(Preprint) AAS 20-101

MARS 2020 AUTONOMOUS ROVER NAVIGATION*

Michael McHenry†, Neil Abcouwer, Jeffrey Biesiadecki, Johnny Chang, Tyler Del Sesto, Andrew Johnson, Todd Litwin, Mark Maimone, Jack Morrison‡, Richard Rieber, Olivier Toupet, Philip Twu

Rovers have been critical elements of Mars Exploration, beginning with Sojourner in 1997, Spirit and Opportunity in 2004, and most recently the Mars Science Laboratory's Curiosity rover, which has now traveled more than 23 km since its landing in 2012. In the summer of 2020, NASA and the Jet Propulsion Laboratory (JPL) will launch the Mars 2020 rover with the goal of acquiring curated samples from Mars for possible return to Earth by a future mission. While the mobility mechanisms are inherited from the MSL rover, a number of significant technological advancements to software and avionics were made in order to meet mission objectives. In this paper, we present the most significant improvements in the area of Autonomous Rover Navigation, specifically:

- Use of the Vision Compute Element (an FPGA-equipped co-processor) to accelerate image processing.
- Software changes to enable image and navigation processing to occur in parallel with vehicle motion.
- A new path-planner algorithm named "Enhanced Nav" enabling autonomous drives in more challenging terrains than Curiosity can traverse.

INTRODUCTION

The Mars 2020 (M2020) mission continues NASA's history of rover-based exploration of Mars that began with the Sojourner rover in 1997 and continues today through the operation of the Mars Science Laboratory's (MSL) Curiosity rover, which has been operating on Mars since 2013.

M2020's science goals include:

- Identifying evidence of past environments capable of supporting life

*This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

† Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109.

‡ Digital Insights, 59-647 Kaala Road, Kamuela, HI 96743.

- Seeking evidence of past life
- Caching samples of rocks for possible return to Earth, which could be capable of documenting evidence of past life

Mobility is critical to these objectives. Mobility enables an otherwise fixed lander to explore widely, reach distinct areas of scientific interest, and collect critical contextual knowledge that informs sample interpretation.

Baseline Reference Scenario and Drive Distance per Sol

As part of the M2020 team's effort to interpret the science objectives as engineering requirements, the project developed the 'Baseline Reference Scenario' that includes quantitative mobility objectives of a representative surface mission[1]. It describes a notional mission involving traveling from the landing site to two distinct geological 1km-wide regions of interest (thus requiring a total of 12 km of driving over 85 Martian days).

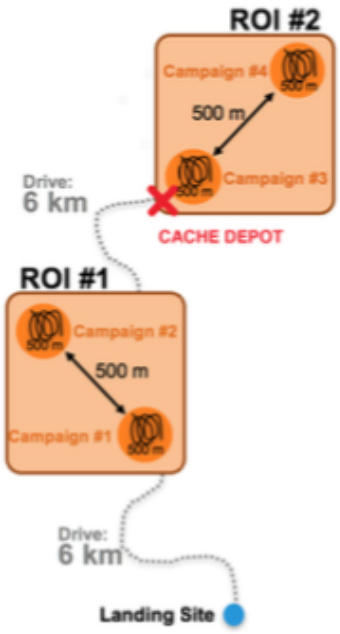

Figure 1. Baseline Reference Scenario.

The gap between this high-level goal, which implies an average traverse rate of 141 m/sol, and MSL performance highlighted the need for significant improvements to the mobility system. MSL mobility operations are influenced by many factors so direct comparison is difficult but some simple metrics illustrate how different M2020's mobility goals are from MSL operations.

