Vistas and Wall-Floor Intersection Features: Enabling Autonomous Flight in Man-made Environments

Kyel Ok, Duy-Nguyen Ta and Frank Dellaert

***Abstract*— We propose a solution toward the problem of autonomous flight and exploration in man-made indoor environments with a micro aerial vehicle (MAV), using a frontal camera, a downward-facing sonar, and an IMU. We present a general method to detect and steer an MAV toward distant features that we call *vistas* while building a map of the environment to detect unexplored regions. Our method enables autonomous exploration capabilities while working reliably in textureless indoor environments that are challenging for traditional monocular SLAM approaches. We overcome the difficulties faced by traditional approaches with *Wall-Floor Intersection Features*, a novel type of low-dimensional landmarks that are specifically designed for man-made environments to capture the geometric structure of the scene. We demonstrate our results on a small, commercially available quadrotor platform.**

I. Introduction

We address the problem of vision-based autonomous navigation and exploration in man-made environments for Micro Aerial Vehicles (MAVs). With its wide range of applications in military and civilian services, research in autonomous navigation and exploration for MAVs has been growing significantly in recent years. Despite many similar characteristics to ground robots, problems such as autonomous navigation, obstacle avoidance, and map building on an aerial robot have been much more challenging due to payload limitations, power availability, and extra degrees-of-freedom.

Recent work in autonomous MAV navigation and exploration has been insufficient due to aforementioned challenges. Related work, described in Section II, either neglects to address the power and payload limitations by using heavy and power-hungry sensors or uses vision-only but comes short of achieving autonomous exploration capabilities.

We present an autonomous navigation and exploration method, using a lightweight frontal camera, an IMU and a downward-facing sonar for height measurements. Our key contribution is combining map-building and detection of distant features, which we call *vistas,* to enable exploration strategies that could not be achieved before. For example, we utilize our map of inferred structure to detect unexplored regions of interest, such as new hallway openings. This type of capability could not be achieved in previous vision-based MAVs, without dedicating additional sensors for this purpose (i.e. frontal and side-facing sonars [1]).

Our first contribution is using vistas to determine the robot steering direction, enabling robust navigation. Our vistas are derived from first principles of what it means to be *distant*; hence, they are not hallway-specific like the previous work that depends on vanishing points detected from spurious edges [1] or hallway-specific cues [2]. Moreover, vistas are also derived from scale-space features and inherit the properties such that they are easily and reliably detected and tracked in many types of environments.

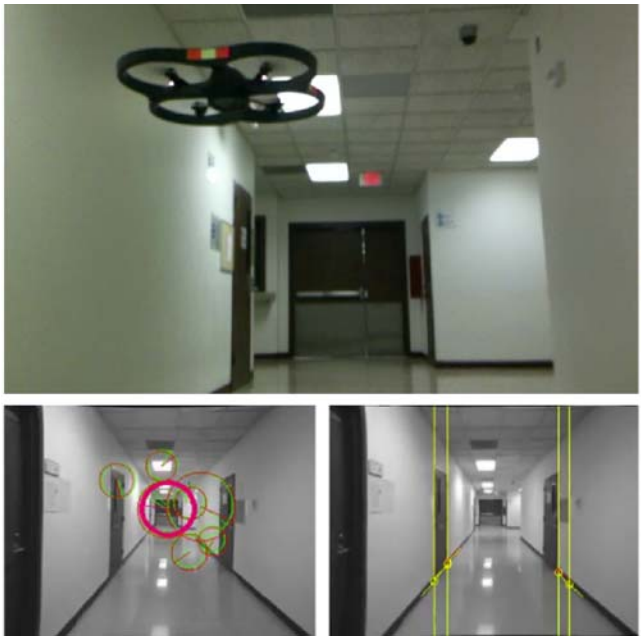


Fig. 1: Our method uses *vistas* (bottom left) to maintain long-term orientation consistency and relies on a map of *Wall-Floor Intersection Features* (bottom right) to infer the scene structure. We present our results in an indoor setting using a Parrot AR.Drone (top).

