HARDWARE ACCELERATION OF A VISUAL LOCALISATION SYSTEM ON THE SURFACE OF MARS

Daniel Townson[1], Niklaus Kamm[1], Mark Woods [1]

[1]SCISYS, 23 Clothier Rd., Bristol, BS4 5SS, UK, E-mail: daniel.townson@scisys.co.uk

[1]SCISYS, 23 Clothier Rd., Bristol, BS4 5SS, UK, E-mail: niklaus.kamm@scisys.co.uk

[1]SCISYS, 23 Clothier Rd., Bristol, BS4 5SS, UK, E-mail: mark.woods@scisys.co.uk

1 INTRODUCTION

SCISYS is experienced in producing resilient software implementations of critical space flight algorithms. Recent developments in space qualified FPGA technologies have enabled a range of new applications for hardware accelerated algorithms for space applications. Where previously data volumes and constraints on execution time, power budget and processing power would have made on board processing of data infeasible, new developments in FPGA technology now allow for more data handling to occur on board of the spacecraft, permitting for more flexibility in mission design and operation as well as more efficient utilization of the available bandwidth. The reduction in execution time and resource usage allows for more complex algorithms to be used in a wider range of use cases. One such case is for on board data processing such as visual odometry, which determines the location of a vehicle based on processing done on image data gathered by an on board camera. This paper presents the partial transfer of the VisLoc visual odometry algorithm for Mars to an FPGA and discusses a wider range of potential applications in space.

2 VISLOC VISUAL ODOMETRY

The accurate localization of a vehicle on the Martian surface is crucial in allowing for operation of the vehicle to continue while direct contact to earth is interrupted. Due to the limited amount of time communication with a spacecraft in Mars orbit is possible, determination of the vehicle's current position and attitude has to be carried out locally. This additionally allows for the limited bandwidth to be used more effectively, allowing for more scientific data to be transmitted.

Solutions for local navigation estimates in the past have included wheel odometry, which measures the rotation of the rover's wheels to extrapolate the distance travelled. This approach however is very terrain dependent, as things like wheel slippage or stuck wheels will steadily introduce an accumulating error or cause outright erroneous readings.

One possible solution to this is to make estimates based on visual data gathered by cameras on board of the vehicle. By making estimates based on the relative movement of the camera location from one frame to the subsequent frame, the location data is independent of the terrain, achieving a high degree of accuracy in determination of both position and attitude even in difficult conditions.

This is achieved via the extraction of features from a frame which represent highly recognizable pixels. These features are then matched with the identified features from the subsequent frame. When a pair of features is recognized, the difference in location from one frame to the next is used to geometrically infer the movement of the camera in between the frames. This means visual odometry can be performed even in challenging terrain as long as the image offers enough light and definition for features to be recognizable, and as long as the movement between frames is not so large that features have moved off the edge of the frame before the next image is taken.

The Visual Localisation flight software algorithm (VisLoc) was developed for the ExoMars rover. It is based on the core algorithm known as OVO (Oxford Visual Odometry), developed at the University of Oxford [1]. VisLoc was adapted over a number of projects to be a viable method of visual localization for Martian surface vehicles [2]. After subsequent further development by SCISYS as part of the European Space Agency's ExoMars Rover Mission the VisLoc algorithm reached a technology readiness level (TRL) of 8.

In this paper we discuss the results of a study investigating the integration of an FPGA board into the VisLoc algorithm to accelerate the execution time of VisLoc with the aim of achieving an execution frequency of 1Hz while maintaining full parity between the software based algorithm and its accelerated counterpart. This accelerated version of the algorithm would then be deployed on European Space Agency's Sample Fetching Rover (SFR), which intends to cover considerably larger distances than ExoMars in a similar timeframe [3].

3 FPGA INTEGRATION

Being part of a vehicle on Mars poses unique constraints for a visual localisation algorithm due to very limited availability of CPU and electrical power as well as memory. While the low velocity of a Martian vehicle can largely eliminate difficulties such as motion blur which make visual odometry difficult on earth, the Martian environment and Mars rovers offer their own set of unique additional challenges for visual localisation. Besides the lowered mass and power budgets, low image resolution of 512x512 pixels, stark shadows covering large parts of the image, rover parts protruding into the frame and the potential for dust storms can greatly hinder the processing of an image. Large parts of images are also covered by the sky, which is not suitable for feature detection. The algorithm must thus be able to process images in real time with minimal resources, while correctly identifying and mitigating a number of factors that could render parts of an image unusable.