+ Curiosity accumulated 12km of odometry as of sol 1187

+ The single longest drive sol was 142 meters

+ Curiosity drove more than 100m only 18 times over the first 2500 sols

+ The median drive distance per sol is approximately 28 meters

One reason that MSL rarely commands long-distance drives is that Curiosity's use of computer vision processing as part of its autonomous behavior capability will reduce the average overall speed of the rover. Its average speed can be 20-90% slower depending on how much autonomous processing is enabled, due to having to stop the vehicle to grab one or more image pairs, record them into flash memory, and nominally process them before choosing the next drive step. Thus when planning a drive, the operations team has to trade the use of those more sophisticated behaviors against the increased drive duration and subsequent reduction in time and power available for science investigations and other rover activities. The Visual odometry (VO) behavior can provide more accurate localization over the course of the drive, measuring and compensating for slip as opposed to relying entirely on wheel odometry for distance (and the rover's inertial measurement unit for attitude). The Hazard Avoidance behavior enables the rover to automatically detect and avoid geometric terrain hazards but incurs a significant slowdown. For shorter drives, operators can choose to drive without autonomous terrain assessment, using previously downlinked stereo imagery to evaluate the terrain themselves. For longer drives, limited image resolution and terrain occlusions can make ground assessment impossible, and the operators would need to enable the rover's onboard capability to assess terrain safety and autonomously avoid hazards. As with Curiosity's predecessors the Mars Exploration Rovers, tradeoffs exist between manual path planning versus the use of on-board autonomy and must be carefully assessed by rover operators.[2]

When executing a GO_TO command in which Curiosity drives itself to a specified waypoint, the use of visual odometry reduces the average speed of the vehicle from approximately 103 m/hr to 46 m/hr. When performing hazard detection and hazard avoidance (without visual odometry), the average rover speed is reduced to 15.7 m/hr. Using both visual odometry and hazard avoidance at the same time reduces the rover speed to just 11.3 m/hr and has only rarely been exercised.

Terrain Complexity

Another way in which the M2020 mission places greater demands on the mobility system is the need to drive autonomously through more challenging terrain. As the M2020 mission design evolved, potential landing sites were evaluated carefully with respect to the frequency of expected mobility hazards. In addition, the initial Baseline Reference Model was refined to reflect both landing site specific traversability and regions of interest (ROI) as well as updated models of mission operation constraints. The conclusion was that the M2020 rover must be capable of driving autonomously through terrain with greater density of hazards than Curiosity is capable of. In terms of Cumulative Fraction Area (CFA), a measure of the proportion of the ground surface covered by rocks, Curiosity has predominantly been in terrain classified from orbital imagery as having a CFA of

7% or less. Many of the landing sites considered for M2020 had high proportions of terrain with CFAs higher than 7%, including some with contiguous areas of 10-15% CFA. [3]

Curiosity's hazard detection algorithm and software is largely inherited from that used on the Mars Exploration Rovers and was designed to be simple enough to run on those rover's 20 MHz RAD6000 processors. That simple hazard detection algorithm treats the rover as a disk whose diameter encompasses the vehicle's body at any orientation with some margin. [4] Treating the rover as a disk also helped keep the path planning software simple because vehicle safety assessment is independent of vehicle orientation. We were able to show that this basic attribute impeded autonomous navigation performance in terrains with CFAs above 7% and reduced the likelihood of finding a 100m path through 15% CFA terrain to near zero.

Summary

Early in the project it was recognized that we needed an updated mobility system that was able to drive autonomously faster and further per sol. In addition, we identified the need for algorithm improvements that made a more sophisticated assessment of terrain safety.

To address those needs, the M2020 mobility team have incorporated three major improvements described in this paper. 1) We have integrated the Vision Compute Element (VCE); a second RAD750 CPU with an FPGA-powered co-processor that serves to accelerate image progressing that underlies Visual Odometry and the binocular stereo-ranging that underlies onboard hazard detection. 2) We have modified the software to enable Thinking-While-Driving (TWD) so that the image and Autonomous Navigation (AutoNav) processing can happen while the rover is executing the previously selected arc. 3) We have developed an enhanced autonomous navigation (ENav) algorithm that allows for traversing more challenging terrain.

MSL COMPARISON

In order to illuminate the constraints on the rover mobility system and its operations, this section will briefly describe key elements of the MSL Curiosity and the M2020 rover.

