the MAV can also determine where turns are located the next time they are visited.

The new measurement problem at turns is to compute the instantaneous velocity, (u, v) of every helix (moving feature) that the MAV is able to detect as shown in Figure 9. In other words, an attempt is made to recover $V(x, y, t) = (u(x, y, t), (v(x, y, t)) = (dx/dt, dy/dt)$ using a variation of the pyramidal Lucas-Kanade method. This recovery leads to a 2D vector field obtained via perspective projection of the 3D velocity field onto the image plane. At discrete time steps, the next frame is defined as a function of a previous frame as $I_{t+1}(x, y, z, t) = I_t(x + dx, y + dy, z + dz, t + dt)$. By applying the Taylor series expansion,

$$I(x, y, z, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial z}\delta z + \frac{\partial I}{\partial t}\delta t \qquad (10)$$

then by differentiating with respect to time yields, the helix velocity is obtained in terms of pixel distance per time step k.

At this point, each helix is assumed to be identically distributed and independently positioned on the image plane. And each helix is associated with a velocity vector $V_i = (v, \varphi)^T$ where φ is the angular displacement of velocity direction from the north of the image plane where $\pi/2$ is east, π is south, and $3\pi/2$ is west. Although the associated depths of the helix set appearing at stochastic points on the image plane are unknown, assuming a constant $\dot{\psi}$, there is a relationship between distance of a helix from the camera and its instantaneous velocity on the image plane. This suggests that a helix cluster with respect to closeness of individual

(1) Start from level $L(0) = 0$ and sequence $m = 0$
(2) **Find** $d = \min(h_a - h_b)$ in M where $h_a \neq h_b$
(3) $m = m + 1$, $\Psi'''(k) = \text{merge}([h_a, h_b])$, $L(m) = d$
(4) **Delete from** M: rows and columns corresponding to $\Psi'''(k)$
(5) **Add to** M: a row and a column representing $\Psi'''(k)$
(6) if $(\forall h_i \in \Psi'''(k))$, **stop**
(7) else, go to (2)

Algorithm 1: Disjoint cluster identification from heat MAP M.

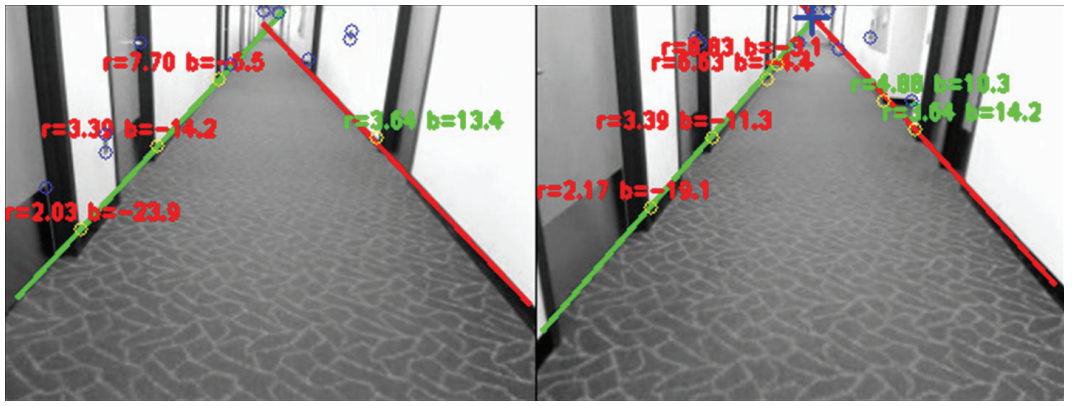

Figure 5: On-the-fly range measurements. Note the crosshair indicating the algorithm is currently using the infinity point for heading.

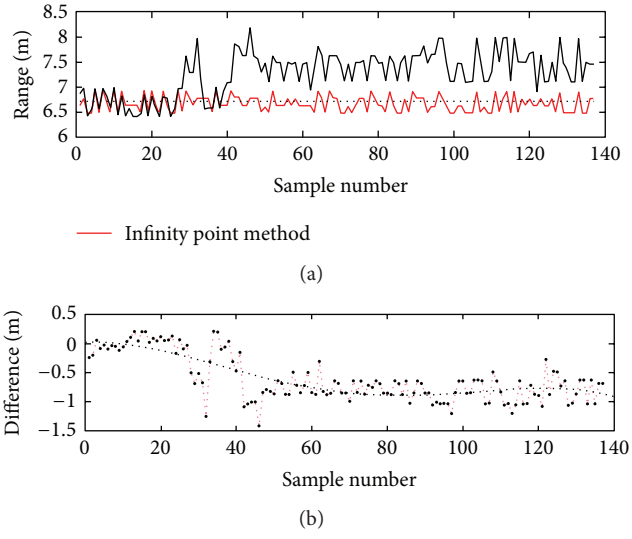

Figure 6: (a) Illustrates the accuracy of the two-range measurement methods with respect to ground truth (flat line). (b) Residuals for the top figure.