This large amount of processing overhead involves a large number of mathematical operations which slow down execution of the algorithm, but can be greatly accelerated by making use of pipelining and parallel processing on FPGA hardware.

By demonstrating that partial porting of the algorithm’s functionality to an FPGA can result in significant reduction of execution time, we could provide a range of options allowing for a trade-off between resource usage and execution time.

Suitability for hardware acceleration was assessed on a case by case basis for individual software modules of the algorithm. Factors such as sequential logic and dependencies within the process could prevent pipelining and parallelization, making modules less suitable for hardware acceleration. Additionally, the limitation in bandwidth of streaming interfaces between the CPU and FPGA introduced fixed data streaming delays before processing on the FPGA could commence, so the architectural design was adjusted to minimize streaming interfaces where possible.

Special care was given to ensure that outputs of the accelerated algorithm were equal to the software version in order to limit the effect on the TRL. This was validated using a series of simulated trajectories, such as the one in Figure 1 below, representing the expected conditions on the Martian surface, allowing for statistical analysis of results to 3-sigma accuracy. Parity of results was retained across the full validation dataset of the ExoMars mission.

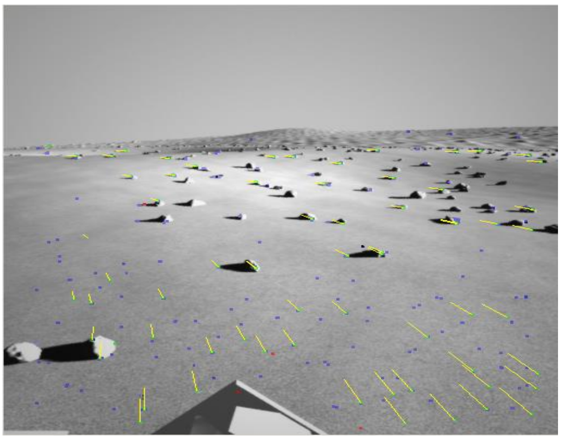

Figure 1: Example Frame of a generated Mars vehicle trajectory showing feature matches as yellow lines

Adding additional functionality to the FPGA then resulted in progressively diminishing returns, while increasing the required size of the FPGA. Due to the limited time frame of the study, the aim was to not exceed the capabilities of a medium BRAVE FPGA, which contains 35000 look-up tables (LUTs) [4]. This led to a total of eleven functions being transferred to hardware.

While further transfer of functionality to the FPGA, potentially even to the extent of fully running the algorithm on the hardware, is theoretically possible, it was not deemed feasible to achieve within the time constraints and it would have required considerable changes to the source algorithm, potentially affecting the accuracy of the resulting data in the process.

4 TESTING AND VALIDATION

VisLoc was thoroughly tested on both simulated and real Mars representative trajectories. Real trajectories were taken from the SEEKER trial in the Atacama Desert in Chile, where a representative rover moved through a Mars-like environment autonomously for several kilometers [5]. Simulated trajectories were either produces by SCISYS or provided by Airbus. The SEEKER trajectories allowed for testing under very realistic circumstances, providing real camera images from a representative vehicle in a very close approximation of the operational environment on Mars. Simulated trajectories meanwhile were used for targeted testing of specific scenarios, such as heavy shadows and low texture ground materials as shown in Figure 2 below.

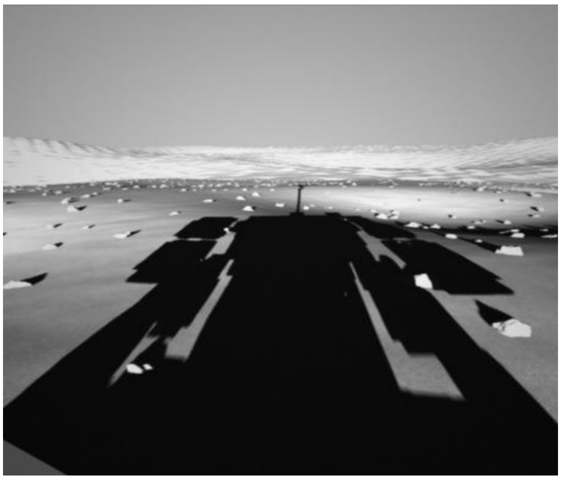

Figure 2: Heavily shadowed simulated image

After verifying that VisLoc was capable of meeting its requirement of an accuracy of 1% of total distance travelled in both real and simulated trajectories, the algorithm was tested under extreme conditions to determine the limitations of environments that were still sufficient for accurate visual localisation.