Our second contribution is an indoor mapping paradigm that allows full exploration. In addition to vistas, for intelligent exploration schemes, the MAV needs some knowledge about the scene structure. We infer the structure from a map of compact and low-dimensional landmarks that are informative enough to capture the most important geometric information about the scene. Our novel landmarks that we call *Wall-Floor Intersection Features* lie on the perpendicular intersection of vertical lines on the wall and the horizontal floor plane. They encode the direction of the wall and can capture any type of corners whether straight, convex or concave. We build a map of our landmarks online using the state-of-the-art inferencing engine, iSAM2 [3].

We combine our contributions to demonstrate an autonomous exploration system on an inexpensive quadrotor.

The authors are with the Center for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, Georgia, USA,
{kyelok, duynguyen, dellaert}@gatech.edu

II. Related Work

Recent work [4], [5], [6] successfully demonstrates MAV navigation and exploration in indoor environments using a map built with laser scanners. [7] present a full SLAM solution for an MAV equipped with a laser scanner to autonomously navigate in indoor environments. [8] presents a helicopter navigating with a laser scanner to avoid different types of objects such as buildings, trees, and 6mm wires in the city. However, these methods are severely limited to short-term operations due to their heavy payload and high power usage. Moreover, active sensors such as laser scanners are undesirable in many applications (e.g., military), due to the risk of cross-talk and ineligibility for covert operations. Therefore, we preclude the use of laser scanner and other heavy and power-hungry sensors.

Recent work in vision-based autonomous navigation neglects to provide exploration capabilities enabled by building a map of the environment. For example, [1] detects the vanishing point at the end of the hallway by finding intersection of long lines along the corridors. Similarly, on a ground robot, [2] fuses many specific properties present at the end of hallways such as high entropy, symmetry, self-similarity, etc. to infer the hallway directions. However, neither methods have a vision-based exploration capability to steer the robot toward undiscovered regions. [1] attempts to solve the problem but relies on supplementary sonar sensors to detect openings to the sides. However, this method does not infer the scene structure and cannot support any planning algorithms to efficiently explore the area, whereas our combined method can support any planning algorithm to navigate toward unexplored regions.

On the other hand, state-of-the-art map-building methods are insufficient for usage in indoor navigation. Some work relies on a downward camera for building a map [9], [10], [11] but lacks the ability to avoid obstacles. Many other vision-based methods build 3D point-cloud based maps [9], [10], [11] but in textureless indoor environments, the point-clouds are too sparse to reveal the 3D structure needed for path/motion planning. Although some [12], [13] build a map from edges in the environment, they neglect to infer the environment structure crucial for robot navigation. Furthermore, state-of-the-art vision-based methods that reconstruct the indoor scene [14], [15], [16] either rely on the indoor Manhattan world assumption or require expensive multi-hypothesis inference methods [15], [16]. Our method based on Wall-Floor Intersection Features improves on previous work with the ability to work in textureless environments, using sparse yet informative scene representation, and not relying on the indoor Manhattan world assumption.

III. Autonomous Navigation Toward Vistas

One of the first tasks in autonomous navigation and exploration is to determine the direction toward open space. In this section, we derive from first principles a general approach that can potentially be applied to any type of environment to steer the robot.

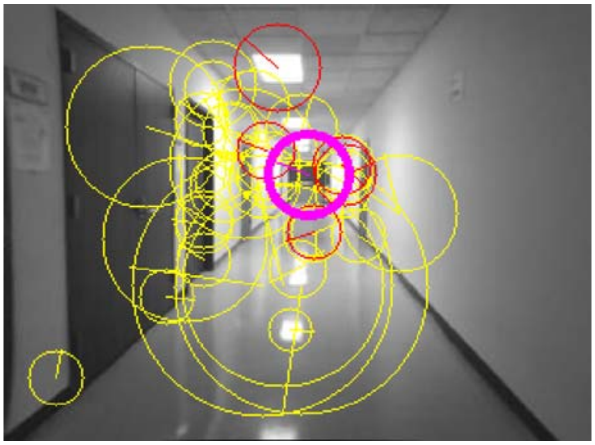


Fig. 2: Detected *vistas* (in red) and features that do not satisfy the *vista* criteria (in yellow) are shown. The closest *vista* to the mean of all the detected *vistas* (pink feature) is selected as the steering direction for the robot.

