Water Detection Based on Sky Reflections

NASA’s Jet Propulsion Laboratory, Pasadena, California

This software has been designed to detect water bodies that are out in the open on cross-country terrain at mid-to-far-range (approximately 20–100 meters), using imagery acquired from a stereo pair of color cameras mounted on a terrestrial, unmanned ground vehicle (UGV). Non-traversable water bodies, such as large puddles, ponds, and lakes, are indirectly detected by detecting reflections of the sky below the horizon in color imagery. The appearance of water bodies in color imagery largely depends on the ratio of light reflected off the water surface to the light coming out of the water body. When a water body is far away, the angle of incidence is large, and the light reflected off the water surface dominates. We have exploited this behavior to detect water bodies out in the open at mid-to-far-range. When a water body is detected at far range, a UGV’s path planner can begin to look for alternate routes to the goal position sooner, rather than later. As a result, detecting water hazards at far range generally reduces the time required to reach a goal position during autonomous navigation. This software implements a new water detector based on sky reflections that geometrically locates the exact pixel in the sky that is reflecting on a candidate water pixel on the ground, and predicts if the ground pixel is water based on color similarity and local terrain features (see figure).

Assuming a water body can be modeled as a horizontal mirror, a ray of incident light reflected off the surface of a water body enters a pixel of a camera’s focal plane array (FPA). Since the angle of incidence is equal to the angle of reflection (according to the law of reflection), a direct ray from the tail of the incident ray (and within the same vertical plane as the incident ray) will enter the camera’s FPA at a pixel whose color will indicate the color of the sky being reflected along the reflected ray. Because the distance between the camera and the sky is much larger than the distance between the camera and candidate water points at normal detection ranges, the direct ray and the incident

The application itself consists of two unique applications running in the ARINC 653 system: a target application and the FAILSAFE model-based health monitoring application. The target application is a high-level simulation of the Shuttle Abort Control System (ACS), developed specifically for this task. The target application is a two-partition application with one partition allocated to the sequencing behavior, and one partition allocated to the application I/O. The health monitor application executes in its own partition. The three application partitions communicate via ARINC 653 ports and message queues, which are specified in the system module.xml configuration file. Real-time system data is provided to the health monitor via the use of ARINC 653 sampling ports that allow the health monitor application to intercept any traffic coming across the ports of interest.

This task was turned into a goal-based function that, when working in concert with the software health manager, aims to work around software and hardware problems in order to maximize abort performance results. In order to make it a compelling demonstration for current aerospace initiatives, the prototype has been additionally imposed on a number of requirements derived from NASA’s Constellation Program.

Lastly, the ARINC 653 standard imposes a number of requirements on the system integrator for developing the requisite error handler process. Under ARINC 653, the health monitoring (HM) service is invoked by an application calling the application error service, or by the operating system or hardware detecting a fault. It is these HM and error process details that are implemented with the MDS technology, showing how a static-analytic approach is appropriate for identifying fault determination details, and showing how the framework supports acting upon state estimation and control features in order to achieve safety-related goals.

This work was done by Gregory A. Horvath, David A. Wagner, and Hui Ying Wen of Caltech and Matthew Barry of Kestrel Technology for NASA’s Jet Propulsion Laboratory. For more information, contact iaoffice@pl.nasa.gov.

This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-46981.
Nonlinear Combustion Instability Prediction

Marshall Space Flight Center, Alabama

The liquid rocket engine stability prediction software (LCI) predicts combustion stability of systems using LOX-LH2 propellants. Both longitudinal and transverse mode stability characteristics are calculated. This software has the unique feature of being able to predict system limit amplitude.

New methods for predicting stability have been created based on a detailed physical understanding of the combustion instability problem, which has resulted in a computationally predictive algorithm that allows determination of pressure oscillation frequencies and geometry, growth rates for component modes of oscillation, development of steepened wave structures, limit (maximum) amplitude of oscillations, and changes in mean operation chamber conditions.

The program accommodates any combustion-chamber shape. The program can run on desktop computer systems, and is readily upgradeable as new data become available.

This program was written by Gary Flando of the University of Tennessee, Space Institute; Calspan Center, for Marshall Space Flight Center. For further information, contact Sammy Nabors, M SFC Commercialization Assistance Lead, at sammy.a.nabors@nasa.gov. Refer to MFS-32549-1.

Characterization of Cloud Water-Content Distribution

NASA’s Jet Propulsion Laboratory, Pasadena, California

The development of realistic cloud parameterizations for climate models requires accurate characterizations of sub-grid distributions of thermodynamic variables. To this end, a software tool was developed to characterize cloud water-content distributions in climate-model sub-grid scales.

This software characterizes distributions of cloud water content with respect to cloud phase, cloud type, precipitation occurrence, and geo-location.