Individual test cases were created investigating the effects of optical depth, camera shutter speed and vehicle velocity.

VisLoc was found to be still within requirement accuracy for vehicle velocities of up to 16cm/s, double the intended velocity of the Mars Sample Fetch Rover. It was also capable of handling optical depth values of up to 2.6, simulating severe dust storms or very dark dusk or dawn conditions using images such as the one in Figure 3. Shutter speed was the most significant indicator of result accuracy, with large variance between test values ranging from 10ms to 150ms.

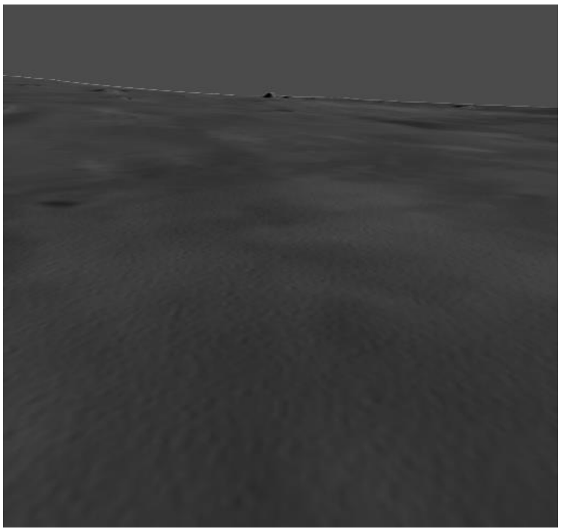

Figure 3: Simulated image at optical depth 2.6

5 PERFORMANCE ANALYSIS

During the original development of the VisLoc algorithm it was to be run on board of the ExoMars rover which was equipped with a LEON2 processor running at 96MHz. This study examined the suitability of the algorithm for the European Sample Fetching Rover (SFR), which is intended to be running on a LEON4 processor at 250MHz. Since test results were only available for execution times when using a LEON2 processor, times for a LEON4 processor were estimated assuming a roughly linear correlation between processor frequency and execution time.

The software implementation of the VisLoc algorithm had a worst case execution time of 3772ms on a LEON2 processor, which results in an estimated time of 1448ms on a LEON4 processor. This suggests that with the upgraded processor frequency used on SFR, to reach an execution frequency of 1Hz, a reduction of execution time of roughly 31% or 448ms is necessary.

Figure 4 below shows the reduction of the worst case execution time as algorithm functions are gradually transferred to the FPGA on both processors. The order of the transferred functions is in this case not sequential but instead functions were prioritized based on their suitability for the FPGA and their overall execution time during a single instance of the algorithm.

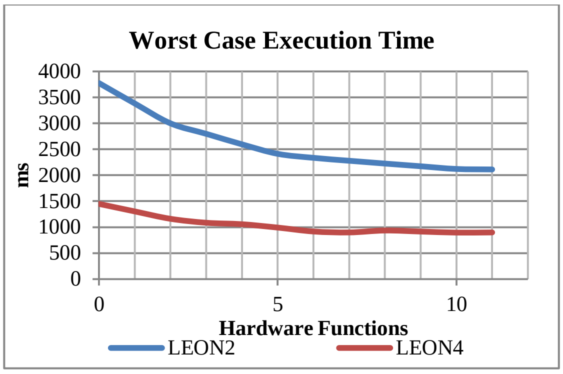

Figure 4: Worst case execution time with a variable number of hardware functions

The implementation of the major data processing steps of the algorithm resulted in a significant reduction of overall execution time, with the transfer of only five functions resulting in an overall reduction of execution time of 36.08% on the LEON2 processor.

The transfer of the full set of functions resulted in a total acceleration of 1661ms or 44.03% for a LEON2 processor. Applying hardware acceleration to the same functions on a LEON4 processor would result in a gain of 653ms or 45.09%, showing that FPGA acceleration could reduce execution time regardless of the processor clock speed when viewed relative to the overall execution time of the algorithm and would be sufficient to achieve an execution frequency of 1Hz. However the actual yield of the hardware acceleration as a percentage of the software execution time for the same functions is diminishing as the clock speed of the processor increases. The execution time of accelerated functions on LEON2 was reduced by 88.24% while on the LEON4 processor the reduction was only 74.68%. This would suggest that given a sufficiently high processor frequency, depending on the individual mission constraints, hardware acceleration may not be needed for the algorithm to perform at a frequency that is sufficient for navigation on Mars.