instantaneous velocities is likely to belong on the surface of one planar object, such as a door frame. Let a helix with a directional velocity be the triple $h_i = (V_i, u_i, v_i)^T$ where (u_i, v_i) represents the position of this particle on the image plane. At any given time (k); let Ψ be a set containing all these features on the image plane such that $\Psi(k) = \{h_1, h_2, \ldots, h_n\}$. The z component of velocity as obtained in (10) is the determining factor for φ. Since we are most interested in the set of helix in which this component is minimized, $\Psi(k)$ is resampled such that

$$\Psi'(k) = \left\{\forall h_i, \left\{\varphi \approx \frac{\pi}{2}\right\} \cup \left\{\varphi \approx \frac{3\pi}{2}\right\}\right\} \tag{11}$$

sorted in increasing velocity order. $\Psi'(k)$ is then processed through histogram sorting to reveal the modal helix set such that,

$$\Psi''(k) = \max \begin{cases} \text{if } (h_i = h_{i+1}), & \sum_{i=0}^{n} i, \\ \text{else}, & 0. \end{cases} \tag{12}$$

$\Psi''(k)$ is likely to contain clusters that tend to be distributed with respect to objects in the scene, whereas the rest of the initial helix set from $\Psi(k)$ may not fit this model. An agglomerative hierarchical tree T is used to identify the clusters. To construct the tree, $\Psi''(k)$ is heat mapped, represented as a symmetric matrix M, with respect to Manhattan distance between each individual helixes:

$$M = \begin{bmatrix} h_0 - h_0 & \cdots & h_0 - h_n \\ \vdots & \ddots & \vdots \\ h_n - h_0 & \cdots & h_n - h_n \end{bmatrix}. \tag{13}$$

The algorithm to construct the tree from M is given in Algorithm 1.

The tree should be cut at the sequence m such that $m + 1$ does not provide significant benefit in terms of modeling

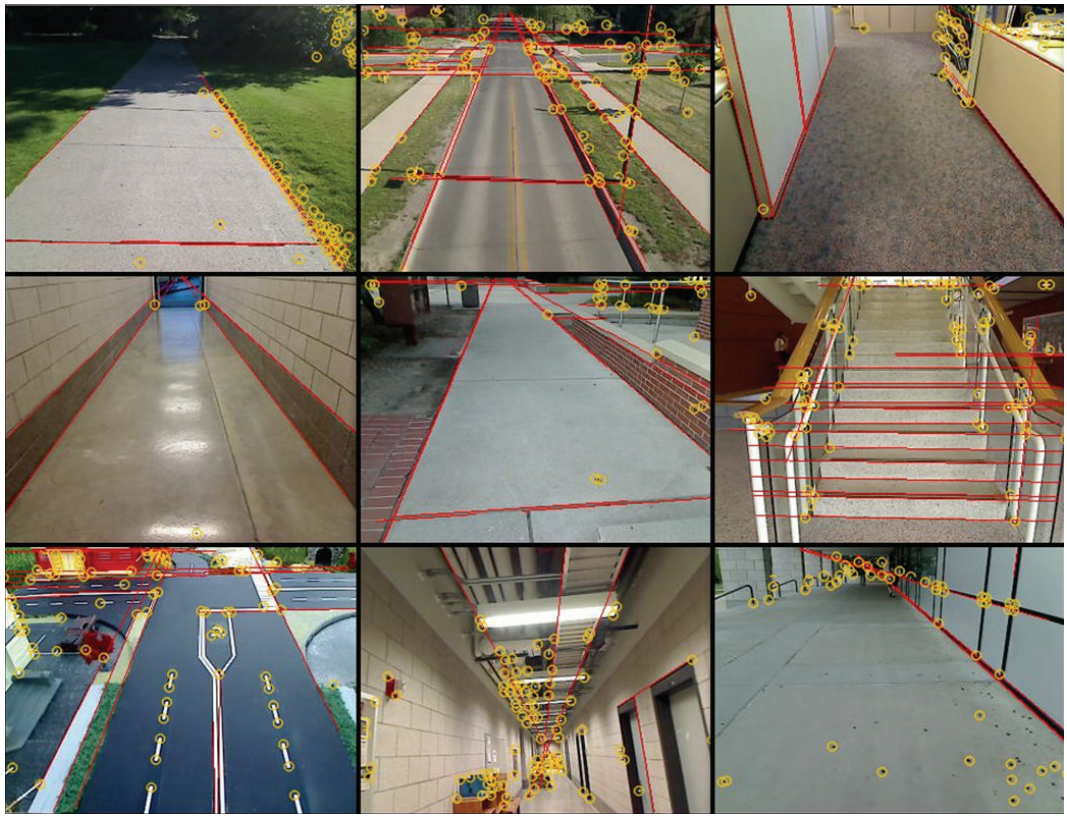

Figure 7: While we emphasize hallway like indoor environments, our range measurement strategy is compatible with a variety of other environments, including outdoors, office environments, ceilings, sidewalks, and building sides, where orthogonality in architecture is present. A minimum of one perspective line and one feature intersection is sufficient.

the clusters. After this step, the set of velocities in $\Psi'''(k)$ represent the largest planar object in the field of view with the most consistent rate of pixel displacement in time. The system is updated such that $\Psi(k+1) = \Psi(k) + \mu(\Psi'''(k))$ as the best effort estimate as shown in Figure 8.

It is a future goal to improve the accuracy of this algorithm by exploiting known properties of typical objects. For instance, single doors are typically a meter-wide. It is trivial to build an internal object database with templates for typical consistent objects found indoors. If such an object of interest could be identified by an arbitrary object detection algorithm, and that world object of known dimensions, $\dim = (x, y)^T$, and a cluster $\Psi'''(k)$ may sufficiently coincide, cluster depth can be measured via $\dim(f/\dim')$ where dim is the actual object dimensions, f is the focal length and $\dim'$ represents object dimensions on image plane.