A. Vista Size Change Criterion

We use *vistas* to refer to those landmarks that are far away from the robot and can be used as a steering direction toward empty space when exploring in an unknown environment.

One important property of vistas is that, due to their far distance to the robot, the size of their projection in the camera frame does not change significantly when flying toward them. This property is already well-known in perceptual psychology under the τ-theory [17] by David Lee, saying that the time-to-collision (TTC) to an object is the ratio τ of the object's image size to the rate of its size change. Some work has utilized this property to compute TTC using optical flow [18], [19] or direct methods [20], [2].

Using this property, we derive vistas from relative size change of scale-space features such as SIFT [21] or SURF [22]. The optimal size of these features are computed by fitting a 3D quadratic function to the feature responses in scale-space around the max response [21].

Let s_1, s_2 be feature sizes and Z_1, Z_2 be their distance from the camera at frames 1 and 2. Since $s_i = f\frac{S}{Z_i}$, where f is the camera focal length and S is the true size of landmark, we have $s_1/s_2 = Z_2/Z_1$. It can be easily shown that $\frac{\Delta s}{s_2} = -\frac{\Delta Z}{Z_1} = \frac{t_z}{Z_1}$ where $\Delta s = s_2 - s_1$ is the absolute size change of the feature and $t_z = -\Delta Z = Z_1 - Z_2$ is the amount of forward movement of the robot between two frames, easily obtained from integrating an IMU, using a motion model, or fusing optical flow and corner tracking on a bottom-looking camera, as already implemented on the AR.Drone [23].

Let Z_{1min} be the minimum safety distance to the landmark in camera frame 1 so that any landmarks with $Z_1 \geq Z_{1min}$ can be considered vistas. The relative size change of vistas must satisfy

$$\frac{\Delta s}{s_2} \leq \frac{t_z}{Z_{1min}} \tag{1}$$

As shown in Figure 2, this criterion leads to a simple yet efficient way to detect distant landmarks in the environment.

B. Vista Rotation-predictability Criterion

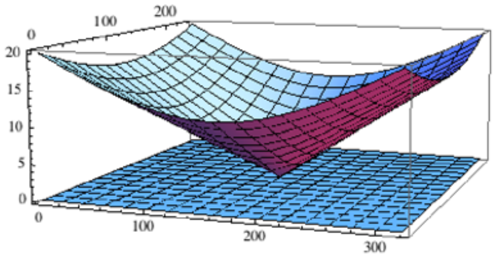


Fig. 3: Minimum Z^r_{1min} distances for rotation-predictable features for $t_x = t_y = 0,\ t_z = 0.1$. The horizontal xy-plane is the image pixel coordinate, and the vertical z-axis is the minimum Z_1 required at each pixel. Plot with camera calibration: $o_x = 160,\ o_y = 120,\ f_x = f_y = 210$.

The minimum safety distance Z_{1min} of vistas in the previous section could be chosen arbitrarily as long as it is safe for the robot to avoid collision with the wall at the moving speed. However, to ease the prediction and tracking of the vistas, we enforce another geometric property of distant landmarks that their projection in the image should be predictable using pure camera rotation, unaffected by the translation. We call this "rotation-predictability" criterion.

We derive this additional requirement for our vistas basing on a well-known fact that if a point is at infinity, its projection in the camera image can be purely determined by the camera rotation. In our case, the camera translation between two consecutive frames is insignificant compared to the distance from the camera to the landmarks, hence has no effect on the landmark position in the image.

More specifically, let p_1 and p_2 be the 2D homogeneous forms of the landmark projections in camera frames 1 and 2. Also, let $K = \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$ be the camera calibration matrix, and $X^1_2 = \{R, t\} \in \mathbb{SE}(3)$ be the odometry of the camera from frame 1 to frame 2. If the landmark P is at infinity or if the camera motion is under a pure rotation ($t = 0$), its projections p_1 and p_2 are related by the infinite homography $H = KR^2_1K^1$ between the two images [24]:

$$p_2 = p^r_2 \sim KR^2_1K^{-1}p_1,$$

where $R^2_1 = R^\top$, and $\sim$ denotes the equivalent up to a constant factor.