Mechanical

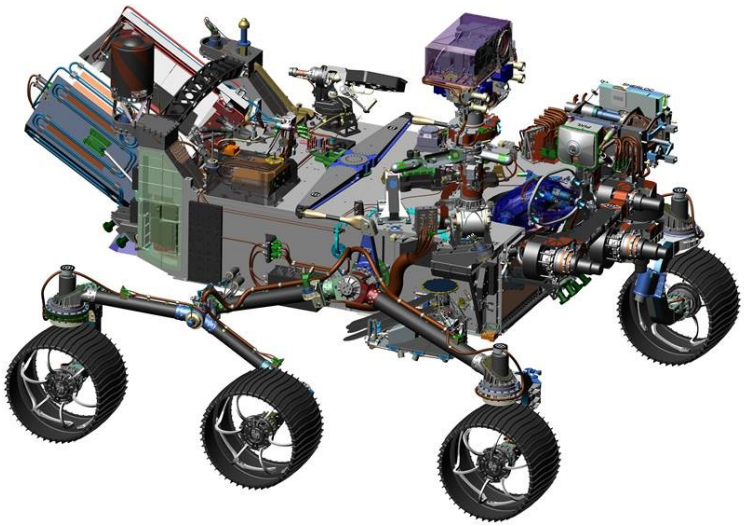

Figure 2. The M2020 Rover.

Mechanically, the M2020 rover mobility system is almost identical to Curiosity. They share the same 6-wheel rocker-bogie suspension system that has been common to all NASA Mars rovers. The 4 corner wheels are steerable and enable the rover to turn-in-place and drive straight or curved arc trajectories, but not crab sideways. Avionics limitations prevent the rover from moving the steering actuators and the mobility actuators at the same time, so both rovers must stop to adjust their steering angles. The M2020 rover's wheels have an improved design and slightly different geometry but those changes do not significantly impact mobility performance at the planning level described here.

Imaging

The M2020 rover has a suite of enhanced engineering cameras (EECams) used for onboard mobility.[5] These new cameras have a 20 MPixel Bayer pattern focal plane enabling acquisition of high-resolution color images, but onboard mobility processing relies on 1280x960 monochrome images obtained from 4x down sampling. Curiosity uses a combination of mast-mounted pointable Navigation camera (NavCam) imagery together with imagery from the body mounted Front Hazard Detection (HazCam) cameras. Curiosity's NavCams have 1024x1024 focal planes and optics with a 45x45 deg FOV. The higher resolution of the M2020 cameras allowed much wider field-of-view optics (95x71 deg) which in-turn enables M2020 to reduce the need for the mast to slew horizontally during mobility.

Processor

On-board processing primarily happens on the Rover Compute Element (RCE), which is a dual-redundant avionics box powered by a BAE RAD750 running at 133 MHz and providing 266 MIPs. In practice, approximately 50% of the available CPU cycles are consumed by baseline housekeeping functions not directly related to autonomous

navigation. That restriction on available computational horsepower is a significant limitation on the algorithms that can be run on the rover and a large part of why the use of Curiosity's autonomy results in such a large reduction in traverse rate.

Curiosity, like the Mars Exploration Rovers (MER) Spirit and Opportunity, possesses two key vision-based capabilities: Stereo Vision and Visual Odometry. Stereo Vision combines images from two cameras (either the mast-mounted NavCams or body-mounted HazCams) to compute the 3D location corresponding to most pixels in the image. These surface measurements are then used to update a rover-centered 2.5D elevation map used for hazard detection and path planning. [6] MER and MSL use the GESTALT algorithm for autonomously navigating to a specified waypoint while avoiding detected hazards. Autonomous navigation using GESTALT consists of repeated cycles of:

+ stereo imaging (from one or more source cameras/pointing directions)

+ stereo processing

+ elevation map updating

+ hazard map updating

+ path selection

+ driving a short segment along the selected path

+ repeat

Visual Odometry calculates the full 6 degree-of-freedom actual rover motion using image features and associated range estimates obtained from before-and-after pairs of stereo NavCam images. Once confirmed, the Visual Odometry pose estimate replaces the wheel odometry (and optionally also the inertial measurements) to update the vehicles onboard position and attitude estimate. MSL's visual odometry algorithm significantly improved upon the MER implementation's computational efficiency and robustness.[7][8][9]

One of the significant M2020 changes described below is the use of a new avionics component that serves as a coprocessor during mobility. The Vision Compute Element (VCE) marries an additional RAD750 with a Xilinx V5QV radiation hard FPGA on which is run critical image processing functions.

Operations

The basic operations of the M2020 rover will be similar to MSL. For a few months after landing the operations team will operate on a cycle synchronized to a Martian day (24 hours and 40 minutes). Data from the rover will typically be relayed via one of an array of Mars orbiters. Daily commanding to the rover from its Earth operators will typically be direct from the NASA's Deep Space Network to the rover's High Gain Antenna (HGA) and occurs at a much slower bit-rate than rover-to-orbiter communication.

