Software

Change_Detection.m

The Change_Detection.m MATLAB tool detects changes in an image by comparing the image to a background estimation. The change detection software is a MATLAB function designed to work on either a single image or a sequence of images, and computes changes with respect to a background image by one of four techniques:

1. Frame-to-frame change detection — the change detection image represents the difference between two frames.
2. Rolling median change detection — the change detection image represents the difference between an image and a median background estimation taken over a subset of the total frames presented to the function.
3. Rolling mean change detection — the change detection image represents the difference between an image and a mean background estimation taken over a subset of the total frames presented to the function.
4. Rolling mode change detection — the change detection image represents the difference between an image and a mode background estimation taken over a subset of the total frames presented to the function.

The software is an improvement on other scripting techniques by functionalizing the code with an INPUT/OUTPUT structure format that may work either at the command line or in conjunction with a separate graphic user interface function. The function also allows periodic background estimations instead of background estimations on a per-frame basis.

All computation is done on grayscale imagery, so color or multispectral images will be first converted to grayscale. Differences from the background that exceed a preset threshold are reported as changes in the form of a returned binary mask.

The algorithm can work in single-frame or multi-frame mode. In single-frame mode, only one image is being investigated for changes relative to a background. In multi-frame mode, a sequence of images is being investigated for changes relative to the background.

After the image changes are detected, the image is then converted into a binary representation of the change detection image by setting a threshold. The threshold is given as multiples of a standard deviation of the values in the difference image (the difference between the image and the background). The binary change detection mask is then multiplied by the original image in the sequence to obtain the values of the pixels that have changed. The outputted mask is the same size as the inputted frame, but now only the pixels that have changed are non-zero.

This work was done by David M. Palacios and Steven J. Lewis of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-47671.

AGATE: Adversarial Game Analysis for Tactical Evaluation

AGATE generates a set of ranked strategies that enables an autonomous vehicle to track/trail another vehicle that is trying to break the contact using evasive tactics. The software is efficient (can be run on a laptop), scales well with environmental complexity, and is suitable for use onboard an autonomous vehicle. The software will run in near-real-time (2 Hz) on most commercial laptops. Existing software is usually run offline in a planning mode, and is not used to control an unmanned vehicle actively.

JPL has developed a system for AGATE that uses adversarial game theory (AGT) methods (in particular, leader-follower and pursuit-evasion) to enable an autonomous vehicle (AV) to maintain tracking/trailing operations on a target that is employing evasive tactics. The AV trailing, tracking, and reacquisition operations are characterized by imperfect information, and are an example of a non-zero sum game (a positive payoff for the AV is not necessarily an equal loss for the target being tracked and, potentially, additional adversarial boats). Previously, JPL successfully applied the Nash equilibrium method for onboard control of an autonomous ground vehicle (AGV) travelling over hazardous terrain.

This work was done by Terrance L. Huntsberger of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-48697.

Ionospheric Simulation System for Satellite Observations and Global Assimilative Modeling Experiments (ISOGAME)

ISOGAME is designed and developed to assess quantitatively the impact of new observation systems on the capability of imaging and modeling the ionosphere. With ISOGAME, one can perform observation system simulation experiments (OSSEs). A typical OSSE using ISOGAME would involve: (1) simulating various ionospheric conditions on global scales; (2) simulating ionospheric measurements made from a constellation of low-Earth-orbiters (LEOs), particularly Global Navigation Satellite System (GNSS) radio occultation data, and from ground-based global GNSS networks; (3) conducting ionospheric data assimilation experiments with the Global Assimilative Ionospheric Model (GAIM); and (4) analyzing modeling results with visualization tools. ISOGAME can provide quantitative assessment of the accuracy of assimilative modeling with the interested observation system. Other observation systems besides those based on GNSS are also possible to analyze.

The system is composed of a suite of software that combines the GAIM, including a 4D first-principles ionospheric model and data assimilation modules, an Internal Reference Ionosphere (IRI) model that has been developed by international ionospheric research communities, observation simulator, visualization software, and orbit design, simulation, and optimization software.

The core GAIM model used in ISOGAME is based on the GAIM++ code (written in C++) that includes a new high-fidelity geomagnetic field representation (multi-dipole). New visualization tools and analysis algorithms for the OSSEs are now part of ISOGAME.

This work was done by Xiaoqing Pi, Anthony J. Mannucci, Olga P. Verkhoglyadova, Philip Stephens, Brian D. Wilson, Vardan Akopian, Attila Komjathy, and Byron A. Iijima of Caltech