View Planning and Navigation Algorithms for Autonomous Bridge Inspection with UAVs

Prajwal Shanthakumar[1(✉)], Kevin Yu[1], Mandeep Singh[2], Jonah Orevillo[1], Eric Bianchi[1], Matthew Hebdon[1], and Pratap Tokekar[1]

[1] Virginia Tech, Blacksburg, USA
{prajwal,klyu,j0n4h0,beric7,mhebdon,tokekar}@vt.edu
[2] Indian Institute of Technology Kanpur, Kanpur, India
mandeeps@iitk.ac.in

1 Introduction

Current methods for bridge inspection involve significant manual effort, with humans in harnesses and operators controlling cranes. It is time consuming, often requires closure of roads, expensive and potentially dangerous. With drone technology maturing and several commercial solutions entering the market, using UAVs for the task of bridge inspection is a great alternative to make bridge inspection faster, safer and less expensive [1].

Manual flight of UAVs, however, requires skillful piloting, and can be challenging in windy conditions or when there is no line-of-sight (long bridges, flight under the bridge, etc.). GPS based approaches for autonomous or assisted manual flights may not be possible because GPS can be unreliable around the bridge structure. Thus, GPS denied navigation for safe and accurate flight around bridges is the main motivation for our work [2].

Further, even if the GPS problem is not present, planning way-point based missions in the global frame of reference is challenging. Choosing waypoints can be tedious and would require knowledge of the bridge structure in the global frame. Planning for altitude is difficult because the bridge can slope up or down and the terrain around the bridge is often uneven. The bridge could also be over water. Hence, we explore the approach of navigating the UAV in a frame relative to the bridge structure.

In this paper, we describe our work on designing a fully autonomous UAV system to inspect bridges. Section 2 describes the UAV system and our strategy for autonomous bridge inspection. Section 3 discusses experiments and results. We conclude in Sect. 4 and indicate the direction of future work.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-33950-0_18) contains supplementary material, which is available to authorized users.

J. Xiao et al. (Eds.): ISER 2018, SPAR 11, pp. 201–210, 2020.
https://doi.org/10.1007/978-3-030-33950-0_18

2 Technical Approach

The UAV system and the algorithm for autonomous bridge inspection are described below.

2.1 System Description

The UAV built for bridge inspection weighs 3.5 kg and is shown in Fig. 1. The following are the components onboard:

- Pixhawk Autopilot with built-in IMU and Compass, running PX4 firmware [3]: for low-level flight control.
- NVIDIA Jetson TX2 running ROS (Robot Operating System): for processing sensor data and publishing the desired velocity to the Pixhawk.
- Scanse Sweep 2D LIDARs [4]: to estimate bridge structure and enable autonomous navigation.
- PointGrey Flea3 Monocular Camera (global shutter, 60 FPS, 2.0MP): for visual odometry.
- GoPro Camera: to collect images of the bridge structure for defect identification.

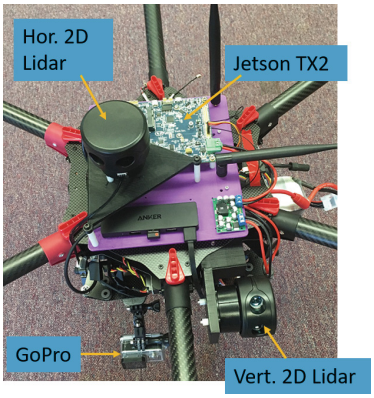

Fig. 1. UAV for bridge inspection.

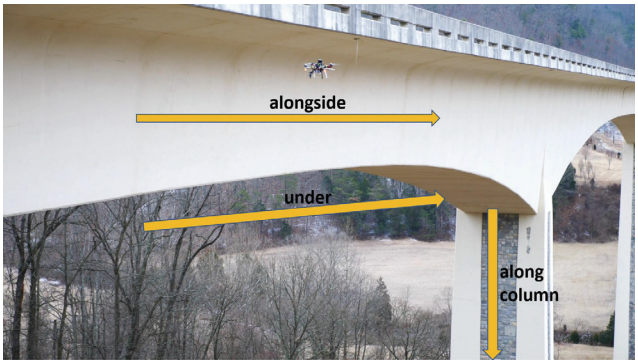

Fig. 2. Illustration of some useful navigation routines.

