

Philip D.S. Thoren
April 2006
UMass Lowell Student Research Symposium, 2006

This is the content of the text submitted as an abstract for SRS 2006.

Mobile robotic applications require fast reaction times to handle real world situations. The ability to sense and act accordingly is determined by the control system because it governs the robot behavior. The Phission system is meant to provide a concurrent software subsystem for processing that executes in parallel with the control system. A parallel approach is taken to avoid blocking the control system with computationally expensive processing.

Video capture hardware and faster processors that consume less power are constantly decreasing in price and are being used more often on mobile robot platforms. With the availability of these components, vision processing is being used more frequently as well. However, sensing and perception of visual information has a greater potential to delay other sensor acquisition and processing. Most vision applications and algorithms involve computationally intensive algorithms that can delay the acquisition of data from other sensors and the transmission of crucial robot control signals, particularly if the programs are serially designed.

Phission provides an answer to these problems with a thread-safe modular set of classes designed to ease implementation of a parallel data acquisition software subsystem. Phission runs in parallel with the robot control loop while still giving the feel of serial program design. Computationally expensive sensor processing subsystems run in separate threads and the control loop is free to send control commands with greater frequency. This reduces collisions or other accidents.

The Phission API supplies interfaces for capture, processing and data display. All thread management is performed internally to these classes to ease implementation of processing subsystems for the developer or researcher. Phission runs on Linux, Windows and Blackfin systems with native video capture classes, several image display classes and a pipeline class for processing any combination of filters from the set of implemented vision filters. Phission is implemented in a combination of C and C++ with support for importing into Python and Java code.