Why do we need Bridge Inspector?

- 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 Vehi-
cles to inspect the bridge and do 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 Washing-
ton) and VisualFM (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.
- VisualFM limits the visualization to SIM 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.

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 upsampling 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.