However, if the camera motion also involves a translation, i.e. $t \neq 0$, and the landmark is not at infinity, the relationship between p_1 and p_2 is:

$$\begin{aligned} p_2 = p^t_2 &\sim K(R^2_1 Z_1 K^{-1}p_1 + t^2_1) \\ &\sim p^r_2 + \frac{1}{Z_1}Kt^2_1, \end{aligned}$$

where $t^2_1 = -R^\top t$.

Consequently, the rotation-predictability criterion infers that p^t_2 must be well approximated by p^r_2. In this case, the effect of the camera translation t on p_2 is negligible and insensible by the camera, i.e., in homogeneous form, $\frac{1}{Z_1}Kt^2_1 \approx kp^r_2$, for some scalar $k \in \mathbb{R}$. To satisfy this constraint, we impose the condition that the non-homogeneous distance between p^t_2 and p^r_2 has to be less than 1 pixel, i.e.,

$$||\frac{1}{z_{p^r_2}}p^r_2 - \frac{1}{z_{p^t_2}}p^t_2||^2 \leq 1,$$

where $z_{p^r_2}$ and $z_{p^t_2}$ are the third components of p^r_2 and p^t_2, respectively. Solving for this constraint leads to the minimum depth Z^r_{1min} of the landmark such that its image projection can be purely determined by the camera rotation as follows:

$$\begin{aligned} &Z^r_{1min}(x, y, t) = \\ &t_z + \sqrt{[f_x t_x + t_z(o_x - x)]^2 + [f_y t_y + t_z(o_y - y)]^2} \quad (2) \end{aligned}$$

where $t = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^\top$, and (x, y) is the non-homogeneous coordinate of p_1.

This formula shows that the minimum Z^r_{1min} distance of the landmark in the first camera view depends on its position (x, y) in the first image, and also the camera translation t. Figure 3 shows the Z^r_{1min} required for each pixel landmark location in the image where the camera moves in z direction.

Note that at the Focus of Expansion (FoE), where the camera translation vector intersects with the camera image plane, the minimum Z^r_{1min} is very close to the camera. As a trivial example, when the camera moves forward without rotation, $R = I_{3\times3}$, the minimum distance for rotation-predictability criterion is $Z^r_{1min} = t_z$; i.e., any point along the camera optical axis will not be affected by the camera translation as long as it is in front of the second camera view.

Although such limitations exist in the FoE region, the rotation-predictability criterion is still useful to reject false vistas outside the region. Thus, we use $\max(Z^r_{1min}, Z_{1min})$ for the minimum distance in equation (1) to create the final criteria to track vistas on a frame to frame basis.

IV. Wall-Floor Intersection Features for Smoothing and Mapping

Despite vistas' ability to steer a robot toward open space, vistas alone can not grant fully autonomous exploration capabilities. In order to detect directions toward unexplored regions and adopt an intelligent planning scheme, it is critical to obtain a map of the environment. In other words, while flying toward vistas, we need to build a map of landmarks that contains sufficient information about the environment to simultaneously localize the robot and plan exploration strategies. Although, the well-studied problem of Simultaneous Localization and Mapping can be solved using state-of-the-art incremental smoothing and mapping algorithms such as iSAM2 [3], the problem of lack of texture in indoor environments still imposes difficulties in the landmark representation. We address this problem with our novel landmarks, *Wall-Floor Intersection Features*.

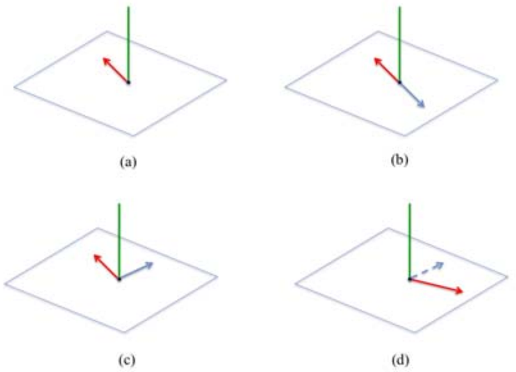


Fig. 4: (a) Our landmark encodes a vertical line position and a wall direction. (b) Two landmarks with opposite wall directions can share the same vertical line. (c) Two landmarks encoding an edge. (d) Two landmarks, one invisible.