6 OTHER APPLICATIONS

In parallel with the evaluation of the original OVO VO algorithms the autonomy and robotics team at SCISYS also continued their development of science autonomy techniques [6]. These were integrated and tested with SCISYS’ GNC system (including VO) in representative field tests in the Atacama [7]. During these trials several kilometers of autonomous driving data was collected and subsequently analyzed. This allowed for identification of potential areas of functional overlap and acceleration via an FPGA. Due to the large amount of data that has to be processed during machine learning applications, the potential acceleration provided by an FPGA is very attractive. These efforts have now been taken further in SCISYS’ ESA Novelty or Anomaly Hunter (NOAH) activity, which strives to process image data on board the vehicle and autonomously detect potentially scientifically interesting novelties or anomalies [8]. More generally, FPGAs could be used in a variety of scenarios where sensors provide a volume of data that is not suitable for transmission with the bandwidth restrictions faced by most spacecraft. Pre-processing the data on-board, both by selection and prioritization via filters or machine learning, or by compression can greatly increase the efficiency of bandwidth usage, and the time and resources required to perform it can potentially be greatly reduced by using FPGAs.

7 CONCLUSION

Over the course of the study it was determined that hardware acceleration using an FPGA can significantly reduce the overall execution time of a visual odometry algorithm, allowing for processing of sufficient data to navigate accurately across extensive distances even at speeds far greater than current generation Mars rovers. Transfer of algorithm functionality to hardware can be performed in a modular manner while maintaining navigation accuracy and can still yield significant acceleration, as long as software modules are previously scrutinized for hardware suitability. A fixed delay created by the bandwidth of streaming interfaces between CPU and FPGA hardware introduces a latency into the system. Given a modular design, modules should thus be chosen in a manner that minimizes the need for transfer of large data volumes between software and hardware. Given a sufficiently high processor frequency, hardware acceleration and the associated reduction in TRL of the new platform may become unnecessary depending on specific mission constraints. In certain cases software execution may even outpace the hardware implementation of an individual module due to the head start granted by the lack of data streaming delay. In such a case, a modular approach to hardware acceleration may not be the best choice for the given use case, as a full software solution requires significantly less development time, while a full hardware solution will bring significantly higher reduction in execution time but requires an extended period of development and testing. Modular approaches are thus most suitable to algorithms that feature isolated sections of highly complex mathematical operations with limited dependencies which can be implemented on an FPGA with maximum yield in terms of execution time while minimizing the requirement for data exchange.

8 REFERENCES

[1] W. Churchill and P. Newman, “Experience Based Navigation: Theory, Practice and Implementation,” Oxford University, Oxford, 2012.

[2] M. Woods, A. Shaw, A. Gily and F. Didot, “High-level autonomy for long term exploration robotics,” in 11th Symposium on Advanced Space Technologies, Robotics and Automation, Noordwijk, 2011.

[3] A. Merlo, J. Larranaga and P. Falkner, “Sample Fetching Rover (SFR) for MSR,” in Advanced Space Technologies, Robotics and Automation, Noordwijk, 2013.

[4] NanoXplore, From Radiation Hardening To BRAVE FPGA devices, Geneva: CERN, 2017.

[5] M. Woods, A. Shaw, E. Tidey, B. V. Pham, L. Simon, R. Mukherji, B. Maddison, G. Cross, A. Kisdi, W. Tubby, G. Visentin and G. Chong, “Seeker - Autonomous Long-range Rover Navigation for Remote Exploration,” Journal of Field Robotics, pp. 940-968, 2014.

[6] M. Woods, A. Shaw, I. Wallace, M. Malinowski and P. Rendell, “Demonstrating Autonomous Mars Rover Science Operations in the Atacama Desert,” in I-SAIRAS, Sapporo, 2010.

[7] M. Woods, A. Shaw and I. Wallace, “Chameleon Field Trial: Toward Efficient, Terrain Sensitive Navigation,” in Advanced Space Technologies, Robotics and Automation, Noordwijk, 2015.

[8] N. Reads, M. Woods and S. Karachalios, “Novelty or Anomaly Hunter - Driving next-generation science autonomy with large high quality dataset collection,” in I-SAIRAS, Madrid, 2018.