4. SLAM Formulation

Our previous experiments [16, 17] showed that, due to the highly nonlinear nature of the observation equations, traditional nonlinear observers such as EKF do not scale to SLAM in larger environments containing a vast number of potential landmarks. Measurement updates in EKF require quadratic time complexity due to the covariance matrix, rendering the data association increasingly difficult as the map grows. An MAV with limited computational resources is particularly impacted from this complexity behavior. SLAM utilizing Rao-Blackwellized particle filter similar to [23] is a dynamic Bayesian approach to SLAM, exploiting the conditional independence of measurements. A random set of particles is generated using the noise model and dynamics of the vehicle in which each particle is considered a potential location for the vehicle. A reduced Kalman filter per particle is then associated with each of the current measurements. Considering the limited computational resources of an MAV, maintaining a set of landmarks large enough to allow for accurate motion estimations yet sparse enough so as not to produce a negative impact on the system performance is imperative. The noise model of the measurements along with

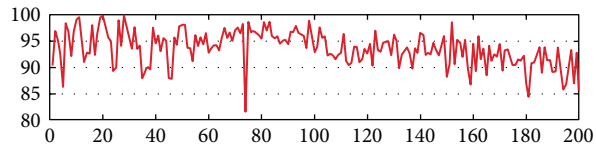

Figure 8: This graph illustrates the accuracy of the Helix bearing algorithm estimating 200 samples of perfect 95 degree turns (calibrated with a digital protractor) performed at various locations with increasing clutter, at random angular rates not exceeding 1 radian-per-second, in the absence of known objects.

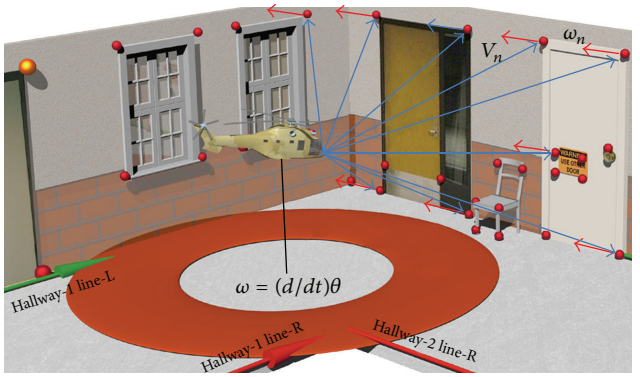

FIGURE 9: The helix bearing algorithm exploits the optical flow field resulting from the features not associated with architectural lines. A reduced helix association set is shown for clarity. Helix velocities that form statistically identifiable clusters indicate the presence of large objects, such as doors, that can provide estimation for the angular rate of the MAV during the turn.

the new measurement and old position of the feature are used to generate a statistical weight. This weight in essence is a measure of how well the landmarks in the previous sensor position correlate with the measured position, taking noise into account. Since each of the particles has a different estimate of the vehicle position resulting in a different perspective for the measurement, each particle is assigned different weights. Particles are resampled every iteration such that the lower weight particles are removed, and higher weight particles are replicated. This results in a cloud of random particles of track towards the best estimation results, which are the positions that yield the best correlation between the previous position of the features and the new measurement data.

The positions of landmarks are stored by the particles such as $\text{Par}_n = (X_L^T, P)$ where $X_L = (x_{ci}, y_{ci})$ and P is the 2×2 covariance matrix for the particular Kalman Filter contained by Par_n. The 6DOF vehicle state vector, x_v, can be updated in discrete time steps of (k) as shown in (14) where $R = (x_r, y_r, H)^T$ is the position in inertial frame, from which the velocity in inertial frame can be derived as $\dot{R} = v_E$. The vector $v_B = (v_x, v_y, v_z)^T$ represents linear velocity of the body frame, and $\omega = (p, q, r)^T$ represents the body angular rate. $\Gamma = (\phi, \theta, \psi)^T$ is the Euler angle vector, and L_{EB} is the Euler angle transformation matrix for (ϕ, θ, ψ). The 3×3 matrix T converts $(p, q, r)^T$ to $(\dot{\phi}, \dot{\theta}, \dot{\psi})$. At every step, the MAV is assumed to experience unknown linear and angular accelerations, $V_B = a_B \Delta t$ and $\Omega = \alpha_B \Delta t$, respectively,

$$x_v(k+1) = \begin{pmatrix} R(k) + L_{EB}(\phi, \theta, \psi)(v_B + V_B)\Delta t \\ \Gamma(k) + T(\phi, \theta, \psi)(\omega + \Omega)\Delta t \\ v_B(k) + V_B \\ \omega(k) + \Omega \end{pmatrix}. \tag{14}$$

There is only a limited set of orientations a helicopter is capable of sustaining in the air at any given time without partial or complete loss of control. For instance, no useful lift is generated when the rotor disc is oriented sideways with respect to gravity. Moreover, the on-board autopilot incorporates IMU and compass measurements in a best-effort scheme to keep the MAV at hover in the absence of external control inputs. Therefore, we can simplify the 6DOF system dynamics to simplified 2D system dynamics with an autopilot. Accordingly, the particle filter then simultaneously locates the landmarks and updates the vehicle states x_r, y_r, θ_r described by

$$\mathbf{x}_v(k+1) = \begin{pmatrix} \cos\theta_r(k)\, u_1(k) + x_r(k) \\ \sin\theta_r(k)\, u_1(k) + y_r(k) \\ u_2(k) + \theta_r(k) \end{pmatrix} + \gamma(k), \tag{15}$$

where $\gamma(k)$ is the linearized input signal noise, $u_1(k)$ is the forward speed, and $u_2(k)$ the angular velocity. Let us consider one instantaneous field of view of the camera, in which the center of two ground corners on opposite walls is shifted. From the distance measurements described earlier, we can derive the relative range and bearing of a corner of interest (index i) as follows

$$\mathbf{y}_i = \mathbf{h}(\mathbf{x}) = \left(\sqrt{\tilde{x}_i^2 + \tilde{y}_i^2},\ \tan^{-1}\left[\pm\frac{\tilde{y}_i}{\tilde{x}_i}\right],\ \psi \right)^T, \tag{16}$$

where ψ measurement is provided by the infinity-point method.