2.2 Algorithm Description

Our current approach for complete inspection coverage of all bridge surfaces is to autonomously execute a series of maneuvers from a library of navigation routines (Fig. 2). The algorithm consists of three parts: (1) Local Controller for execution of each of the routines through real-time LIDAR based navigation. (2) Supervisor to determine if we have completed a navigation routine and can progress

to the next (based on special cases in LIDAR data and/or Visual Odometry) (3) Coverage Planner to find the sequence of navigation routines to employ for complete visual coverage of the bridge. While the UAV autonomously navigates the bridge, a camera onboard records images of the bridge that can be examined for defects and/or used for reconstruction.

2.2.1 Local Controller

For safe and accurate autonomous traversal of bridge surfaces, the employed algorithm must be fast enough to not only run real-time but also robust to factors such as wind disturbances, unreliable GPS and Compass.

We employ 2D LIDAR (a rotating 1D LIDAR) based local control for this purpose. Using two 2D LIDARs (one horizontal and one vertical), we implement routines for traversing beside the bridge, along columns and under the deck. We exploit geometry inherent in bridges and treat bridge surfaces as planes. A 2D LIDAR scan of a plane shows up as a line as shown in Fig. 3.

A Hough Transform based approach is used to find the specific line of interest that approximates a bridge surface. PID control is employed to drive the UAV parallel to the bridge surface, maintaining a fixed distance to it. The software architecture is illustrated in Fig. 4. A more detailed description of Estimation and Control is provided below.

Fig. 3. White dots: LIDAR data; Red line: best fit using Hough Transform.

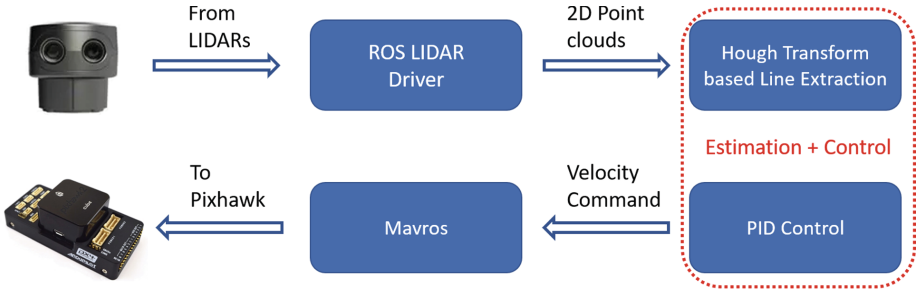

Fig. 4. Software architecture for autonomous navigation.

Estimation

To understand how bridge surfaces are estimated, consider, for instance, the problem of estimating the *vertical span* of the bridge girder surface (highlighted in red in Fig. 5). The data used for this purpose is the 2D point cloud from a vertically scanning LIDAR. The following are the steps involved (explained with ref. to Figs. 5, 6 and 7).

- Points very close (on the UAV itself) and very far (ground plane, distant trees, etc.) are first filtered out (circled in blue in Fig. 7).
- The Hough Transform is then used to find lines in the filtered data. Let's assume the three lines as shown in Figs. 5 and 7 are found with enough confidence.
- We are interested in extracting only the red line which approximates the vertical span of the girder surface. This is a line with an expected slope of $\approx 90^\circ$ and width 3–5 m. Therefore, the yellow line with slope $\approx 10^\circ$ and the green line with width ≈ 1 m can be filtered out (Figs. 5 and 7).

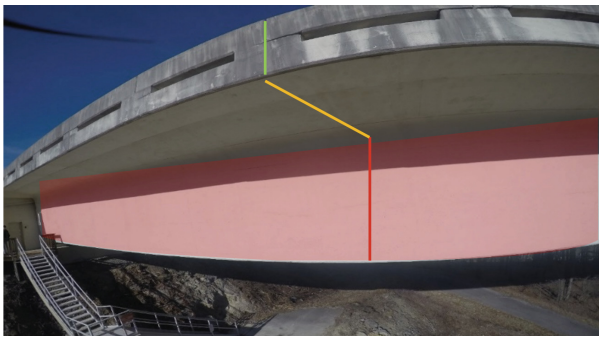

Fig. 5. The bridge girder surface highlighted in red.