VISION COMPUTE ELEMENT (VCE)

The first mobility computation upgrade is incorporation of the Vision Compute Element (VCE) which also plays a critical role in the Lander Vision System (LVS) used during the Entry, Descent and Landing phase. [10]

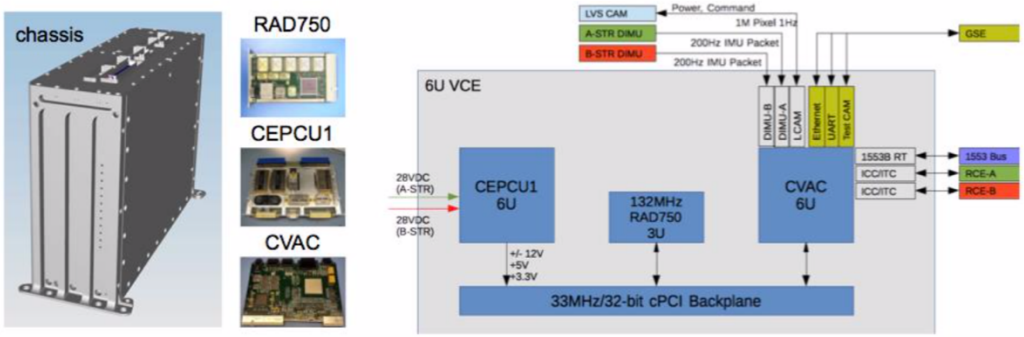

Figure 3. The Vision Compute Element hardware.

The VCE consists of 3 6U boards: A BAE RAD750 identical to the one of the RCE; A Power Conditioning Unit; and the M2020-developed Computer Vision Accelerator Card (CVAC) containing a Xilinx Virtex-5QV Field Programmable Gate Array (FPGA). The V5QV is a reprogrammable radiation-hard FPGA which is compatible with commercial Virtex-5 devices. It is the computational power of this FPGA that enables M2020 to perform image processing faster and at greater resolution than Curiosity. The CVAC also contains a house-keeping FPGA, 1 GB DDR2 SDRAM, 32 MB NOR non-volatile storage, and 16 GB NAND non-volatile storage used for telemetry.

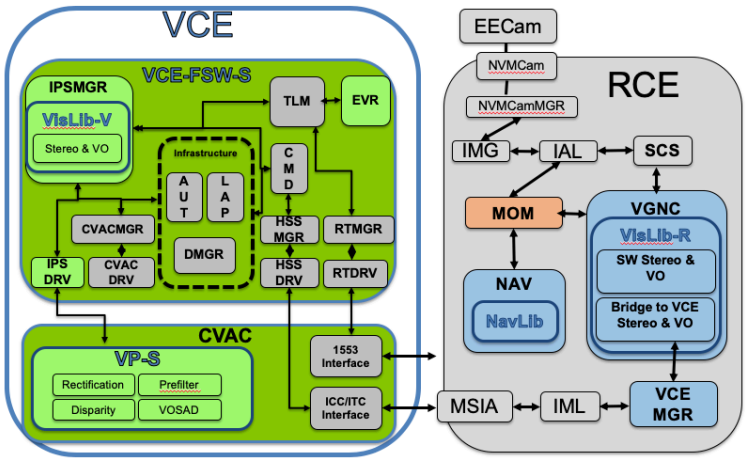

Figure 4. Architectural Integration of VCE with main Rover Compute Element (RCE).

Figure 4 shows the basic architecture of the M2020 mobility system and its distribution between the RCE and VCE. The VCE serves as a co-processor, responsible only for

two critical image processing functions (Visual Odometry and Stereo Ranging). All other functions including image acquisition, motor control, pose estimation, hazard detection and hazard avoidance/path planning are all performed on the RCE processor. In fact, the RCE is capable of performing the image processing functions normally performed by the VCE if it was to become unavailable. The much greater time required by the RCE to perform the same functions would translate to significantly slower drive rates.

The RCE communicates to the VCE via both 1553 and bi-directional high-speed LVDS serial (~6Mb/s). In surface operations most data exchange and commanding occur over the serial line while 1553 is limited to health and status telemetry, time synchronization and some command functions common with the LVS application.