This measurement equation can be related with the states of the vehicle and the ith landmark at each time stamp (k) as shown in (17) where $\mathbf{x}_v(k) = (x_r(k), y_r(k), \theta_r(k))^T$ is the vehicle state vector of the 2D vehicle kinematic model. The measurement equation $\mathbf{h}_i(\mathbf{x}(k))$ can be related with the states of the vehicle and the ith corner (landmark) at each time stamp (k) as given in (17):

$$\mathbf{h}_i(\mathbf{x}(k)) = \begin{pmatrix} \sqrt{(x_r(k) - x_{ci}(k))^2 + (y_r(k) - y_{ci}(k))^2} \\ \tan^{-1}\left(\dfrac{y_r(k) - y_{ci}(k)}{x_r(k) - x_{ci}(k)}\right) - \theta_r(k) \\ \theta_r \end{pmatrix}, \tag{17}$$

where x_{ci} and y_{ci} denote the position of the ith landmark.

4.1. Data Association. Recently detected landmarks need to be associated with the existing landmarks in the map such that each new measurement either corresponds to the correct existent landmark or else registers as a not-before-seen landmark. This is a requirement for any SLAM approach to function properly (i.e., Figure 11). Typically, the association metric depends on the measurement innovation vector. An exhaustive search algorithm that compares every measurement with every feature on the map associates landmarks if the newly measured landmarks is sufficiently close to an existing one. This not only leads to landmark ambiguity but also is

computationally intractable for large maps. Moreover, since the measurement is relative, the error of the vehicle position is additive with the absolute location of the measurement.

We present a new, faster, and more accurate solution, which takes advantage of predicted landmark locations on the image plane. Figure 5 gives a reference of how landmarks appear on the image plane to move along the ground lines as the MAV moves. Assume that $p^k_{(x,y)}$, $k = 0, 1, 2, 3, \ldots, n$ represents a pixel in time which happens to be contained by a landmark, and this pixel moves along a ground line at the velocity v_p. Although landmarks often contain a cluster of pixels size of which is inversely proportional with landmark distance, here the center pixel of a landmark is referred. Given that the expected maximum velocity, $V_{B\max}$, is known, a pixel is expected to appear at

$$p^{k+1}_{(x,y)} = f\left(\left(p^k_{(x,y)} + (v_B + V_B)\,\Delta t\right)\right), \tag{18}$$

where

$$\sqrt{\left(p^{k+1}_{(x)} - p^k_{(x)}\right)^2 + \left(p^{k+1}_{(y)} - p^k_{(y)}\right)^2} \tag{19}$$

cannot be larger than $V_{B\max}/\Delta t$ while $f(\cdot)$ is a function that converts a landmark range to a position on the image plane.

A landmark appearing at time $k + 1$ is to be associated with a landmark that has appeared at time k if and only if their pixel locations are within the association threshold. In other words, the association information from k is used. Otherwise, if the maximum expected change in pixel location is exceeded, the landmark is considered new. We save computational resources by using the association data from k when a match is found, instead of searching the large global map. In addition, since the pixel location of a landmark is independent of the noise in the MAV position, the association has an improved accuracy. To further improve the accuracy, there is also a maximum range beyond which the MAV will not consider for data association. This range is determined taking the camera resolution into consideration. The farther a landmark is, the fewer pixels it has in its cluster, thus the more ambiguity and noise it may contain. Considering the physical camera parameters resolution, shutter speed, and noise model of the Logitech-C905 camera, the MAV is set to ignore landmarks farther than 8 meters. Note that this is a limitation of the camera, not our proposed methods.

Although representing the map as a tree based data structure which, in theory, yields an association time of $O(N \log N)$, our pixel-neighborhood based approach already covers over 90% of the features at any time, therefore a tree based solution does not offer a significant benefit.

We also use a viewing transformation invariant scene matching algorithm based on spatial relationships among objects in the images, and illumination parameters in the scene. This is to determine if two frames acquired under different extrinsic camera parameters have indeed captured the same scene. Therefore, if the MAV visits a particular place more than once, it can distinguish whether it has been to that spot before.

Our approach maps the features (i.e., corners, lines) and illumination parameters from one view in the past to the other in the present via affine-invariant image descriptors. A descriptor D_t consists of an image region in a scene that contains a high amount of disorder. This reduces the probability of finding multiple targets later. The system will pick a region on the image plane with the most crowded cluster of landmarks to look for a descriptor, which is likely to be the part of the image where there is most clutters, hence creating a more unique signature. Descriptor generation is automatic and triggered when turns are encountered (i.e., Helix Bearing Algorithm). A turn is a significant, repeatable event in the life of a map which makes it interesting for data association purposes. The starting of the algorithm is also a significant event, for which the first descriptor D_0 is collected, which helps the MAV in recognizing the starting location if it is revisited.