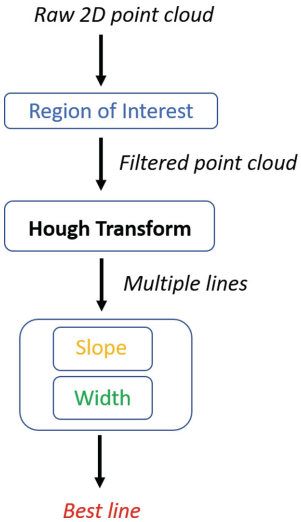

Fig. 6. Steps involved in the Hough Transform based Line Extraction process

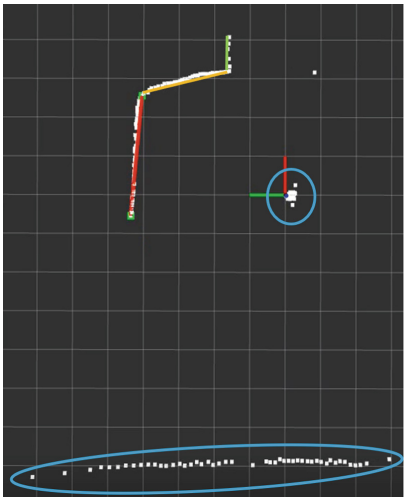

Fig. 7. Visualization (rviz) of filtered out points and extracted lines

Control

The goal of the control algorithm is to drive the UAV parallel to the (estimated) bridge surfaces. This involves two independent PID loops along two perpendicular axes to maintain position with respect to the surface. A constant nominal velocity along the third axis drives the UAV parallel to the bridge surface.

As an example, for flight beside the bridge girder (Fig. 8):

- One PID loop maintains the desired distance along the axis perpendicular to the girder surface (green axis).
- Another PID loop maintains altitude w.r.t the bridge girder (red axis).
- A nominal velocity drives the UAV along the axis parallel to the girder surface (blue axis).

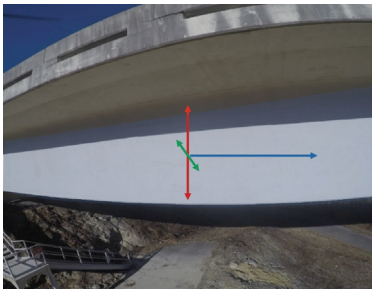

Fig. 8. Independent PID loops control position along two axes; constant velocity along the third.

2.2.2 Supervisor

The role of the Supervisor is to switch between navigation routines that the Local Controller executes, in order to execute the tour for complete coverage planned by the Coverage Planner.

Special cases in the data from the two 2D LIDARs have been identified to determine when to switch from one routine to another. For instance, consider the scenario where the UAV is traversing up a bridge column as shown in Figs. 9(a)(b). On encountering the bridge girder, the number of points in the horizontal cut of the LIDAR increases dramatically as shown in Figs. 9(c)(d). At this juncture, the Supervisor can switch from the Column Following routine to the Girder Following routine.

However, even a coarse estimate of global position can be used to better inform the Supervisor. For this purpose, we are exploring Visual Odometry. Semi-direct approaches such as SVO [5] and Visual-Inertial Odometry algorithms such as VINS-Mono [6] were found to be prone to error when subject to rotational motion. To adapt existing VO for bridge inspection, removal of outliers (clouds and dynamic objects) from features detected by the indirect methods such as ORB SLAM [7] is underway. Integration of feedback from LIDAR to correct the scale drift of ORB SLAM 2 is also in progress.

It is to be noted that the Supervisor need not necessarily be fully automated. Once the UAV has completed coverage of one bridge surface using the Local Controller, we could have a human operator switch the UAV to another routine (bridge surface). This can be done in tricky scenarios such as when switching from flight under the bridge to flight beside the bridge.