Largely because of the need to minimize changes to the basic rover avionics design, the rover's cameras are not directly interfaced to the VCE. Instead, images are acquired by the RCE using heritage interfaces and then transmitted to the VCE for processing. The inefficiency of this communication was a trade made to limit interdependencies and avoid changes to RCE hardware and software design inherited from MSL.

FPGA Image Processing

The visual odometry algorithm adopted for FPGA implementation is different from that of MER and MSL. In particular, the algorithm is more suitable for FPGA implementation in that 1) it leverages the availability of previously computed dense stereo rather than using sparse left to right matching and 2) it relies on feature matching rather than template-based search. [11]

At the outset of the M2020 project, the core FPGA implementations of stereo processing was already mature having been developed and tested as part of a long series of NASA, DoD, and internal funding. [12] In the years immediately preceding the M2020 project, the stereo FPGA implementation had been demonstrated in a flight context by being coarsely integrated with MSL's Vehicle System TestBed (VSTB) using commercial development boards. As part of the M2020 development effort the existing FPGA cores were updated to fit within the CVAC memory and system architecture.

Figure 4 also shows the specific image processing modules implemented in the FPGA. The first three are the primary steps of binocular stereo ranging while the last is a key component of M2020's visual odometry.

+ Image Rectification: Takes as input a raw camera image containing lens distortion and resamples it to produce a rectified image that corresponds to a perfect pin-hole camera and whose rows obey the stereo epipolar constraint (the image of a point in the rectified left image must appear at the same row in the rectified right image). The resampling is represented by a 'warp table' which is computed on the VCE's RAD750 using camera model information passed from the RCE with each image.

+ Prefilter: A preprocessing step that serves as a high-pass filter to remove low-frequency intensity changes and thus provide robustness to overall brightness differences between the left and right images.

+ Disparity: This is the core stereo matching functionality. It finds the portion of the right rectified image (along the same row) that best matches a sub window around the current pixel of the left image. The difference in column location between the left pixel and its match in the right image (the disparity) is directly related to range. Thus, the output of this module is a disparity image that needs only a little triangulation math to be translated to range (and later, to a 3D location).

+ The fourth FPGA module is the VOSAD (Visual Odometry Sum Absolute Difference) core which is responsible for evaluating prospective matches between features in subsequent image pairs.

Summary

MSL AutoNav typically performs 4 stereo computations per planning cycle. Three separate NavCam wedges separated by 40 degrees and a near-field front HazCam pair. Each is performed at 256x256 resolution. On the other hand, M2020 processes a single 1280x960 stereo pair resulting in approximately 5 times more range measurements per cycle. M2020's not having to slew its mast to 3 different pointing vectors during autonomous navigation saves considerable time. The decrease in M2020's single wedge FOV of 90 deg (as compared to Curiosity's effective FOV of 125 deg) is mitigated by its higher angular resolution (2.5x) and its use of always-on VO.

THINKING-WHILE-DRIVING

M2020's wheels, like Curiosity's, drive at a maximum speed of 4.2 cm/sec. If the rover drove continuously in a straight line this would equate to 151 m/hr. In reality, there is additional overhead associated with steering (the drive motors must be stopped when the steering motors are active), motor ramp up and down, and more. When Curiosity's visual odometry or hazard avoidance behavior is enabled, the overhead increases dramatically due to the stop-image-process-drive nature of its mobility planning system. Curiosity typically takes 1m steps when utilizing visual odometry alone and steps as small as 50 cm when using hazard avoidance in obstacle-rich areas. Given that the rover drive rate is CPU limited in these modes, not performing any autonomy related processing during the time the wheels are in motion is inefficient. At MSL drive rates, the proportion of the drive time to 'thinking' time is small and the inefficiency tolerable. But M2020's need for faster driving calls for utilizing the available CPU cycles while driving.

