Using a High-Fidelity Simulation Framework for Performance Singularity Identification and Testing

C. Scrapper, R. Madhavan and S. Balakirsky
Intelligent Systems Division
National Institute of Standards and Technology (NIST)
Gaithersburg, MD 20899-8230, U.S.A.
Email: {*chris.scrapper*, *raj.madhavan*, *stephen.balakirsky*}@*nist.gov*

Abstract

A common way to evaluate the performance of a system is to compare the algorithmic outputs with ground truth to identify divergences in the system's performance and discover the errors it is prone to. In the absence of such ground truth or as a follow-on to performance evaluation, performance analysis at the algorithmic level can provide developers insight into performance singularities. Such performance singularity identification and testing provides real-time meta-data that allows developers to understand the impact of singularities on the overall performance of the system. As an example of the concepts developed in this paper, we present a navigation solution based on image registration algorithms and the methodology used for the identification and testing of performance singularities of this algorithm.

1. Introduction

At the National Institute of Standards and Technology (NIST), we have been developing test methods to classify the performance characteristics of a system using quantitative metrics that facilitate the inter–comparison of experimental results. *De facto* standard testbeds provide a baseline that target specific aspects of the system, allowing researchers to assess the performance of various systems in different scenarios and environmental conditions.

A common way to evaluate the performance of a system is to compare the algorithmic outputs with ground truth to discover irregularities and artifacts that exist. In turn, these inconsistencies determined by such *performance evaluation* are used to identify divergences in the system's performance and discover the errors it is prone to. *Performance analysis* at the algorithmic level provides developers insight into performance singularities (which we define as the point where an algorithm fails to be well-behaved due to systematic and non-systematic errors). *Performance singularity identification and testing* provides real-time meta-data that allows developers to understand the impact of singularities on the overall performance of a system.

As an example of the concepts developed in this paper[1], we will present a navigation solution based on image registration algorithms and the methodology used for the identification and testing of performance singularities of this algorithm. The development of navigation solutions is motivated by urban search and rescue applications where a mobile robot is required to traverse undulating terrain and cope with unknown and unstructured environments. This research will use the baseline control framework, Mobility Open Architecture Simulation and Tools (MOAST) [3], and a high-fidelity simulation testbed, Unified System for Automation and Robot Simulation (USARSim) [5]. MOAST is an open-source control framework for a wide range of robotic systems in a variety of different domains. Since the methods employed to formulate a stable navigation solution are heavily dependent on the type of environment the system is operating in, the sensor capabilities, and the conditions found in that environment, it is critical for MOAST to employ redundant methods of pose estimation in order to develop a robust and stable navigation solution. USARSim is being used as a testbed to explore the performance characteristics of different navigation solutions.

This paper is structured as follows: Section 2

[1] Commercial equipment and materials are identified in this paper in order to adequately specify certain procedures. Such identification does not imply recommendation or endorsement by NIST, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

presents a brief overview of the high-fidelity simulation testbed and the baseline control framework. Section 3 details the principles of an iconic navigation solution that will be used as a basis to discuss the main theme of this paper. In Section 4 the results, analysis, and discussion of performance singularity identification and testing are provided. Section 5 concludes the paper by summarizing the findings and outlining our continuing work.

2 High-Fidelity Simulation Testbed

Robotic simulation systems, such as [1, 2, 4], are commonly used in the development of autonomous systems and advanced robotic algorithms. They provide a cost-effective tool that enables developers to customize repeatable testing scenarios to test specific aspects of autonomous navigation and mapping systems. In order to provide convincing arguments about a system's performance and reliability, the simulation systems must be capable of capturing the stochastic nature of a real world environment. USARSim [5] is an open-source package that provides a high-resolution, physics-based simulation that solves many of the practical problems faced by robotic simulators. Initially developed to support development of robotic algorithms in the urban search and rescue environment, USARSim has expanded its core functionality to provide a general-purpose, multi-agent simulation system with a set of unique characteristics unmatched by other simulation systems. One of the most important characteristics of this system is the robot and sensor models validation[13]. Significant efforts on the validation of simulated models in USARSim have resulted in close correspondence between simulated data extracted from USARSim and their real world counterparts [12, 9].

MOAST [3] is an open-source, turn-key hierarchical control system that was originally developed to promote the research of advanced robotic algorithms [11]. Based on the 4–D Real-time Control System (4D/RCS) architecture [6], MOAST provides a modularized hierarchical framework that allows for the transparent transference of data between a matrix of real and virtual components. This framework is glued together through well-defined interfaces, communications protocols, and detailed specifications on individual subsystem input/output (I/O) that allow developers to freely swap components. Internal tools provide developers with state-by-state, time-stamped snapshots that allow researchers to quantitatively measure and classify the performance characteristics of new algorithms and the means to analyze the overall impact on the system's performance by means of comparison.