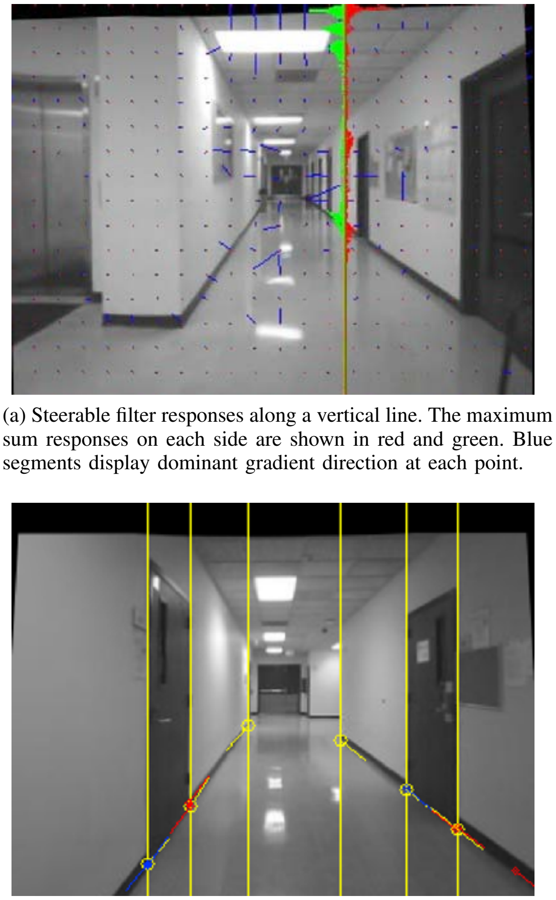
A. *Wall-Floor Intersection Features*

Choosing the right type of landmarks is challenging for indoor vision-based SLAM, due to the textureless scene. [25] proposes to recognize the floor-wall boundary in each column of the input image. [15], on the other hand, categorizes all possible types of corners in indoor environments to generates hypotheses of the environment structure. Recently, [16] generates and evaluates multiple hypotheses of wall-floor intersection lines from detected edges in the images, whereas [14] utilizes the floor-ceiling planar homology.

Inspired by these previous work, we propose a landmark representation that can encode the intersection of a vertical line on the wall and the intersecting floor plane. These new landmarks, named *Wall-Floor Intersection Feature,* are derived from our observation that a vertical line in the scene is most likely associated with a wall and an intersecting floor plane, whose location is estimated by the downward sonar sensor, allowing easy localization of the landmark in space.

Our landmark representation, shown in Figure 4, can employ different wall configurations by encoding only **a single wall direction** in each landmark and allowing two landmarks with different wall directions to co-exist at the same vertical line. This alleviates the need to explicitly model all types of concave/convex corners as previously done in [15], and can deal with non-right wall angles by allowing arbitrary angles between wall directions at the same vertical edge. For example, if a vertical line is on a single wall (ie. vertical edge of a door), the angle between the landmarks would be 180° and if the line is an intersection of two different walls (ie. corners), then the two landmarks will form an angle other than 180°. As such, our representation can efficiently capture the structure of the scene.

Requiring only a 2D position and a single direction, our landmarks can be represented as $\mathbb{SE}(2)$, an element of the Lie-group, where the representation is compact and standard Gauss-Newton optimization is straightforward. Moreover, our landmarks are also easy to detect for both vertical lines and wall directions, as discussed in the next section.



(a) Steerable filter responses along a vertical line. The maximum sum responses on each side are shown in red and green. Blue segments display dominant gradient direction at each point.

(b) Wall-floor corner detection. Detected features are shown in yellow and images of landmarks are shown in red (positive horizontal gradient) and blue (negative horizontal gradient).