Every time a descriptor D_t is recorded, it contains the current time t in terms of frame number, the disorderly region $I_{x,y}$ of size $x \times y$, and the estimate of the position and orientation of the MAV at frame t. Thus, every time a turn is encountered, the system can check if it happened before. For instance, if it indeed has happened at time $t = k$ where $t > k$, D_k is compared with that of D_t in terms of descriptor and landmarks, and the map positions of the MAV at times t and k are expected to match closely, else it means the map is diverging in a quantifiable manner.

The comparison formulation can be summarized as

$$R(x, y) = \frac{\sum_{x',y'}\left(T\left(x', y'\right) - I\left(x + x', y + y'\right)\right)^2}{\sqrt{\sum_{x',y'} T(x', y')^2 \cdot \sum_{x',y'} I(x + x', y + y')^2}}, \tag{20}$$

where a perfect match is 0, and poor matches are represented by larger values up to 1. We use this to determine the degree to which two descriptors are related as it represents the fraction of the variation in one descriptor that may be explained by the other. Figure 10 illustrates how this concept works.

5. Experimental Results

As illustrated in Figures 12, 13, and 14, our monocular vision SLAM correctly locates and associates landmarks to the real world. Figure 15 shows the results obtained in an outdoor experiment with urban roads. A 3D map is built by the addition of time-varying altitude and wall positions, as shown in Figure 16. The proposed methods prove robust to transient disturbances, since features inconsistent about their position are removed from the map.

The MAV assumes that it is positioned at $(0, 0, 0)$ Cartesian coordinates at the start of a mission, with the camera pointed at the positive x-axis; therefore, the width of the corridor is represented by the y-axis. At anytime during the mission, a partial map can be requested from the MAV via Internet. The MAV also stores the map and important video frames (i.e., when a new landmark is discovered) on-board for a later retrieval. Video frames are time linked to the map. It is therefore possible to obtain a still image of the surroundings

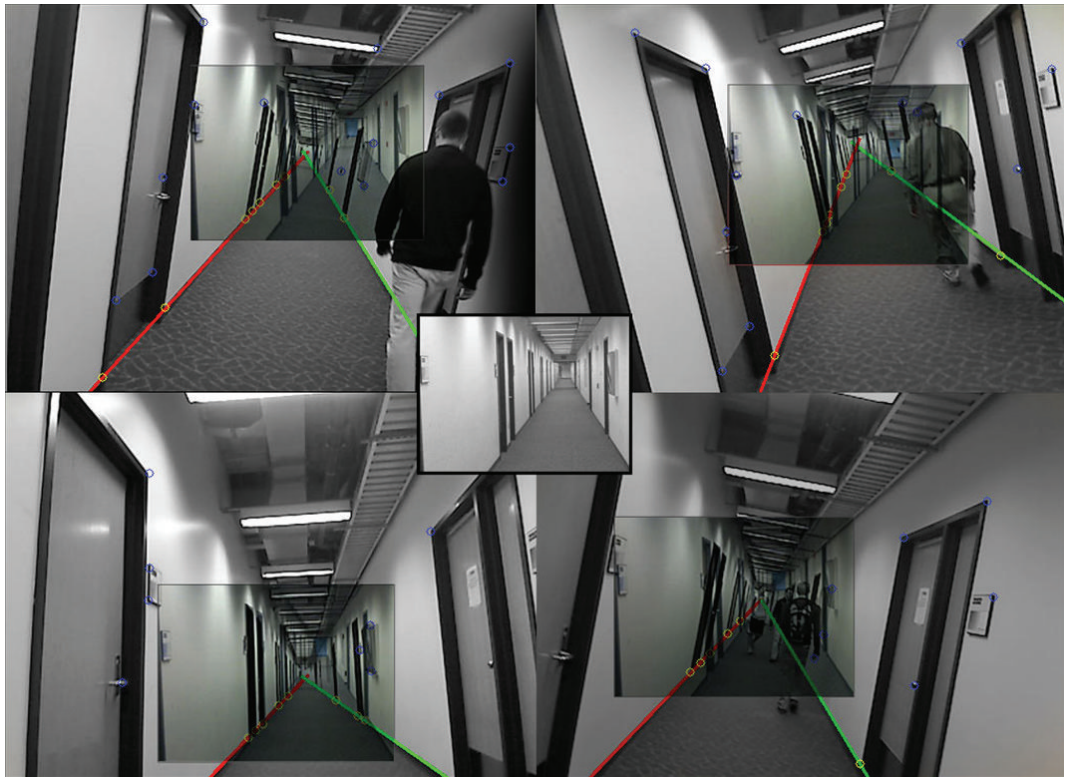

Figure 10: Data association metric where a descriptor is shown on the middle.

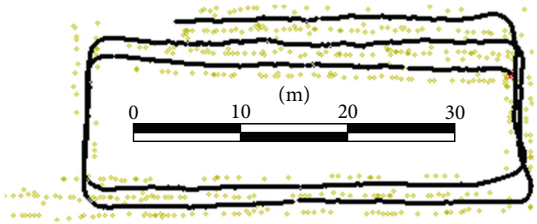

Figure 11: Map drift is one of the classic errors introduced by poor data association, or lack thereof, negatively impacting the loop-closing performance.