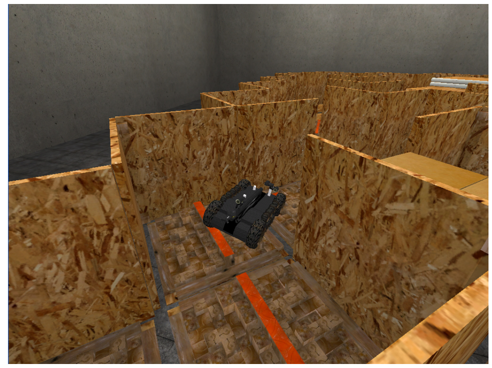


Figure 1. USARSim provides virtual access to existing test methods and robotic platforms that enables users to refine assumptions about the environment and explore novel sensor configurations.

Integration of USARSim's high-fidelity models with MOAST allows researchers to develop advanced robotic algorithms, classify their performance characteristics, and evaluate the overall impact of the algorithms on a robotic system before implementation on real robotic hardware. The availability of simulated reference test methods, as shown in Figure 1, allows researchers to refine assumptions about domains that they will eventually operate in. Due to space constraints, this section has presented only a brief overview of MOAST and USARSim. For further details on either of these systems, please refer to the system manual for MOAST [3] and USARSim [5].

3 Navigation Solutions using Visual Odometry

In this paper, the notion of a *navigation solution* will be defined as "the system's ability to sense the environment, create internal representations of its environment, and estimate pose, consisting of position and orientation, with respect to a fixed coordinate frame". The urban search and rescue environment presents an extremely harsh environment that does not guarantee static landmarks or the presence of geometric primitives used as reference markers in many navigation solutions. For such environments, we have employed an iconic approach, termed Visual Odometry (VO), that uses direct correlation of unprocessed data. Using unprocessed data will eliminate the need to define feature models and avoid misclassification due to imperfect sensor models. This technique, referred to in the literature as a scan matching technique, uses exteroceptive sensors which is an important intermediate step that will lead to an absolute navigation solution. Absolute navigation solutions have the advantage of being independent of the errors that arise in relative navigation solutions, thus providing a method for keeping the resulting errors bounded. Based on a *fine range image registration method*, known as Iterative Closest Point (ICP) algorithm [8], VO uses point-to-point correspondences in consecutive sets of data points obtained from a laser range finder (scans) to compute relative pose estimates. In its simplest form, this navigation solution computes these pose estimates using a maximum likelihood alignment to find the best fit between two sets of data points as shown below:

1. For each point in data set $\mathbf{D}$, compute its nearest neighbor in data set $\mathbf{M}$.
2. Compute the incremental transformation ($\mathbf{R}$,$\mathbf{T}$) using Singular Value Decomposition (SVD) based on correspondences obtained in step 1.
3. Apply the incremental transformation from step 2 to $\mathbf{D}$.
4. If relative changes in $\mathbf{R}$ and $\mathbf{T}$ are less than a predetermined threshold or a tolerable number of iterations is exceeded, terminate. Else go to step 1.

Recent improvements in the search strategy for finding data associations between the two sets of data has made this algorithm a computationally efficient way of generating navigation solutions in environments with minimal structure [10]. However, shortcomings in the basic ICP algorithm can lead to erroneous pose estimates, jeopardizing the integrity of the Visual Odometry algorithm [7]. In order to improve the convergence characteristics and to yield more accurate results, we need to implement techniques that will assist this algorithm to deal with spurious points/false matches and to account for occlusions and outliers produced by these shortcomings. Understanding such shortcomings of a particular algorithm can provide insight into the performance singularities that might arise in the navigation solutions and will provide insight into how to overcome these singularities. Below we identify two shortcomings of the VO algorithm and then discuss plausible solutions that will be used to evaluate the performance characteristics of the resulting navigation solutions in Section 4.

