https://doi.org/10.2352/ISSN.2470-1173.2020.6.IRIACV-325
© 2020, Society for Imaging Science and Technology

An Evaluation of Embedded GPU Systems for Visual SLAM Algorithms

Tao Peng, Dingnan Zhang, Don Lahiru Nirmal Hettiarachchi, John Loomis;
University of Dayton, Dayton, Ohio, United States

Abstract

Simultaneous Localization and Mapping (SLAM) solves the computational problem of estimating the location of a robot and the map of the environment. SLAM is widely used in the area of navigation, odometry, and mobile robot mapping. However, the performance and efficiency of the small industrial mobile robots and unmanned aerial vehicles (UAVs) are highly constrained to the battery capacity. Therefore, a mobile robot, especially a UAV, requires low power consumption while maintaining high performance. This paper demonstrates holistic and quantitative performance evaluations of embedded computing devices that run on the Nvidia Jetson platform. Evaluations are based on the execution of two state-of-the-art Visual SLAM algorithms, ORB-SLAM2 and OpenVSLAM, on Nvidia Jetson Nano, Nvidia Jetson TX2, and Nvidia Jetson Xavier.

Introduction

Automated Guided Vehicles (AGVs) have been used in the industry for many years. AGVs are mainly used to speed up various industrial processes and reduce industrial accidents in hazardous environments. Typically, AGVs use guide wires or a digital floor layout for navigation. Due to this reason, a road map of the floor plan needs to be designed in an optimum way for an AGV to perform efficiently. Even though AGVs are good for executing a specific task in a controlled environment, designing a road map for an AGV is a challenging task due to various reasons, such as different layouts, AGV specifications, and task prioritization [1]. Since AGVs road maps designed manually, it lacks any sense of unanticipated changes in the environment unless the specific change is pre-programmed to the map. That can lead to various performance issues that require more tasks to be pre-programmed towards task prioritization, according to variations in the environment.

Simultaneous Localization and Mapping (SLAM) solves the task prioritization problem by estimating the location of the robot and mapping the environment, simultaneously. Therefore, SLAM systems can perform given tasks independently with varying environmental conditions. Due to this reason, research studies predict that more than 15 million SLAM-enabled commercial and industrial robots, such as Autonomous Mobile Robots (AMRs), will be used by the year 2030 [2].

There are two common SLAM methods that are being used in most of the applications, LiDAR SLAM and Visual-SLAM [3] [4]. The LiDAR SLAM method uses LiDAR sensors, and the Visual SLAM method use cameras to extract features and map the environment [5]. In the past, LiDAR SLAM was more popular than Visual SLAM due to its high precision mapping capability over the Visual-SLAM method. This is mainly due to the lack of availability of high-quality cameras that can extract more features from the environment. However, in recent years, the Visual SLAM method started to draw considerable attention from the LiDAR SLAM due to the advancement in high-end camera development and the availability of such cameras at a low cost. Compared to 3D LiDAR SLAM, which uses expensive 3D LiDAR sensors, the Visual SLAM uses inexpensive cameras as sensors [5]. Even though the price of the 3D LiDAR sensors is at least 100 times higher than Visual SLAM camera sensors, Visual SLAM has a higher computational cost compared to LiDAR SLAM. Therefore, the main trade-off between the LiDAR SLAM and Visual SLAM is the sensor cost and higher computational cost [6].

This paper mainly focuses on state-of-the-art Visual SLAM methods such as ORB-SLAM2 [7] and OpenVSLAM [8]. Since Visual SLAM methods are computationally expensive because of the nature of local computation on embedded systems, mobile robots, especially UAVs, require maintaining a lower power consumption while maintaining high performance [9]. Therefore, choosing a development platform is an essential task. In March 2017, NVIDIA introduced an embedded development platform, the Jetson TX2, which is the game-changer of handling the state-of-the-art Visual SLAM algorithm like ORB-SLAM2. Since the Jetson platform can handle state-of-the-art Visual SLAM methods, three different devices, Nvidia Jetson Nano, Nvidia Jetson TX2, and Nvidia Jetson Xavier are selected to evaluate Visual SLAM methods. The two state-of-the-art Visual SLAM algorithms we used to test the hardware for the performance and efficiency are ORB-SLAM2 and OpenVSLAM. Comparing the results obtained by testing power consumption, metrics accuracy, and the processing frame rate on Nvidia Jetson embedded systems, it is shown that the Nvidia Jetson TX2 is the best suitable hardware for most Visual SLAM applications rated for industrial applications.

Following the introduction, the next section consists of an overview of the background and related work of Visual SLAM on embedded systems. Then a section that describes a review of NVIDIA Jetson hardware. The next two sections are dedicated to describing the methodology and obtained experimental results. The last section is dedicated to the conclusion and future work.