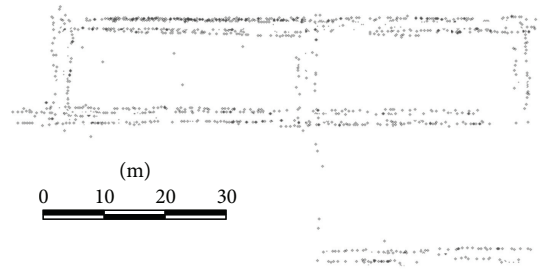

Figure 12: Experimental results of the proposed ranging and SLAM algorithm, showing the landmarks added to the map, representing the structure of the environment. All measurements are in meters. The experiment was conducted under incandescent ambient lightning.

of any landmark for the surveillance and identification purposes.

In Figure 12, the traveled distance is on the kilometer scale. When the system completes the mission and returns to the starting point, the belief is within one meter of where the mission had originally started.

5.1. The Microaerial Vehicle Hardware Configuration. Saint Vertigo, our autonomous MAV helicopter, serves as the primary robotic test platform for the development of this study (see Figure 17). In contrast with other prior works that predominantly used wireless video feeds and Vicon vision tracking system for vehicle state estimation [24], Saint Vertigo performs *all* image processing and SLAM computations onboard, with a 1 GHz CPU, 1 GB RAM, and 4 GB storage. The unit measures 50 cm with a ready-to-fly weight of 0.9 kg and 0.9 kg of payload for adaptability to different missions. In essence, the MAV features two independent computers. The flight computer is responsible for flight stabilization, flight automation, and sensory management. The navigation computer is responsible for image processing, range measurement, SLAM computations, networking, mass storage, and, as a future goal, path planning. The pathway between them is a dedicated on-board link, through which the sensory feedback and supervisory control commands are shared. These commands are simple directives which are converted to the appropriate helicopter flight surface responses by the flight computer. The aircraft is IEEE 802.11 enabled, and all

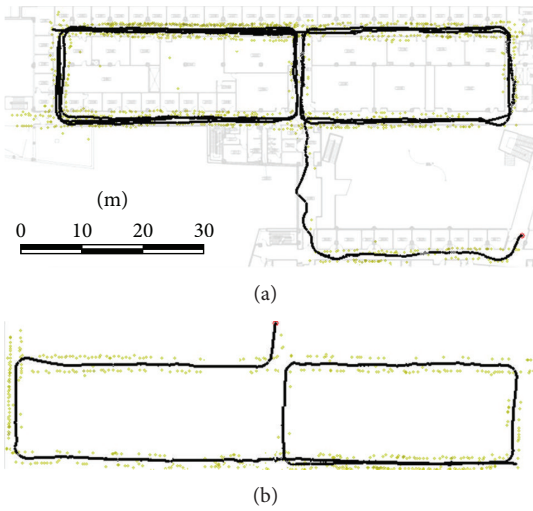

Figure 13: (a) Experimental results of the proposed ranging and SLAM algorithm with state observer odometer trail. Actual floor-plan of the building is superimposed later on a mature map to illustrate the accuracy of our method. Note that the floor plan was not provided to the system a priori. (b) The same environment mapped by a ground robot with a different starting point, to illustrate that our algorithm is compatible with different platforms.

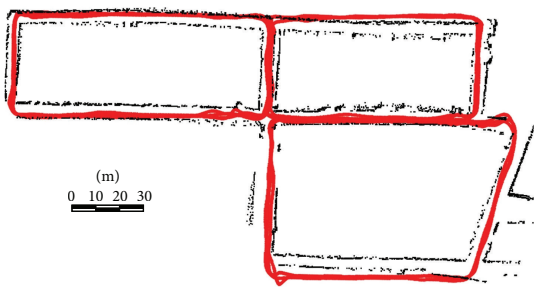

Figure 14: Results of the proposed ranging and SLAM algorithm from a different experiment, with state observer ground truth. All measurements are in meters. The experiment was conducted under fluorescent ambient lightning and sunlight where applicable.

Figure 15: Results of the proposed ranging and SLAM algorithm from an outdoor experiment in an urban area. A small map of the area is provided for reference purposes (not provided to the algorithm) and it indicates the robot path. All measurements are in meters. The experiment was conducted under sunlight ambient conditions and dry weather.

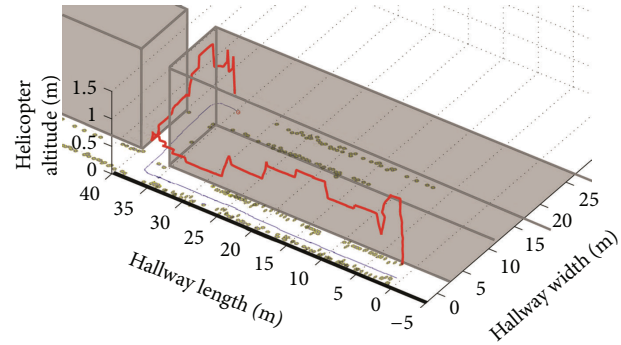

Figure 16: Cartesian (x, y, z) position of the MAV in a hallway as reported by proposed ranging and SLAM algorithm with time-varying altitude. The altitude is represented by the z-axis and it is initially at 25 cm as this is the ground clearance of the ultrasonic altimeter when the aircraft has landed. MAV altitude was intentionally varied by large amounts to demonstrate the robustness of our method to the climb and descent of the aircraft, whereas in a typical mission natural altitude changes are in the range of a few centimeters.