3.1 Data Association

Point-to-point data association used by the basic VO algorithm treats scan data as a set of discrete locations. This association of the data points is used to derive a transformation between the successive scans

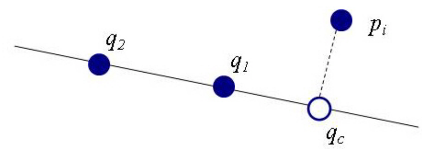


Figure 2. Pseudo point matching is a variant of a point-to-plane data association technique that is an alternative to point-to-point data association.

to estimate relative movement of the vehicle. However, the data in a scan represents a surface and not a set of discrete locations. This can lead to spurious point matching and can cause errors in the pose estimate. Pseudo point matching is a point-to-plane data association technique that approximates the real distance between a point and a plane [14] (shown in Figure 2). We establish correspondence with a virtual point, q_c, that is the closest point on the line defined by q_1 and q_2 using the relation:

$$q_c = q_1 + \frac{(p_i - q_1) \bullet (q_2 - q_1)}{\|q_2 - q_1\|^2}(q_2 - q_1)$$

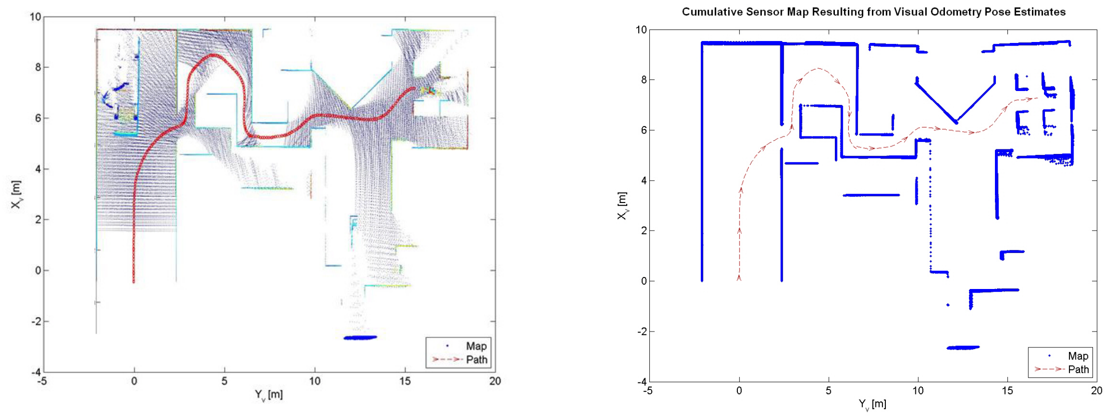
3.2 Thresholding

The least-square objective function used in the basic VO algorithm has no means to tackle uncertainties inherent in sensor data and to evaluate the validity of correspondences. This means that all correspondences between the data points are equally weighted. In order to deal with spurious points/false matches and to account for occlusions and outliers, we modify and weight the least-squares objective function such that:

$$min_{(\mathbf{R},\mathbf{T})} \sum_i w_i \, ||\mathbf{M}_i - (\mathbf{R}\mathbf{D}_i + \mathbf{T})||^2 \tag{1}$$

If the Euclidean distance between a point p_i in one set and its closest point q_i in the other, denoted by $d_i \triangleq d(p_i, q_i)$, is bigger than the maximum tolerable distance threshold $\mathcal{D}_{max}$, then w_i is set to zero in Equation (1). This means that an p_i cannot be paired with a q_i since the distance between reasonable pairs cannot be very big. The value of $\mathcal{D}_{max}$ can be set in one of two ways:

1. adaptively in a robust manner by analyzing distance statistics [14] or



(a) The use of the reference data sets that include ground truth of vehicle pose and visualization tools can provide developers with a map that can be used to better understand the nature of the environment that is being mapped.

(b) Visual Odometry produces pose estimates that can be used to create a cumulative sensor map. This map produces insight into the internal representations of a navigation solutions.

Figure 3. Comparisons of the ground truth maps with maps produced by navigation solutions is the first step in the performance identification and testing of navigation solutions. These comparisons can help identify areas where performance singularities arise.

2. statically using a user defined threshold

The threshold is implemented with respect to two observations: (a) If $\mathcal{D}_{max}$ is too small, then several iterations are required for the algorithm to converge and several good matches will be discarded, and (b) If $\mathcal{D}_{max}$ is too big, then the algorithm may not converge at all since many spurious matches will be included.

4 Performance Singularity Identification and Testing

Fundamental to the success of the performance singularity identification and testing is the development of reference data sets and visualization tools. Reference data sets allow for repeatable trials and the inter–comparison of the results between different navigation solutions. Visualization tools provide developers with a mechanism to gain insight into the nature of the environment and the internal representations contained in the navigation solutions.

