BRIDGE INSPECTOR v0.1

A software that makes visual inspection of bridges cheaper, safer, and more scalable.

THE ROBOTICS INSTITUTE

by Mohammad Abdull Zafar
PIs: Daniel Huber & Sebastian Scherer

Why do we need Bridge Inspector?

- 1 in 9 US bridges are considered to be structurally deficient!

- Bridge safety is a vital requirement of the U.S. road network, but Bridge Inspections are costly, disruptive and dangerous.

- Inspection can be revolutionized by automating the process of Bridge Inspection using Unmanned Arial Vehicles to inspect the bridge and doing the inspection using high fidelity models on the computer.

- This requires a software that can visualize the 3D model of the Bridge and integrate it with high fidelity images which can then be used for inspection.

What other solutions are there?

- Softwares such as PhotoSynth, PhotoTour (experimental) by Microsoft (and University of Washington) and VisualSFM (opensource) exist.

- The software are too general purpose, or in some cases missing essential features that are necessary for a complete bridge inspection experience.

- PhotoSynth and PhotoTour have the essential drawback of eliding structural artefacts while stitching images, which makes it unsuitable for detail oriented image inspection.

- VisualSfM limits the visualization to SfM generated models.

How did we implement Bridge Inspector?

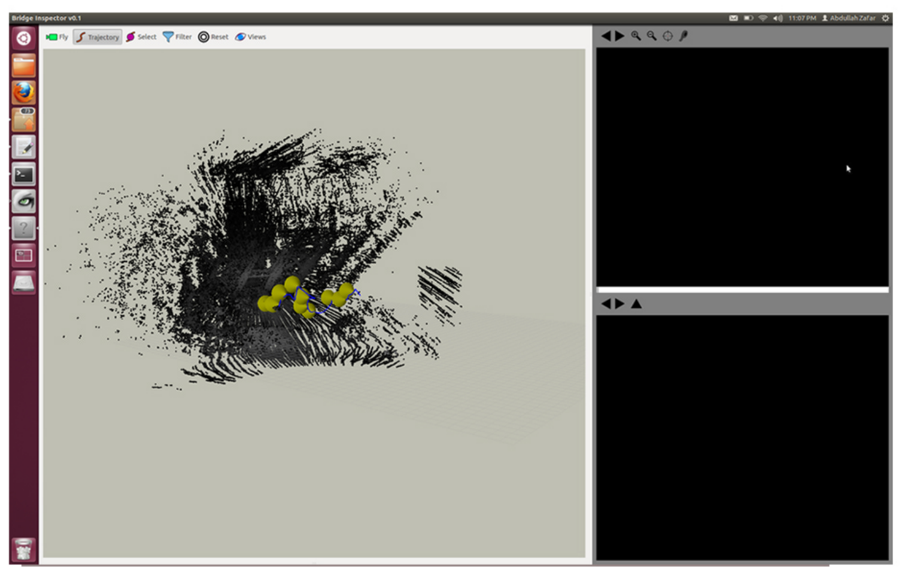
- We currently use a Unmanned Arial Vehicle (oct-rotor Mikrokopter) that flies around the bridge and take pictures of the bridge surface at regular intervals.

- There are three cameras mounted on the UAV. A stereo pair for frontal imagery and a mono for upward angles.

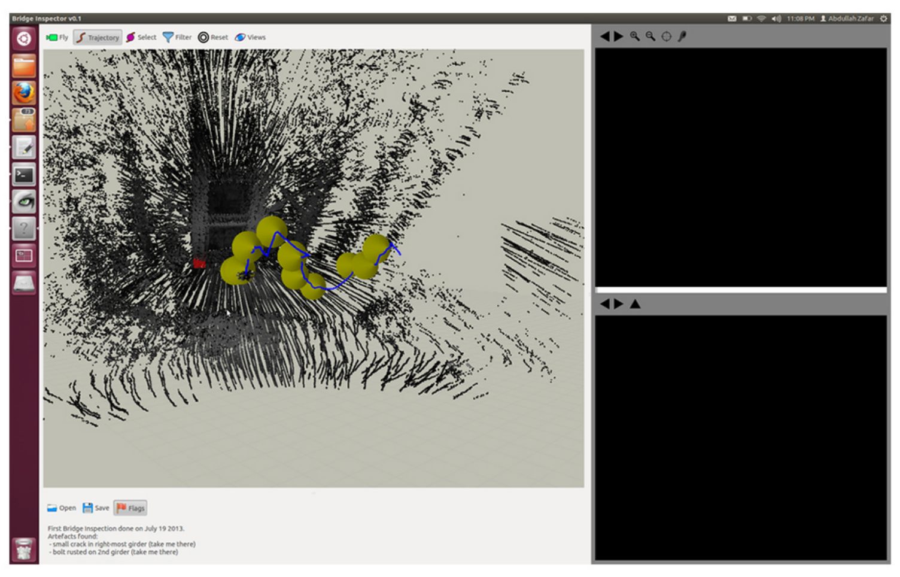
- At the same time, a laser scan is generated using a Hokuyo LIDAR.