Fig. 5: Detection results

B. *Detection and Measurement Model*

First, to detect the vertical line in the landmark, we rectify the image using an IMU, so that vertical lines in the 3D space are also vertical lines in the image, as shown in Figure 5. Then, the vertical line candidates are local maxima in the sum of horizontal image gradients $\mathcal{I}_x$ along each column of the image. Using height estimate from the sonar sensor, each point on the vertical line in the image is associated with one point on the floor plane by back-projection. Then, we only select points with high vertical image gradients $\mathcal{I}_y$ on the bottom half of the image, near the floor.

Then, we detect wall directions for the remaining candidates by (1) quantizing all possible directions on the left and right side of the detected vertical line, (2) summing up steerable filter responses [26] at every pixel along each bin direction and (3) choosing the directions with maximum sum responses on each side (see Figure 5).

Finally, we traverse the image in the detected wall directions as far as the steerable filter response is similar to the original detection. When the response differs by more than a threshold, we stop and store the length of the wall-floor intersection traversed. We finally choose features with lengths larger than a threshold as our landmarks.

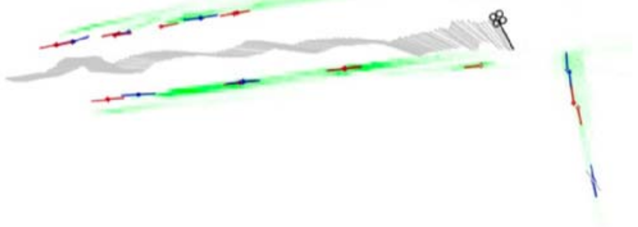


Fig. 6: Wall inference results (green) and an estimated map of Wall-Floor Intersection Features (red and blue).

C. Wall Inference

Our landmarks only capture local information about the wall structure. At places where there are no vertical lines on the wall, no landmarks exist. However, our Wall-Floor Intersection Features are capable of revealing the skeleton structure of the hallway. We perform an additional step to "fill in" the space between the features to yield a complete knowledge of the environment by accumulating evidence of walls in an occupancy grid, with each cell's evidence calculated by extending our features in the wall directions and summing up the image gradient strength along the extended direction. Figure 6 shows our inferred wall structure in the occupancy grid when the drone is turning a corner.

V. Experiments

A. Complete System

In order to obtain an autonomous system, we combine vistas and Wall-Floor Intersection Features with two additional supplementary navigation strategies.

Vistas: We use vistas to choose the steering direction toward open space. This governs the yaw direction of the robot and prevents head-on collision with obstacles.

Wall-Floor Intersection Features: Using iSAM2 [3] Smoothing and Mapping algorithm and our novel landmarks, we create a sparse map of our features and a grid map of inferred wall structure to be used in the later strategies.

Avoiding Side Collisions: Given the local occupancy grid centered at the current robot position, we infer the distance to the walls on the sides of the robot by attempting to equate the distance to the left and the right by changing the MAV's roll rates. This strategy prevents side collisions and navigates the robot in the middle of the environment.

Detecting directions to unexplored regions: We create two masks with openings on the left and right sides and apply them on the local grid map to find salient matches. Matches above a threshold is considered a new opening, and is used to re-direct the MAV.

Combining the four strategies, we obtain a system that can avoid collisions in both forward and side directions, fly toward unexplored open areas, while also simultaneously localizing and building a map of the environment.

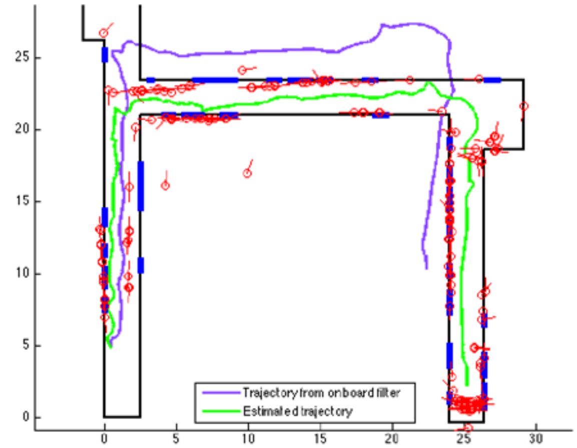


Fig. 7: Comparison between the ground-truth map of the environment and our manually-aligned estimated map. Our Wall-Floor Intersection Features are shown in red and the ground-truth floor layout in black (walls) and blue (doors). Our estimated trajectory of the quadrotor is in green, and the AR.Drone onboard estimate in purple. Accuracy of our estimate, completely constrained in the ground-truth map, shows advantages in using our features in textureless environments.