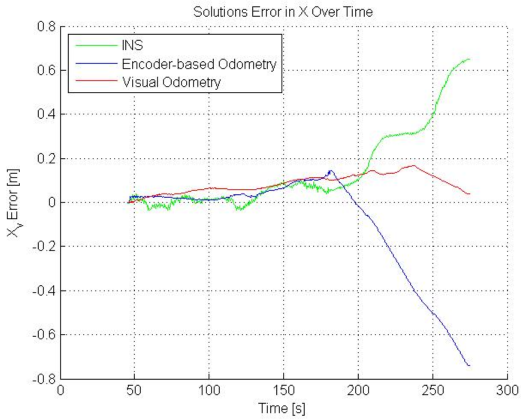
The reference data set used in this research, shown in Figure 3(a), was captured using MOAST to teleoperate a simulated P2AT[2] in an elemental mapping test world developed in USARSim to test autonomous mapping capabilities of the teams at the 2007 Virtual RoboRescue Competition. The `DM-Mapping_250` world (available on the USARSim home page) consists of several sections with varying degrees of complexity in terms of features and mobility characteristics. The data logged from this data set is used for the performance evaluation of navigation solutions.

In this text, a *cumulative sensor map* refers to a composite map consisting of raw sensor data mapped into a relative coordinate frame using the pose estimate from the navigation solution (no filtering data or pruning of the map). Close examination of the cumulative map produced by the Visual Odometry solutions, shown in Figure 3(b), illustrates the integrity and robust nature of an exteroceptive approach to formulating a navigation solution with only marginal errors being produced in the top-right corner of the map. The confined nature of this error suggests that a performance singularity occured during the second half of the run. The performance evaluation will assist us to locate where and when the singularity occured.

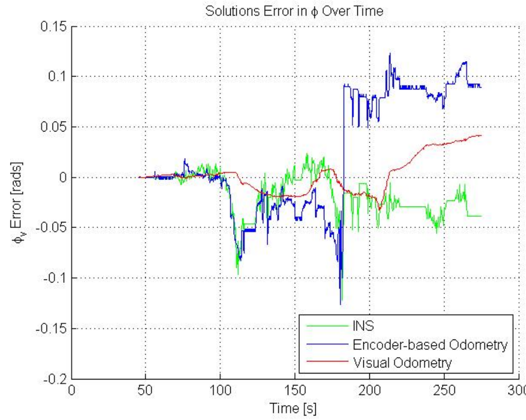
4.1 Performance Evaluation

Ground truth information of the vehicle's pose is essential to the quantitative performance evaluation of

[2]ActivMedia Pioneer 2-AT all-terrain robotic platform.



(a) Error plot shows the error in the X-axis with respect to the relative coordinate frame of theof the INS, Encoder-Based Odometry, and VO.



(b) Error plot shows the error in orientation of the INS, Encoder-Based Odometry, and VO.

Figure 4. Decomposing the translational and rotational errors in the individual components helps to pin-point areas where errors are arise in the navigation solutions. This can help identify singularities and provide insight into the errors that a specific navigation solution is prone to.

navigation solutions. It facilitates the decomposition of the errors arising in the navigation solution and shows the overall performance of different navigation solutions. Visual inspection of the cumulative map and the decomposition of the errors show where the navigation solution diverges and provides the tools to identify specific areas or situations where performance singularities lead to divergence. Examining Figures 4(a) and 4(b) suggests that the VO-based navigation solutions are more resilient to the systematic errors found in dead-reckoning sensors. However, these decompositions shows a significant spike in between 150 and 200 seconds. This spike, occuring about three-quarters into the run, suggests the presence of a performance singularity (e.g. hitting a wall) that may have led to the discontinuities observed in the cumulative sensor map for Visual Odometry.

4.2 Performance Analysis

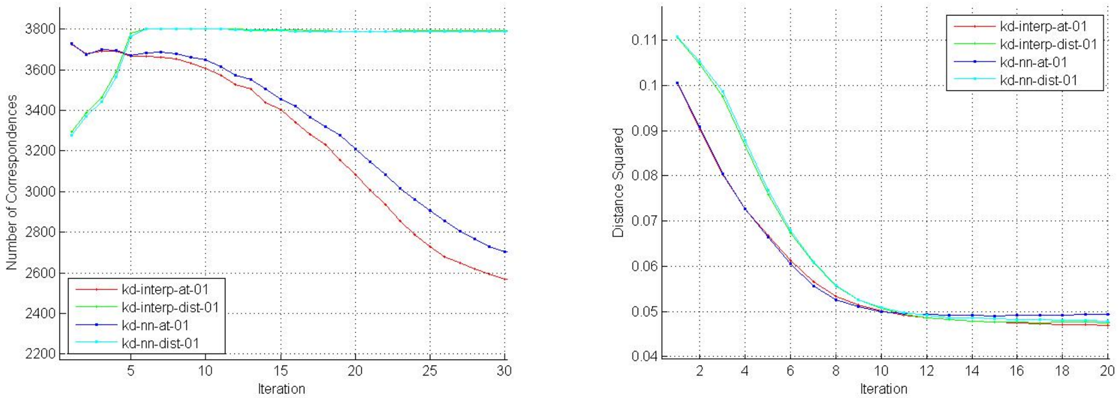
In order to yield more accurate results, it is important to analyze the convergence characteristics and how the obtained correspondences affect the performance of the VO algorithm. In this section, we examine four variants of the VO algorithm that use a permutation of data association and thresholding techniques discussed in Sections 3.1 and 3.2, respectively.