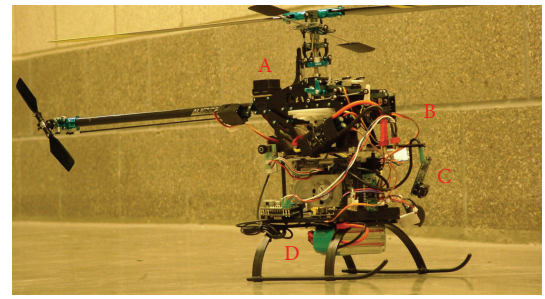

Figure 17: Saint Vertigo, the autonomous MAV helicopter, consists of four decks. The A deck contains collective pitch rotor head mechanics, The B deck comprises the fuselage which houses the power plant, transmission, main batteries, actuators, gyroscope, and the tail rotor. The C deck is the autopilot compartment which contains the inertial measurement unit, all communication systems, and all sensors. The D deck carries the navigation computer which is attached to a digital video camera visible at the front.

its features are accessible over the internet or an ad hoc TCP-IP network. Among the other platforms shown in Figure 18 Saint Vertigo has the most limited computational resources.

5.2. Processing Requirements. In order to effectively manage the computational resources on a light weight MAV computer, we keep track of the CPU utilization for the algorithms proposed in this paper. Table 1 shows a typical breakdown of the average processor utilization per one video frame. Each corresponding task, elucidated in this paper, is visualized in Figure 2.

The numbers in Table 1 are gathered after the map has matured. Methods highlighted with † are mutually exclusive; for example, the Helix Bearing algorithm runs only when the MAV is performing turns, while ranging task is on standby. Particle filtering has a roughly constant load on the system

Figure 18: Our algorithms have been tested on a diverse set of mobile platforms shown here. Picture courtesy of Space Systems and Controls Lab, Aerospace Robotics Lab, Digitalsmithy Lab, and Rockwell Collins Advanced technology Center.

once the map is populated. We only consider a limited point cloud with landmarks in the front detection range of the MAV (see Section 4.1). The MAV typically operates at 80%–90% utilization range. It should be stressed that this numerical figure includes operating system kernel processes which involve video-memory procedures, as the MAV is not equipped with a dedicated graphics processor. The MAV is programmed to construct the SLAM results and other miscellaneous on-screen display information inside the video memory in real time. This is used to monitor the system for our own debugging purposes but not required for the MAV operation. Disabling this feature reduces the load and frees up processor time for other tasks that may be implemented, such as path planning and closed-loop position control.

6. Conclusion and Future Work

In this paper, we investigated the performance of monocular camera based vision SLAM with minimal assumptions, as well as minimal aid from other sensors (altimeter only) in a corridor-following-flight application which requires precise localization and absolute range measurement. This is true even for outdoor cases, because our MAV is capable of building high speeds and covering large distances very rapidly, and some of the ground robots we have tested were large enough to become a concern for traffic and pedestrians. While widely recognized SLAM methods have been mainly developed for use with laser range finders, this paper presented new algorithms for monocular vision-based depth perception and

Table 1: CPU utilization of the proposed algorithms.

Image acquisition and edge filtering	10%
Line and slope extraction	2%
Landmark extraction	20%†
Helix bearing	20%†
Ranging algorithms	Below 1%
Rao-Blackwellized particle filter	50%

bearing sensing to accurately mimic the operation of such an advanced device. We were able to integrate our design with popular SLAM algorithms originally meant for laser range finders, and we have experimentally validated its operation for autonomous indoor and outdoor flight and navigation with a small, fully self-contained MAV helicopter, as well as other robotic platforms. Our algorithms successfully adapt to various situations while successfully performing the transition between (e.g., turns, presence of external objects, and time-varying altitude).

Since the proposed monocular camera vision SLAM method does not need initialization procedures, the mission can start at an arbitrary point. Therefore, our MAV can be deployed to infiltrate an unknown building. One future task is to add the capability to fly through doors and windows. Indeed, the system is only limited by the capabilities of the camera such as resolution, shutter speed, and reaction time. All of those limitations can be overcome with the proper use of lenses and higher fidelity imaging sensors, despite we have used a consumer-grade USB camera. Since the ability to extract good landmarks is a function of the camera capabilities, a purpose-built camera is suggested for future work. Such a camera would also allow development of efficient vision SLAM and data association algorithms that take advantage of the intermediate image processing data.

Our future vision-based SLAM and navigation strategy for an indoor MAV helicopter through hallways of a building also includes the ability to recognize staircases and thus traverse multiple floors to generate a comprehensive volumetric map of the building. This will also permit vision-based 3D path planning and closed-loop position control of MAV based on SLAM. Considering our MAV helicopter is capable of outdoor flight, we can extend our method to the outdoor perimeter of buildings and similar urban environments by exploiting the similarities between hallways and downtown city maps. Further, considering the reduction in weight and independence from GPS coverage, our work also permits the development of portable navigation devices for a wider array of applications such as small-scale mobile robotics and helmet or vest mounted navigation systems.