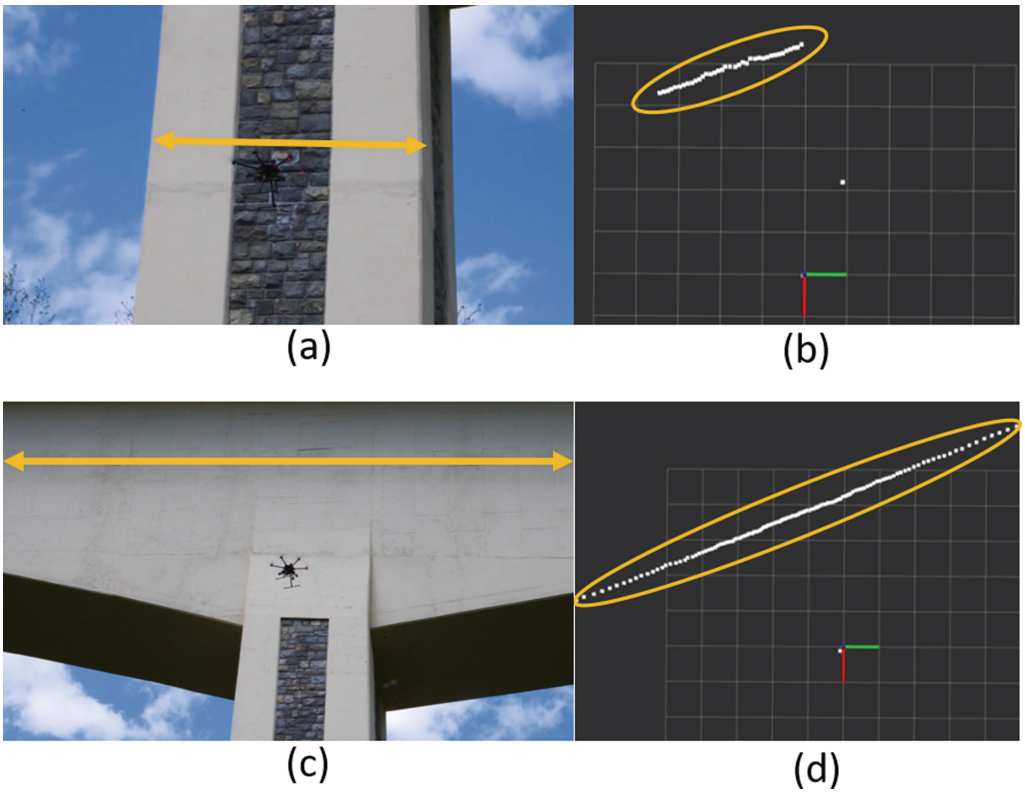

Fig. 9. Switching from column follow to girder follow.

2.2.3 Coverage Planner

To plan a tour for full visual coverage of the bridge structure, we partition the bridge into a set of planar surfaces as shown in Fig. 10. Each surface can be approximated by a polygon. Each polygon has two nodes, one of which will be chosen as the entry node and the other as the exit node. The local controller will ensure navigation between the entry and exit nodes parallel to the bridge surface. The Supervisor will identify that the exit node of the current polygon/entry node of the next polygon has been reached and pick the appropriate routine to continue the tour.

The visual coverage problem is that of finding the sequence in which the bridge surfaces must be traversed, as well as determining the entry node for each of the polygons representing the surfaces.

We formulate visual coverage as a Generalized Traveling Salesperson Problem (GTSP). A GTSP solver takes as input a graph whose nodes have been partitioned into disjoint clusters. The output of the solver is a tour in which each cluster is visited exactly once, while minimizing the total cost to visit all clusters.

In our GTSP formulation, each cluster is a polygon containing an entry and exit node as shown in Fig. 11. There exists an edge between every pair of nodes lying in separate clusters. For any cluster X, let X_1 and X_2 respectively represent

its entry and exit nodes. Then, the cost on an edge from node A_1 in cluster A to node B_1 in cluster B is given by,

$$C(A_1, B_1) = D(A_1, A_2) + D(A_2, B_1)$$

where $D(M, N)$ represents the Euclidean distance between nodes M and N in 3D space, i.e. the cost is the sum of the distance to cover the current polygon (surface) and the distance to reach the entry node of the next polygon from the exit node of the current polygon.