Analyzing the correspondences help developers to better understand the convergence characteristics of a given navigation solution. The rejection rate of the modified least-square objective function discussed in Section 3.2 can be derived from the number of valid correspondences found at each iteration as depicted in Figure 5(a). In this figure, notice that the number of correspondences found by the adaptive thresholding technique at each iteration is monotonically decreasing whereas those obtained by the static thresholding technique appear to be monotonically increasing thus effectively lowering the rejection rate to a minimum. As mentioned in Section 3.2, equally weighting all correspondences can lead to the inclusion of spurious points causing erroneous pose estimations.

The convergence profiles shown in Figure 5(b) provides an insight into how well the algorithms converge to a solution and ultimately how accurate those solutions are. This figure shows the VO algorithms that implement the adaptive thresholding technique converges quicker than the solutions using a static threshold. However, once the solutions start to converge to an estimate, the pseudo-point matching technique outperforms the point-to-point data association used in the basic VO algorithm.

5 Conclusions and Continuing Work

Performance evaluation of a system can be used to identify divergences in the system's performance and



(a) Correspondence analysis provides an insight in the quality of the registration techniques.

(b) Convergence characteristics shows how well and how accurately the Visual Odometry algorithms converges to a solution between two set of scans.

Figure 5. Performance analysis of the variants of the Visual Odometry solutions. The "nn" refers to point-to-point data associations, where the "interp" refers to pseudo-point matching discussed in Section 3.1. The "at" stands for adaptive thresholding, where the "dist" indicates that a static threshold was used, both discussed in Section 3.2.

discover the errors it is prone to by comparing algorithmic outputs with ground truth. As a follow-on process or in the absence of ground truth, performance analysis at the algorithmic level can provide developers insight into performance singularities. Such performance singularity identification and testing that provides real-time meta-data to understand the impact of singularities on the overall performance of the system was the main theme of this paper. We demonstrated some of these ideas using a high-fidelity simulation testbed in the context of navigation solutions for mobile robots.

Our continuing research seeks to develop test methods to classify the performance characteristics of navigation solutions that facilitate the inter-comparison of experimental results. The development of a *de facto standard* testbed for evaluation of navigation solutions will provide a baseline for comparison and the means to target specific aspects of the system, allowing researchers to assess the performance of various systems in different scenarios and environmental conditions.

References

[1] Delta3D. `www.delta3d.org`.
[2] Microsoft Robotics Studio. `msdn.microsoft.com/robotics`.
[3] MOAST. `sourceforge.net/projects/moast`.
[4] The Player Project. `playerstage.sourceforge.net`.
[5] USARSim. `sourceforge.net/projects/usarsim`.
[6] J. Albus and A. Meystel. *Engineering of Mind: An Introduction to the Science of Intelligent Systems*. John Wiley and Sons, Inc., 2001.
[7] T. Bailey. *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. PhD thesis, University of Sydney, 2001.
[8] P. J. Besl and N. D. McKay. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992.
[9] C. Pepper and S. Balakirsky and C. Scrapper. Robot Simulation Physics Validation. In *Proc. of the Perf. Metrics for Intell. Sys. Workshop*, August 2007.
[10] A. Nuchter, K. Lingeman, J. Hertzberg, and H. Surmann. 6D SLAM with Approximate Data Association. In *Proc. Intl. Conf. on Adv. Robotics*, July 2005.
[11] C. Scrapper, S. Balakirsky, and E. Messina. MOAST and USARSim: A Combined Framework for the Development and Testing of Autonomous Systems. In *Proc. SPIE*, volume 6230, 2006.
[12] J. Wang, M. Lewis, S. Hughes, M. Koes, and S. Carpin. Validating USARSim for Use in HRI Research. In *Proc. of the Human Factors and Ergonomics Soc. Annual Meeting*, 2005.
[13] M. Zaratti, M. Fratarcangeli, and L.Iocchi. A 3D Simulator of Multiple Legged Robots based on USARSim. In *Robocup 2006: Robot Soccer World Cup X, Springer, LNAI*, 2006.
[14] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vision*, 13(2):119–152, 1994.