Certain environments and environmental factors prove challenging to our proposed method: bright lights, reflective surfaces, haze, and shadows. These artifacts introduce two main problems; (1) they can alter chromatic clarity, local microcontrast, and exposure due to their unpredictable high-energy nature, and (2) they can appear as false objects, especially when there is bloom surrounding objects in front of problem light source. Further reduction in contrast is possible if scattering particles in the air are dense. We have come to observe that preventative and defensive approaches to such issues are promising. Antireflective treatment on lenses can reduce light bouncing off of the lens, and programming the aircraft to move for a very small distance upon detection of glare can eliminate the unwanted effects. Innovative and adaptive application of servo-controlled filters before the lenses can minimize or eliminate most, if not, all reflections. The light that causes glare is elliptically polarized due to strong phase correlation. This is as opposed to essential light which is circularly polarized. Filters can detect and block polarized light from entering the camera thereby blocking unwanted effects. Application of purpose designed digital imaging sensors that do not involve a Bayes filter can also help. Most of the glare occurs in green light region and traditional digital imaging sensors have twice as many green receptors as red and blue. Bayes design has been inspired from human eye, which sees green better, as green is the most structurally descriptive light for edges and corners. This paper has supplementary material (see Supplementary Material available online at http://dx.doi.org/10.1155/2013/374165) available from the authors which show experimental results of the paper.

Acknowledgments

The research reported in this paper was in part supported by the National Science Foundation (Grant ECCS-0428040), Information Infrastructure Institute (I^3), Department of Aerospace Engineering and Virtual Reality Application Center at Iowa State University, Rockwell Collins, and Air Force Office of Scientific Research.

References

[1] D. H. Hubel, and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, pp. 106–154, 1962.

[2] N. Isoda, K. Terada, S. Oe, and K. IKaida, "Improvement of accuracy for distance measurement method by using movable CCD," in *Proceedings of the 36th SICE Annual Conference (SICE '97)*, pp. 29–31, Tokushima, Japan, July 1997.

[3] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2nd edition, 2003.

[4] F. Ruffier and N. Franceschini, "Visually guided micro-aerial vehicle: automatic take off, terrain following, landing and wind reaction," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2339–2346, New Orleans, Lo, USA, May 2004.

[5] F. Ruffier, S. Viollet, S. Amic, and N. Franceschini, "Bio-inspired optical flow circuits for the visual guidance of micro-air vehicles," in *Proceedings of the International Symposium on Circuits and Systems (ISCAS '03)*, vol. 3, pp. 846–849, Bangkok, Thailand, May 2003.

[6] J. Michels, A. Saxena, and A. Y. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," in *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*, vol. 119, pp. 593–600, August 2005.

[7] A. Saxena, J. Schulte, and A. Y. Ng, "Depth estimation using monocular and stereo cues," in *Proceedings of the 20th international joint conference on Artifical intelligence (IJCAI '07)*, pp. 2197–2203, 2007.

[8] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3D," *ACM Transactions on Graphics*, vol. 25, no. 3, 2006.

[9] A. W. Fitzgibbon and A. Zisserman, "Automatic camera recovery for closed or open image sequences," in *Proceedings of the European Conference on Computer Vision*, pp. 311–326, June 1998.

[10] A. Davison, M. Nicholas, and S. Olivier, "MonoSLAM: real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[11] L. Clemente, A. Davison, I. Reid, J. Neira, and J. Tardos, "Mapping large loops with a single hand-held camera," in *Proceedings of the Robotics: Science and Systems Conference*, June 2007.

[12] F. Dellaert, W. Burgard, D. Fox, and S. Thrun, "Using the condensation algorithm for robust, vision-based mobile robot localization," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '99)*, pp. 588–594, June 1999.

[13] N. Cuperlier, M. Quoy, P. Gaussier, and C. Giovanangeli, "Navigation and planning in an unknown environment using vision and a cognitive map," in *Proceedings of the IJCAI Workshop, Reasoning with Uncertainty in Robotics*, 2005.

[14] G. Silveira, E. Malis, and P. Rives, "An efficient direct approach to visual SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 969–979, 2008.

[15] A. P. Gee, D. Chekhlov, A. Calway, and W. Mayol-Cuevas, "Discovering higher level structure in visual SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 980–990, 2008.

[16] K. Çelik, S.-J. Chung, and A. K. Somani, "Mono-vision corner SLAM for indoor navigation," in *Proceedings of the IEEE International Conference on Electro/Information Technology (EIT '08)*, pp. 343–348, Ames, Iowa, USA, May 2008.

[17] K. Çelik, S.-J. Chung, and A. K. Somani, "MVCSLAM: mono-vision corner SLAM for autonomous micro-helicopters in GPS denied environments," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Honolulu, Hawaii, USA, August 2008.

[18] K. Çelik, S. J. Chung, and A. K. Somani, "Biologically inspired monocular vision based navigation and mapping in GPS-denied environments," in *Proceedings of the AIAA Infotech at Aerospace Conference and Exhibit and AIAA Unmanned...Unlimited Conference*, Seattle, Wash, USA, April 2009.

[19] K. Çelik, S.-J. Chung, M. Clausman, and A. K. Somani, "Monocular vision SLAM for indoor aerial vehicles," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, St Louis, Mo, USA, October 2009.

[20] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 593–600, June 1994.

[21] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[22] K. Çelik and A. K. Somani, "Wandless realtime autocalibration of tactical monocular cameras," in *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV '12)*, Las Vegas, Nev, USA, 2012.

[23] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: a factored solution to the simultaneous localization and mapping problem," in *Proceedings of the AAAI National Conference on Artificial Intelligence*, pp. 593–598, 2002.

[24] J. P. How, B. Bethke, A. Frank, D. Dale, and J. Vian, "Real-time indoor autonnomous vehicle test environment," *IEEE Control Systems Magazine*, vol. 28, no. 2, pp. 51–64, 2008.