To: Professor Merz
From: Benjamin Nitkin
Subject: IGVC Progress Report
Date: November 20, 2013

This week, the software team moved forwards with visual odometry and robot geometry. After looking at several odometry and visual mapping packages, we settled on viso2_ros. It provides an estimate of distance from the origin to the robot's current location.

As we looked at the navigation stack, we realized that we had a few tasks ahead of us. Navigation draws on odometry measurements, a set of transform frames, and a global map. Odometry measurements provide an estimate of current robot position relative to a fixed point. GPS, the IMU, cameras, and wheel encoders will each provide an odometry reading. A transform frame is an ROS construct that provides positions of robot parts relative to each other, updated several times a second. The global map is provided either by the navigation stack itself or an outside node.

Visual odometry relies on the cameras to generate a position estimate. It tracks keypoints between frames to establish robot movement. Given a stereo pair, viso2 also estimates the depth of keypoints, providing better odometry data. Viso2 took more setup than most of the other nodes we've set up. First, it was distributed as source, rather than as an executable. As such, we needed to learn how to compile software in ROS. Once it compiled, trial-and-error was required to set up the poorly-documented node. At the moment, it's receiving camera data, but loses track of position frequently. Calibration parameters and a better mounting bracket should improve the fix reliability.

Transform frames establish relationships between moving pieces. The easiest example is a robot arm: transform frames would be used to describe the relationship of the shoulder to the elbow, the elbow to the hand, and the hand to the fingers. Our robot will use transform frames to specify the camera location relative to the robot, and the robot relative to a fixed world frame. The navigation stack actually uses a pair of world frames: one for odometry, and one for a fixed world frame. One frame provides an approximation of the robot's current position, including sensor drift, and is used for pathfinding. The other doesn't drift, and is used for mapping.

Once they're working, viso2 and our other odometry packages will each provide a transform from /odom, the transform frame representing the odometer origin, to /base_frame, the transform frame representing the robot's chassis. We've established a fixed transform from /map to /odom (between the map's fixed frame and the odometer's relative frame), and another between /base_link and /camera (between the robot chassis and the camera pair). Each odometry estimate will provide a link between the fixed odometer frame and the chassis. Since each frame can have only one parent, we'll generate several odometer frames, then use some heuristic to combine them into a /odom to /base_link transform.