We can form a full cost matrix of our graph by using estimates of distance between every pair of nodes. This is the input to our GTSP solver, Generalized Large Neighborhood Search (GLNS) [8]. Once we obtain a solution from the GLNS solver we are able to get the tour, which visits every cluster exactly once. By using a GTSP formulation, we can guarantee that we visit every polygon, which represents sections of the bridge, and solve for an optimal tour.

Fig. 10. Bridge partitioned into surfaces with entry/exit nodes (blue dots).

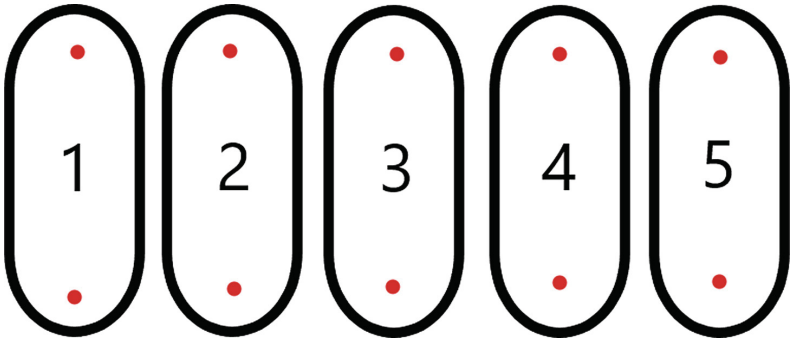

Fig. 11. GTSP clusters with entry/exit nodes (red dots) corresponding to Fig. 10.

3 Experiments and Results

We conducted experimental flights at the Virginia Tech Transportation Institute (VTTI) Smart Road Bridge, which is 53 m tall, 609 m long, 5 span, variable height concrete box girder bridge (Fig. 12). We tested LIDAR based autonomous flight (no GPS) with a GoPro camera onboard collecting images of the bridge structure.

3.1 Flight Beside Bridge Girder

The objective of the experiment was to have the UAV fly alongside the bridge with a nominal velocity of 0.5 m/s maintaining a horizontal separation of 4.5 m to the girder. The LIDAR scan (rotation) rate was 2 Hz, and hence the control signal (velocity in the direction perpendicular to the bridge) was also issued at 2 Hz. (The LIDAR can scan at upto 10 Hz, and a higher control rate can be employed, for instance, in windy conditions. The algorithmic computations only take approximately 1 ms.). Figure 13 shows the control signal driving the UAV to the desired separation of 4.5 m from the structure. Hence, we could experimentally verify that accurate flight alongside the bridge at a fixed horizontal distance from it is feasible, which is helpful for consistent data (image) collection during flight.

Fig. 12. Experiment at the VTTI Smart Road Bridge.

Maintaining the correct altitude with respect to the bridge structure proved to be more challenging. Since the terrain around the bridge is not even, using a downward facing LIDAR is not an option for maintaining altitude. We cannot rely on the barometer alone, not only because of barometric drifts and inaccuracies, but also because the bridge may slope up or down. Hence, we need some method

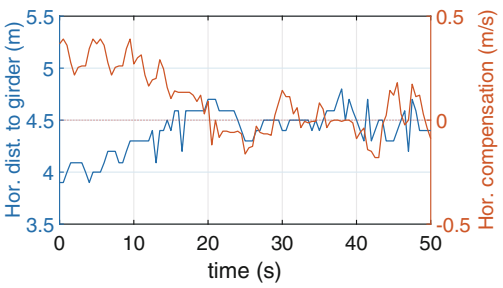

Fig. 13. Correction velocity in the direction perpendicular to the bridge structure maintaining the desired distance of 4.5 m from it.

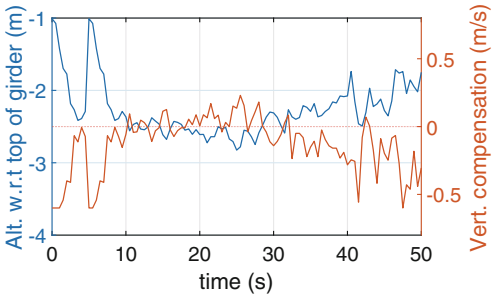

Fig. 14. Correction velocity in the vertical direction maintaining the desired altitude of 2.5 m below the top of the girder