- IMU + GPS is used for visual odometry

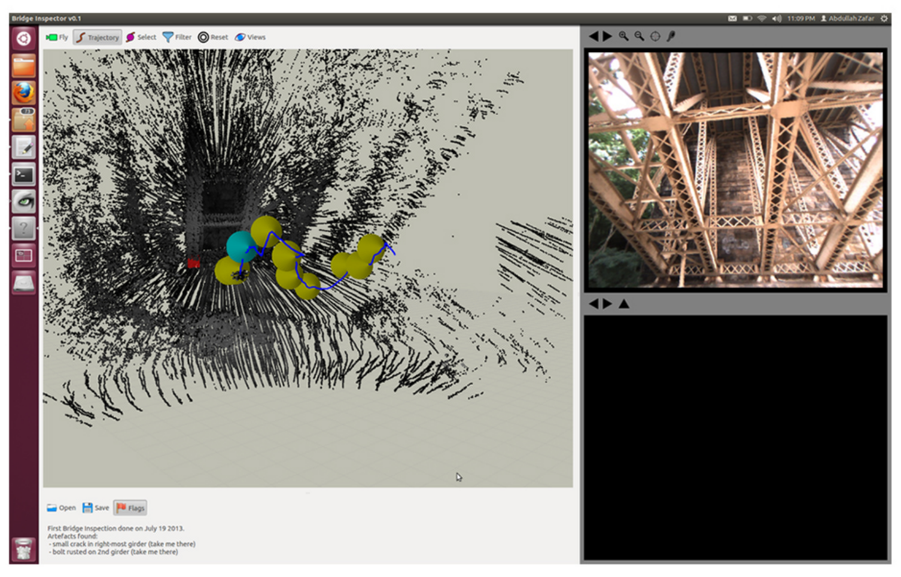
- This data is processed and made into a single file that can be processed by Bridge Inspector .



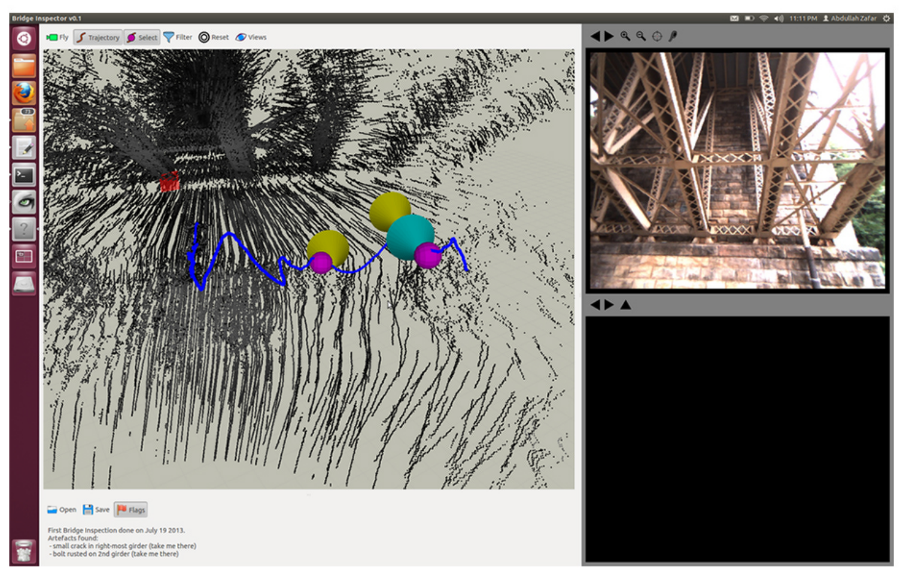
Luke can play around with the 3D bridge model in the 3D viewer and also see the path of the UAV (blue line) and some of the points where images where taken (yellow cones) displayed on the path. But Luke's not here to play, he has a bridge to inspect!