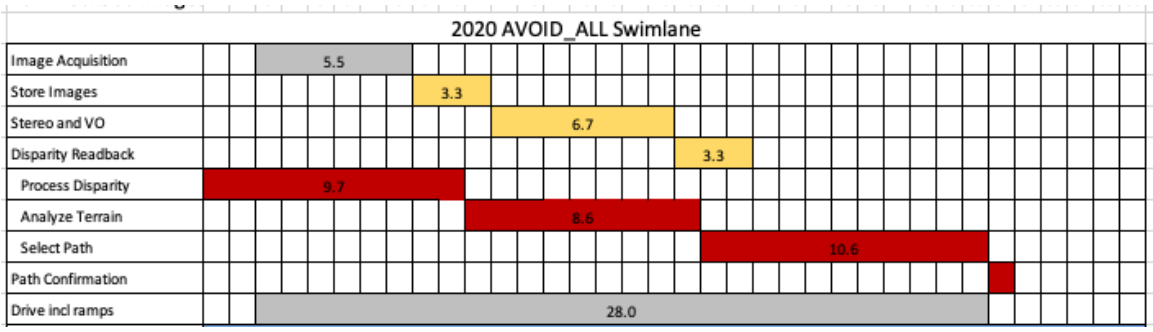

Figure 5. Swimlane illustration of how image processing, navigation processing and driving occur in parallel.

Figure 5 shows a swimlane visualization of a typical drive and planning cycle with overlapping driving of 1m arcs, image processing (on the VCE) and hazard detection and avoidance (entirely on the RCE). In practice, the timing of each component varies, some of them by large amounts. Small variations are introduced by the RCE processor being shared by multiple functions not directly related to mobility such as data management, or parallel instrument activities. Larger variations are introduced by the nature of the environment being navigated. For example, executing sharp arcs results in an overall slower rate of progress because the inside wheels must turn less than the maximum 4.2 cm/s. Similarly, when the ENav algorithm is forced to alter the steering (change the arc curvature) there will be additional time required to re-steer. In more challenging terrain, the ENav algorithm will be forced to execute point turns of varying magnitude.

Note that the image pair acquired at the start of each drive step is read out and processed on the VCE while the rover is in motion. And during the initial portion of the same drive step, the RCE is digesting the range data from the start of the previous already completed motion. Without this parallelization, the M2020 drive rate would be more than halved. The mobility software synchronizes the different activities at some steps. For example, image acquisition happens at the beginning of each drive step.

The delay between when image data is acquired and when the derived information is available presents significant software and algorithmic complexity. For example, VO provides an estimate of where the rover was at the start of the currently executing motion. That knowledge must be incorporated in a manner consistent with the wheel odometry and inertial measurements taken since that time. Similarly, the navigation software must choose (during the 'Select Path' interval) the subsequent motion command to be executed after the current motion completes. In other words, it selects a path based on where it anticipates being at the end of the current motion. And its evaluation of the safety of that motion is based on an elevation map whose most recent data derives from imagery taken at the start of the previous step (i.e., two steps prior to the one being planned).

Ensuring the navigation software is robust to inevitable errors in both the predicted end position of the currently executing motion, as well as the predicted effect of the selected path has been one of the most challenging aspects of developing the M2020 mobility system. The greatest source of these errors is due to vehicle slip and the next section will outline how that robustness is accomplished. In terms of the swimlane, note that the 'Select Path' computation will not start until the most recent visual odometry result has been incorporated. This is to ensure that the selected path is chosen based on the most up-to-date pose information possible. And to partially remediate the risk of planning based on the uncertain end pose, the select motion is re-evaluated just prior to being delivered for execution based on the current pose estimate which, because it is based on recent inertial measurements, will be more accurate than the prediction made at the beginning of the 'Select Path' computation.

Ideally, ENav will identify a safe and effective motion within the time interval shown in the swimlane. But it is not possible to exhaustively evaluate all possible paths within that interval and in such cases ENav will continue to evaluate motions until a satisfactory one is identified. If none is found the drive will fail and it will be left to human operators to make their own determination of the best way forward.

ENHANCED NAVIGATION (ENAV)

The GESTALT algorithm was designed for the 20MHz RAD6000 processor of the MER rovers. That very limiting computational constraint understandably led to design decisions that are less applicable to the VCE-powered M2020 rover. In particular, in the absence of Visual Odometry (which was introduced on MER only at the very end of development and takes over a minute to compute), it would be hard to directly fuse range/elevation measurements over time without introducing discontinuities that would in turn lead to false-positive hazards. Instead GESTALT performs hazard analysis on new stereo data independently and fuses those hazard assessments.