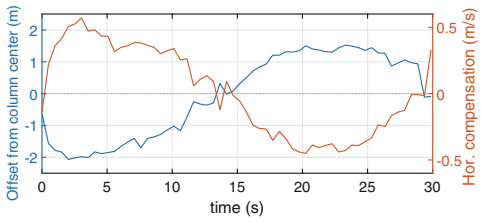

Fig. 15. Correction velocity in the direction tangential to the bridge column centering the UAV w.r.t the column

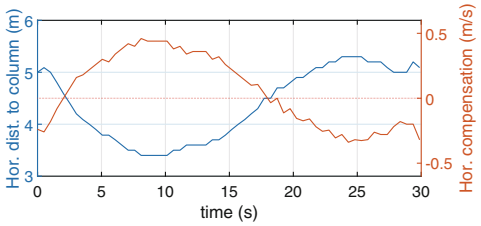

Fig. 16. Correction velocity in the direction perpendicular to the bridge structure maintaining the desired distance of 4.5 m from it

to hold position w.r.t. the bridge structure itself. For this purpose, we use the vertical cut from the 2D LIDAR rotating in the vertical plane that gives us a cross-section of the bridge. The vertical cut was used to maintain an altitude of 2.5 m below the top of the girder as shown in Fig. 14.

3.2 Flight Along Bridge Column

The objective of this experiment was to have the UAV traverse up and down a bridge column with a vertical nominal velocity of 0.5 m/s, while compensating in the horizontal direction to center and maintain a separation of 4.5 m w.r.t the column. The experimental results in Figs. 15 and 16 show the control algorithm compensating to maintain center and the desired distance to the column.

4 Conclusion and Future Work

We presented a fast (real-time) LIDAR based approach for autonomous navigation using two 2D LIDARS scanning in horizontal and vertical planes. We described strategies to switch between navigation routines to cover various bridge surfaces. We envision full coverage of the bridge using three routines for UAV

flight: (1) beside the girder, (2) along bridge columns, and, (3) under the bridge deck. (1) and (2) have already been tested and (3) is conceptually similar to (1) and (2). The GTSP algorithm is implemented similar to our prior work [9]. Future work will involve experiments to test the integration of routines and execution of the full GTSP tour. The LIDAR based approach can also be used to provide safety guarantees in assisted manual flights. Operator (pilot) input can be rejected if it will result in the UAV getting too close to the bridge structure. Adapting ideas from our present approach to different types of bridges (which could require working with other sensors such as Stereo camera for navigation) is also a future direction.

Acknowledgements. This work has been funded by the Center for Unmanned Aircraft Systems (C-UAS), a National Science Foundation-sponsored industry/university cooperative research center (I/UCRC) under NSF Award No. IIP-1161036 along with significant contributions from C-UAS industry members.

References

1. Zink, J., Lovelace, B.: Unmanned aerial vehicle bridge inspection demonstration project. Technical report (2015)
2. Gillins, M.N., Gillins, D.T., Parrish, C.: Cost-effective bridge safety inspections using unmanned aircraft systems (UAS). In: Geotechnical and Structural Engineering Congress 2016, pp. 1931–1940 (2016)
3. Meier, L., Honegger, D., Pollefeys, M.: PX4: a node-based multithreaded open source robotics framework for deeply embedded platforms. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), May 2015
4. SCANSE. http://scanse.io/
5. Forster, C., Pizzoli, M., Scaramuzza, D.: SVO: fast semi-direct monocular visual odometry. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 15–22, May 2014
6. Qin, T., Li, P., Shen, S.: VINS-Mono: a robust and versatile monocular visual-inertial state estimator. arXiv preprint arXiv:1708.03852 (2017)
7. Mur-Artal, R., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM: a versatile and accurate monocular slam system. IEEE Trans. Robot. **31**(5), 1147–1163 (2015)
8. Smith, S.L., Imeson, F.: GLNS: an effective large neighborhood search heuristic for the generalized traveling salesman problem. Comput. Oper. Res. **87**, 1–19 (2017)
9. Yu, K., Budhiraja, A.K., Tokekar, P.: Algorithms and experiments on routing of unmanned aerial vehicles with mobile recharging stations. IEEE (2018, to appear)