Background and Related Work

The ORB-SLAM2 is the state-of-the-art indirect visual SLAM algorithm. It is a complete SLAM system that works with monocular, stereo, and RGB-D cameras. It comes with map reuse, loop closing, and re-localization capabilities. It is designed to work in real-time in standard CPU. In the past four years, hundreds of research papers have been published related to ORB-SLAM2. ORB-SLAM2 works in a wide variety of environments, ranging from small hand-held sequences indoor to a car driven on the street. Inspired by PTAM (Parallel Tracking and Mapping), ORB-SLAM2 uses ORB (Oriented FAST and rotated BRIEF) feature for feature extraction, which performs better than PTAM's FAST corner detection, especially in rotation [10]. ORB-SLAM2 has three main parallel threads, as shown in Figure 1: tracking, local mapping, and loop closing. The tracking thread is considered to be the bottleneck of ORB-SLAM2 because it takes most of the time, and each new frame can not be processed until the current frame is completed [9] [11]. However, based on the nature of the FAST detection and ORB feature extraction, there are many tasks that can be parallelized and offload to GPU from CPU if the computing device has capable GPU on it like the Nvidia Jetson boards [12] [13].

Similar to ORB-SLAM2, the OpenVSLAM is an indirect Visual SLAM algorithm with sparse features, besides the difference in performance, it adds the applicability of a various type of camera models, such as fisheye and equi-rectangular. The fully modular design makes it more straightforward to be understood and modified. Compared to ORB-SLAM2, it can save and load maps, and localize new image based on the prebuilt maps. The OpenVSLAM also comes with the BSD 2-clause license. Compare to the GPL-V3 license of ORB-SLAM2, OpenVSLAM allows commercial use [8].

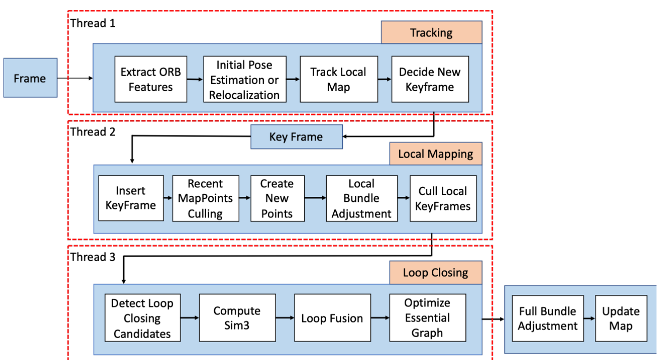


Figure 1. ORB-SLAM2

Hardware

This section introduces NVIDIA Jetson Systems in details. Also, the developed power meter to measure the power consumption in all systems while running SLAM algorithms.

NVIDIA Jetson Systems

The Nvidia Jetson platform is an AI platform for mainly autonomous navigation, introduced by Nvidia in 2014. Each Jetson module has complete system on it with CPU, GPU, DRAM and flash storage. It is extensible and can be easily customized for each application's requirement. Since the Jetson family of modules share the same architecture and SDKs, the deployment is seamless [14]. The Jetson Nano has Quad-core ARM processor and 128-core Nvidia Maxwell GPU, with 4GB LPDDR4 Memory. The Jetson TX2 has Quad-core ARM processor with Dual-core Denver CPU and 256-core Nvidia Pascal GPU, with 8GB LPDDR4 Memory. The Jetson Xavier has 8-core Nvidia Carmel ARMv8.2 CPU and 512-core Nvidia Volta GPU, with 16GB LPDDR4x Memory. For this study Nvidia Jetson family

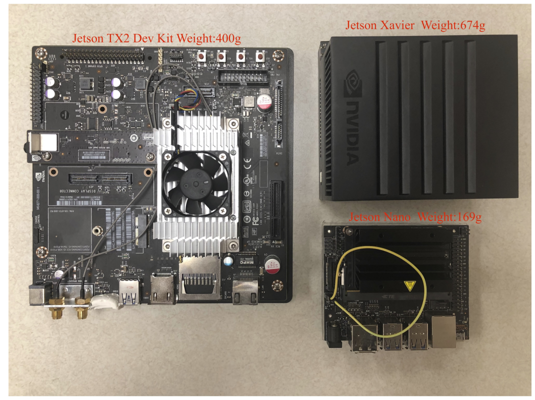


Figure 2. Nvidia Jetson Developer Kits

are selected due to the variety of specification offers by the Jetson systems. The detailed specifications are listed in Table 1.

Figure 2 shows a comparison in size and weight among three Nvidia Jetson developer kits. The Jetson Nano, which is an entry-level Jetson platform, has the lightest weight and size, it has a passive cooling system. Jetson TX2 developer kit is a TX2 module mounting on a very versatile carrier board that comes with many useful I/O for development. This is the mainstream Jetson board on the market now. The Jetson Xavier developer kit is a heavy metal box that weighs 674g, most of the weight is from its massive metal heat sink.