Because the VCE enables Visual Odometry to be performed with next to no additional expense beyond the same stereo calculation used for hazard assessment, M2020's ENav algorithm can reliably fuse range data into a continually updated elevation map and to perform hazard assessment directly on that fused map.

Similarly, the comparatively low resolution of the stereo processing employed by MER and by MSL is more likely to lead to ranging errors that would complicate a direct assessment of the clearance between the rover and the terrain or hazard height. For that and computational reasons, GESTALT, like its MORPHIN predecessor, relies on a series of plane-fit residuals and slopes as the basis for its hazard assessment. [13]

The decision to treat the rover as an orientation-independent disk on previous missions was another design decision likely influenced by the limited computational resources available, but it also flows naturally from the decision to temporally fuse hazard maps rather than elevation maps.

In constructing a new navigation algorithm for a rover with comparatively fast stereo and visual odometry, maintenance of a 2.5D elevation map as the core structure was natural. But we had a critical need for a hazard assessment approach that would fit within the still very limited computational power of the RAD750 while avoiding being overly conservative. Note that porting portions of the GESTALT algorithm to the VCE's FPGA were considered early on but dismissed for multiple reasons, including: 1) the expense and schedule risk of a substantial FPGA development effort, 2) splitting the hazard detection and avoidance software between the VCE and the RCE would complicate the software architecture, and 3) the image processing functions consume nearly all of the available FPGA resources.

The solution adopted is the Approximate Clearance Evaluation (ACE) algorithm.[14] Here we simply outline the approach and describe its benefits in the context of M2020 mobility needs and architecture.

ACE is an elegant closed-form solution for obtaining conservative bounds on vehicle clearance, attitude and suspension angles. Given a particular rover position and heading, it examines the elevation data that lies in the potential footprint of each wheel and determines the min and max height of each wheel. From those heights it computes bounds on the vehicle attitude and suspension angles from which the minimum clearance between the belly pan of the rover and the underlying terrain can be derived.

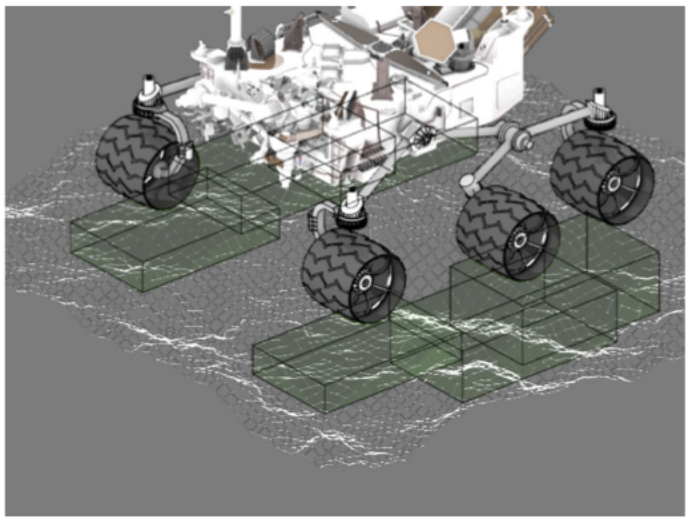

Figure 6. ACE wheel boxes define the portion of the terrain that determines the potential heights of each wheel.

ACE offers principled conservativism. It provides hard bounds on critical metrics of vehicle safety while being free of heuristics, ad hoc criteria weightings or approximations whose conservatism is uncertain. Indeed, key attributes such as tilt and suspension limits assessed by the vehicle's reactive safety checks are directly assessed by ACE. And its closed form nature provides the necessary computational efficiency.

ENav varies the size and shape of the wheelboxes to ensure the safety of commanded motions despite uncertainty introduced by vehicle slip and unanticipated yaw disturbances. In particular, the wheelboxes are grown longitudinally and laterally based on an onboard slip model which takes into account the estimated terrain slopes along the path. Evaluation of terrain further along the proposed motion uses progressively larger wheelboxes to account for the greater pose uncertainty.

The many other elements of M2020's Enhanced Navigation algorithm will be described in detail elsewhere. But it marries a greedy multi-level path planner utilizing explicit measures of execution time together with safety bounds provided by its ACE core into a path planning system that is capable of navigating more challenging terrain than MSL, while fitting into the RCE's limited computational resources and providing greater assurances of vehicle safety than its predecessors.