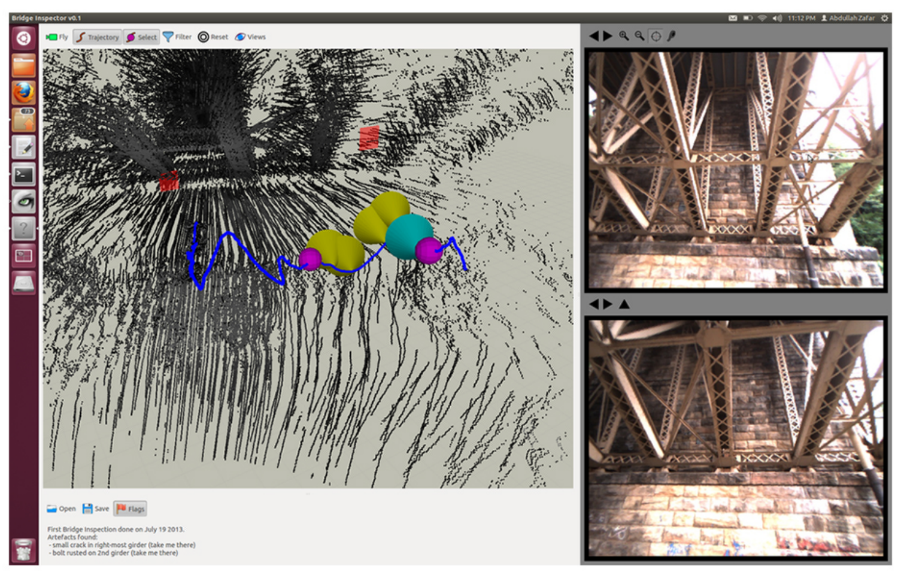
Luke opens a previously saved Inspection Summary that he did a while back and toggles the red Flag option to see the red 'danger' areas in the 3D model. It's time to inspect the red flagged area in more detail.



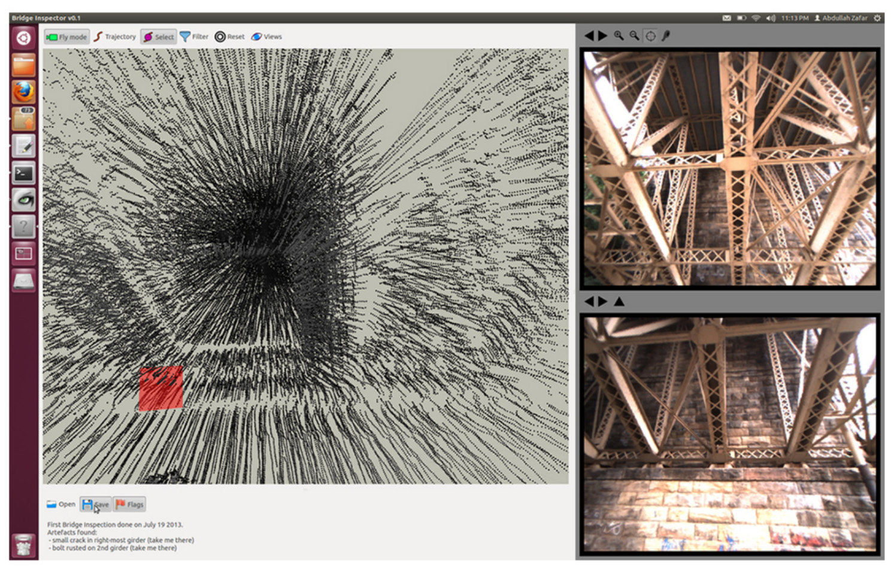
There seems to be a sizable crack! The 2D inspection window is an easy way to scale the contents up to observe cm. level cracks in the structure. Luke toggles through various views of the same crack to corroborate his finding.



Having convinced himself of the sizable crack in the girder, Luke continues with the inspection. He's interested in observing just the right most parts of the bridge now, so he uses the Selection tool to select only the relevant views. He uses the navigation viewer on the bottom right to push extra views into the 3D space.



Luke found a missing bolt in the lower beam! He found it by going through the images in the 2D viewer. Hmm, marking the spot seems like a good idea. Luke grabs the crosshair tool and marks the spot, which appears as a red flag area in the 3D view.



To inspect different parts of the bridge in isolation, Luke toggles through the different views by smooth-zooming in on them. This way he is able to ensure that he did not miss any thing. Satisfied with what he did, Luke saves the project and is done!

Luke's inspecting the Schenley bridge for cracks and shears

feature set

Fly · Trajectory · Select · Filter · Reset · Views · Prev · next · Zoom In/Out · Measure · Mark · Flag

What are trade-offs in coding with RViz/Qt?

- Native support with ROS makes RViz an attractive 3D visualizer for Robotic applications.

- Some useful experimental RViz plugins/libraries are available online, e.g. smooth transition camera controller (which was used in Bridge Inspector), although it is hard to troubleshoot online as ROS community is small.

- RViz Groovy was written in Qt and has good integration with Qt. While Qt library is extensive and well-documented, RViz's library is not.

- RViz C++ library needs more convenience functions so that it is easier to implement in Object Oriented fashion, while it could be said that 3D rendering library of RViz (called Ogre3D) is small enough to be well-exploited in a short time.

What does the future hold for Bridge Inspector?

- The bridge Inspector software is in its proof-of-concept stage and will continue to evolve with each iteration of its production cycle.

- In the next release, network capabilities will be added so that bridge inspection logs can be accessed and shared online.

- Focus will be put on upscaling the fidelity of both 3D and 2D models of the bridge. The 3D models will be made higher fidelity both in geometry and texture. The 2D images will be made higher resolution so that cm. level cracks may be detected.

- 3D point/patch detection system will be made more effective and will account for occlusions (views feature). The 2D image navigation system will be overhauled to include morphing and stitching of images.