B. Experimental Setup

For the evaluation of our complete system using vistas and Wall-Floor Intersection Features, we fly a commercially-available AR.Drone quadrotor through a hallway, as shown in Figure 1. Using the 468 MHz processor on the AR.Drone, we stream 320×240 gray-scale images from the front facing camera at 10 Hz along with the IMU and sonar measurements. The main computing is done off-board, and the control outputs are streamed back to the quadrotor.

C. Map-building Results

We evaluate the quality of our map by first running our system on a set of video frames and sensor data recorded from a manual flight and compare the results with the hand-measured ground-truth of the test environment. Due to drift and unreliability in sensor readings during AR.Drone's take-off sequence [23], we only start our system once the drone stabilizes in the air. Since the entire map depends on the first robot pose at the system start, which is arbitrary due to the drift, we manually rotate our map to match the ground-truth map orientation. Figure 7 demonstrates our map of landmarks approximating the ground-truth structure sufficiently. Although there are some spurious features inside the walls that escaped rejection based on uncertainty, and large features are congregated at the bottom-right corner during the unstable landing sequence, our exploration strategies are unaffected by these small shortcomings in the map.

Furthermore, we compare the quadrotor trajectory estimated by our system with the one provided by the AR.Drone software. As shown in Figure 7, our estimated trajectory is much more accurate, being well-bounded inside the hallway interior, while the AR.Drone's estimate drifts significantly.

D. Autonomous Exploration Results

We also test our system in a hallway environment for (1) autonomously steering toward vistas, (2) building a map of the environment online, and (3) finding new corners and hallway openings to turn to. As shown in Figure 2, the vista detection was robust enough to detect and focus on distant features at the end of hallways, and effectively steer the robot toward that direction. As shown in our attached video[1], the skeleton map was accurate enough for inferring a grid map of the wall-structure, keeping the robot stay in the middle of the hallway and detecting new corners effectively. In addition, our full system could run in real-time at around 8 to 9 fps.

VI. Discussion and Future Work

We have presented a vision-based system that enables autonomous exploration strategies on an MAV in textureless indoor environments, which could not be achieved in previous work in the absence of heavy and power-hungry sensors. With our map of Wall-Floor Intersection Features, we are able to infer the entire scene structure and with vistas, steer toward open areas. We have demonstrated our complete system with two additional strategies (1) to keep the robot in the middle of the hallway, and (2) to detect opening directions to undiscovered regions. Our experiments show promising results toward a fully robust autonomous system for MAV navigation and exploration.

Although our method of combining vistas and Wall-Floor Intersection Features advances autonomous navigation and exploration capabilities of MAVs, there remain some limitations as future work. Our criteria for vistas (1) and (2) may have some practical limitations without perfect projective camera imaging. For example, the rolling shutter effect of the low-quality camera on the AR.Drone may affect the size of the features and violate (1). In addition, the Wall-Floor Intersection Features require future work for mapping with special chessboard-type floors where strong gradient lines exist in the floor's texture. Lastly, our wall-inference and hallway opening detection schemes are sensitive to thresholding values, and require fine-tuning for the specific lighting condition in the environment. A more sophisticated top-down algorithm remains as future work to make the complete system less sensitive to thresholds and be more robust to other lighting conditions.

Acknowledgements

This work is supported by an ARO MURI grant, award number W911NF-11-1-0046.

References

[1] C. Bills, J. Chen, and A. Saxena, "Autonomous MAV flight in indoor environments using single image perspective cues," in *Robotics and Automation, 2011. ICRA 2011. Proceedings of the 2011 IEEE International Conference on*, 2011.