CONCLUSION

The Mars 2020 mobility system achieves its goal of enabling autonomous navigation through difficult terrain at significantly faster traverse rates than its predecessor. It accomplishes this by 1) leveraging the image processing power provided by the Vision Compute Element 2) making maximal use of the limited compute cycles available through its support of Thinking-While-Driving and 3) employing the novel ENav autonomous path-planner which uses the computationally-efficient ACE algorithm to ensure the safety of the vehicle even in the face of control disturbances and localization uncertainty.

ACKNOWLEDGMENTS

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES

[1] S. Milkovich, R. Lange, et al, "Mars 2020 Surface Mission Performance Analysis: Part 1. Science Exploration and Sol Type Modeling," in *AIAA SPACE and Astronautics Forum and Exposition*, Orlando, FL, USA, Sept. 17-19, 2018.

[2] J. J. Biesiadecki, P. C. Leger, and M. W. Maimone, "Tradeoffs Between Directed and Autonomous Driving on the Mars Exploration Rovers," *The International Journal of Robotics Research,* vol. 26, no. 1, pp. 91-104, Jan. 2007.

[3] M. Golombek and D. Rapp, "Size-Frequency Distributions of Rocks on Mars and Earth Analog Sites: Implications for Future Landed Missions," *Journal of Geophysical Research*, vol. 102, no. e2, pp. 4117-4129, Feb. 1997.

[4] J. J. Biesiadecki and M. W. Maimone, "The Mars Exploration Rover Surface Mobility Flight Software Driving Ambition," in *2006 IEEE Aerospace Conference*, Big Sky, MT, USA, Mar. 4-11, 2006.

[5] J. N. Maki, C. M. McKinney, et al, "Enhanced Engineering Cameras (EECAMs) for the Mars 2020 Rover," in *3rd International Workshop on Instrumentation for Planetary Missions*, Pasadena, CA, USA, Oct. 24-27, 2016.

[6] S. B. Goldberg, M. W. Maimone, and L. Matthies, "Stereo Vision and Rover Navigation Software for Planetary Exploration," in *2002 IEEE Aerospace Conference*, Big Sky, MT, USA, Mar. 9-16, 2002.

[7] A. E. Johnson, S. B. Goldberg, Y. Cheng, and L. H. Matthies, "Robust and Efficient Stereo Feature Tracking for Visual Odometry" in *2008 IEEE International Conference on Robotics and Automation*, Pasadena, CA, USA, May 19-23, 2008.

[8] A. Rankin, M. Maimone, J. Biesiadecki, N. Patel, D. Levine, O. Toupet, “Driving Curiosity: Mars Rover Mobility Trends during the first Seven Years”, IEEE Aerospace Conference, March 2020.

[9] M. Maimone, Y. Cheng, L. Matthies, “Two Years of Visual Odometery on the Mars Exploration Rovers,” Journal of Field Robotics, Volume 24 number 3, special issue on Space Robotics, March 2007, 169 – 186.

[10] A. Johnson, S. Aaron, et al, "The Lander Vision System for Mars 2020 Entry Descent and Landing," in *AAS Guidance Navigation and Control Conference*, Breckenridge, CO, USA, Feb. 2-7, 2017.

[11] A. Howard, "Real-Time Stereo Visual Odometry for Autonomous Ground Vehicles," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, Sept. 22-26, 2008.

[12] C. Y. Villalpando, A. Morfopolous, L. Matthies, and S. Goldberg, "FPGA Implementation of Stereo Disparity with High Throughput for Mobility Applications," in *2011 IEEE Aerospace Conference*, Big Sky, MT, USA, Mar. 5-12, 2011.

[13] R. Simmons, L. Henriksen, L. Chrisman, and G. Whelan, "Obstacle Avoidance and Safeguarding for a Lunar Rover," in *AIAA Forum on Advanced Developments in Space Robotics*, Madison, WI, Aug. 1996.

[14] K. Otsu, G. Matherson, S. Ghosh, O. Toupet, and M. Ono, "Fast Approximate Clearance Evaluation for Rovers with Articulated Suspension Systems," *Journal of Field Robotics*, Jul. 2019.