Technical Memorandum

Applied Space Exploration Laboratory (ASEL)
Benjamin M. Statler College of Engineering and Mineral Resources
West Virginia University

Date: 9 October 2014
Memo: ASEL–14–003

To: T. Evans (WVU), M. Strube (NASA/GSFC), J. VanEepoel (NASA/GSFC), C. D'Souza (NASA/JSC)
From: Dr. John O. Woods & Dr. John A. Christian (WVU)

Title: A navigation architecture incorporating periodic, real-time pose re-initialization

1 Introduction

Proposed in this memorandum is a strategy for incorporating a secondary LIDAR-based pose computation strategy into a navigation filter architecture. The goal of the architecture is to facilitate the approach portion of on-orbit proximity operations involving a non-cooperative client.

LIDAR sensors generate a three-dimensional point cloud representation of a visible scene. Given a reasonable initial guess of the starting pose, the iterative closest point (ICP) algorithm can be used to compute updated poses — which can be plugged into a Kalman filter. In turn, the output of the filter can be fed back in to ICP as the initial guess for the next time point. This strategy is problematic, however, as an incorrect ICP output early in the approach may be propagated forward in time, with no chance of recovery.

An alternative involves use of OUR-CVFH, a strategy which requires no initial guess for pose. While this memorandum is primarily focused on the navigation filter architecture, and a full description of OUR-CVFH is outside its scope, it is important to take note of some characteristics of this method. First and foremost, OUR-CVFH runs in real-time — particularly when the object does not fill the sensor field of view. Secondly, OUR-CVFH output poses are estimates, less precise than ICP-produced poses; and this makes them ideal for refinement by ICP. Thirdy, OUR-CVFH produces multiple guesses in approximately the same time required for a single guess. Refinement of these guesses by ICP is advisable both for the dual purposes of determining which is best and producing a good pose reinitialization.

This memorandum describes an architecture (illustrated in Fig. 1) consisting of a sensor, two pose-computing modules reconciled by a FDIR (fault detection, isolation, and recovery) module, and a Kalman filter. As implemented currently, the architecture is based on publish–subscribe with a request–reply synchronization step.

2 Prototype

A basic prototype has been constructed in C++ to demonstrate the feasibility of the proposed messaging architecture. This prototype currently consists of sensor (either a SwissRanger 4000 software interface or GLIDAR, which produces point clouds from 3D models), as well as the OUR-CVFH, ICP, and FDIR modules. The current FDIR implementation does nothing other than receiving the OUR-CVFH and ICP output poses and picking the one with the best ICP score.

2.1 Messaging Architecture

The components must be started in the following order as separate processes:

1. Sensor
2. OUR-CVFH
3. ICP
4. FDIR

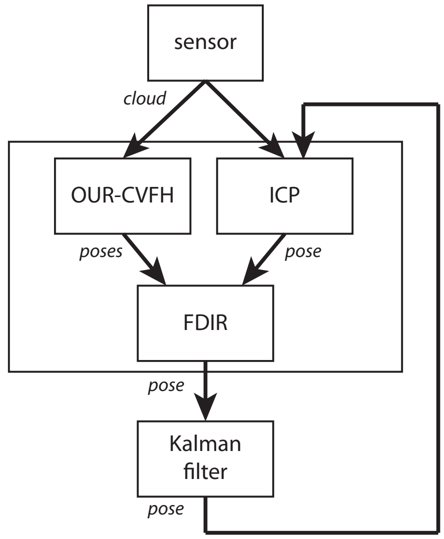

Figure 1: Proposed navigation architecture. Every module publishes pose messages, with the exception of the sensor, which provides point clouds to its subscribers. The OUR-CVFH module may output one or more poses, and is likely to include its own ICP refinement step. The pose output by the Kalman filter is fed back into ICP, when available; prior to that, ICP publishes either an identity matrix or nothing. The OUR-CVFH and ICP modules provide ICP scores for the published poses.

In each case, the component waits for all of its higher-numbered subscribers to send synchronization requests before beginning to publish. In other words, FDIR sends a synchronization request to OUR-CVFH and then to ICP. Upon receiving FDIR's request, those components send requests to the sensor; and once the sensor has received both requests, it begins to publish. Only then does information start to flow downward through the architecture. ICP's subscription to the Kalman filter requires no synchronization step.

The prototype also includes a visualization module, which can optionally subscribe to the other four components. For evaluation purposes, the temporary sensor — GLIDAR — publishes the true pose as well as a point cloud, to which the visualization component can also subscribe.

The architecture is dynamic and can easily be reconfigured. Each module publishes and/or subscribes via TCP (transfer control protocol) ports, which can be renumbered as desired to rearrange the filter. Much rearrangement — and especially simplification — is possible simply by specification of command line options.

3 Conclusions

Acknowledgments

The authors would like to thank ... of West Virginia University (WVU) for reviewing this manuscript and providing critical support along the way. This research was made possible by contract XXXXXX.