[2] V. Murali and S. Birchfield, "Autonomous exploration using rapid perception of low-resolution image information," *Autonomous Robots*, pp. 1–14, 2012.

[3] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *Intl. J. of Robotics Research*, vol. 31, pp. 217–236, Feb 2012.

[4] S. Grzonka, G. Grisetti, and W. Burgard, "Towards a navigation system for autonomous indoor flying," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, 2009, pp. 2878–2883.

[5] A. Bachrach, R. He, and N. Roy, "Autonomous flight in unknown indoor environments," *International Journal of Micro Air Vehicles*, vol. 1, no. 4, p. 217–228, 2009.

[6] M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy, "Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments," *Unmanned Systems Technology XI. Ed. Grant R. Gerhart, Douglas W. Gage, & Charles M. Shoemaker. Orlando, FL, USA: SPIE*, 2009.

[7] S. Grzonka, G. Grisetti, and W. Burgard, "A fully autonomous indoor quadrotor," *Robotics, IEEE Transactions on*, no. 99, pp. 1–11, 2012.

[8] S. Scherer, S. Singh, L. Chamberlain, and S. Saripalli, "Flying fast and low among obstacles," in *Robotics and Automation, 2007 IEEE International Conference on*, 2007, p. 2023–2029.

[9] M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision based mav navigation in unknown and unstructured environments," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, p. 21–28.

[10] M. Achtelik, M. Achtelik, S. Weiss, and R. Siegwart, "Onboard imu and monocular vision based control for mavs in unknown in- and outdoor environments," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[11] S. Weiss, M. Achtelik, L. Kneip, D. Scaramuzza, and R. Siegwart, "Intuitive 3d maps for mav terrain exploration and obstacle avoidance," *Journal of Intelligent and Robotic Systems*, vol. 61, pp. 473–493, 2011.

[12] G. Klein and D. Murray, "Improving the agility of keyframe-based SLAM," in *Eur. Conf. on Computer Vision (ECCV)*, Marseille, France, 2008.

[13] E. Eade and T. Drummond, "Edge landmarks in monocular slam," in *Proc. British Machine Vision Conf*, 2006.

[14] M. D. Flint A. and R. I., "Manhattan scene understanding using monocular, stereo, and 3d features," in *International Conference on Computer Vision*. IEEE, 2011.

[15] D. Lee, M. Hebert, and T. Kanade, "Geometric reasoning for single image structure recovery," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 2136–2143.

[16] G. Tsai, C. Xu, J. Liu, and B. Kuipers, "Real-time indoor scene understanding using bayesian filtering with motion cues," in *International Conference on Computer Vision*, 2011.

[17] D. Lee *et al.*, "A theory of visual control of braking based on information about time-to-collision," *Perception*, vol. 5, no. 4, pp. 437–459, 1976.

[18] N. Ancona and T. Poggio, "Optical flow from 1-d correlation: Application to a simple time-to-crash detector," *International Journal of Computer Vision*, vol. 14, no. 2, pp. 131–146, 1995.

[19] D. Coombs, M. Herman, T. Hong, and M. Nashman, "Real-time obstacle avoidance using central flow divergence and peripheral flow," in *Computer Vision, 1995. Proceedings., Fifth International Conference on*. IEEE, 1995, pp. 276–283.

[20] B. Horn, Y. Fang, and I. Masaki, "Time to contact relative to a planar surface," in *Intelligent Vehicles Symposium, 2007 IEEE*. IEEE, 2007, pp. 68–74.

[21] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[22] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: speeded up robust features," in *Eur. Conf. on Computer Vision (ECCV)*, 2006.

[23] P. Bristeau, F. Callou, D. Vissière, and N. Petit, "The navigation and control technology inside the ar. drone micro uav," in *World Congress*, vol. 18, no. 1, 2011, pp. 1477–1484.

[24] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[25] E. Delage, H. Lee, and A. Ng, "A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2. IEEE, 2006, pp. 2418–2428.

[26] W. Freeman and E. Adelson, "The design and use of steerable filters," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991.

[1] http://youtu.be/x8oyld2m9Cw