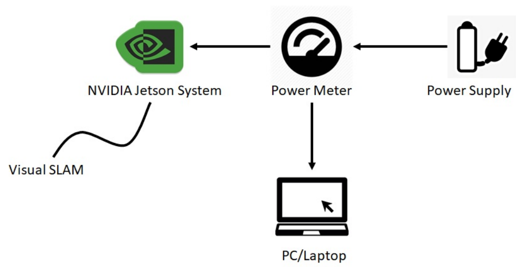
Power Meter

Tegrastats Utility is a convenient and powerful tool that comes with the Nvidia Jetson platform. It reports memory usage and processor usage for Jetson-based devices, also can monitoring the power consumption of the Jetson module but not the entire developer kit. The developer kit includes the Jetson module, the carrier board, and the other devices on the carrier board. The real power consumption is higher than the power consumption measured by Tegrastats Utility. This motivates us to create a physical power consumption meter to evaluate the overall power usage of the developer kit works with our mobile robot and UAV (UAV is more sensitive to the battery capacity and weight). The Architecture of the evaluation system is shown in Figure 3(a).

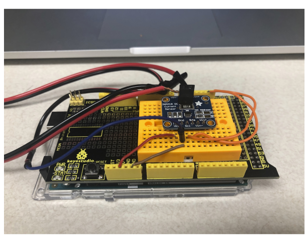
We made a versatile power meter for this project. This power meter consists of an INA219B breakout board and an Arduino Mega 2560 board, which is shown in Figure 3(b). The INA219 board is a current shunt and power monitor. It monitors the shunt voltage drop and bus supply voltage. Current in amperes can be easily read, and multiplying register will calculate power in watts. The measurement range of current and voltage is large enough for Jetson TX2, Jetson Xavier, and Jetson Nano. The power consumption of Jetson systems showed in Table 1 are theoretical maximum values. The power consumption of Jetson TX2, Jetson Xavier, and Jetson Nano are difficult to measure due to the Jetson modules are installed on the carrier boards because the other electronic components also generate power usage. Since the Jetson module cannot be used solely, measuring the real power consumption of the entire system including the carrier board is more meaningful for this project. We design a replaceable power jack, which is shown in Figure 3(b). Jetson TX2, Jetson Xavier and Jetson Nano have different power plugs. A replaceable power jack

Table 1. The comparison on specifications of NVIDIA Nano, Jetson TX2 and Xavier

	NVIDIA Nano	NVIDIA Jetson TX2	NVIDIA Xavier
CPU	Quad-core ARM Cortex-A57 MPCore processor	Dual-core Denver 1.5 64-bit CPU and quad-core ARM Cortex-A57 MPCore processor	8-core NVIDIA Carmel ARMv8.2 64-bit CPU
GPU	128-core NVIDIA Maxwell GPU	256-core NVIDIA Pascal GPU	512-core NVIDIAVolta GPU
Memory	4 GB 64-bit LPDDR4 25.6GB/s	8 GB 128-bit LPDDR4 59.7GB/s	16 GB 256-bit LPDDR4x 136.5GB/s
Storage	16 GB eMMC 5.1	32 GB eMMC 5.1	32GB eMMC 5.1
Ethernet	10/100/1000 BASE-T Ethernet	10/100/1000 BASE-T Ethernet WLAN	10/100/1000 BASE-T Ethernet
Power	5W / 10W	7.5W / 15W	10W / 15W / 30W
Price	129 USD	479 USD	999 USD



(a) The architecture of Evaluation System



(b) Power Meter

Figure 3. Evaluation System

can fit all of the power plugs. We use a computer connected to the power meter to record the real-time power consumption. This power data along with the time stamp helps the analysis of the power usage behavior of these Jetson boards.

Method

We set up a testing system for the evaluation of the power consumption, performance, and accuracy shows in Figure 3 (a): Nvidia Jetson board is running Ubuntu 18.04 with OpenCV 3 and CUDA 9 for testing ORB-SLAM2 and OpenVSLAM, a customized power meter, and a laptop for recording the power measurements and timestamps. Jetson Xavier, Jetson TX2, and Jetson Nano were used to compare the throughput and power efficiency, the specifications of these Jetson boards are shown in Table 1. A Logitech C920 UVC (USB video class) camera is used to test the real-time tracking performance [15]. It runs on fixed 720P, 30FPS. The results of real-time tracking FPS for all boards are shown in Figure 4.

Each Jetson board has a function called Dynamic Energy Profile, which allows users to enable or disable a certain number of CPU cores and limit the frequency or total power consumption. In our experiments, we tested several power settings for each board, but for a fair comparison, we compare the performance with the maximum power setting to study the bottleneck of the hardware. It is noteworthy that Jetson Nano supports up to 5V 4A through the Barrel Jack connector, if a jumper is put on the J48 header on the board. Also, Jetson Nano will limit its power consumption and the 4GB RAM is not enough for some applications. Therefore, we enabled the feature called Swapfile in Linux for Jetson Nano. The 8GB RAM in TX2 and 16GB RAM in Xavier is capable of the Visual SLAM algorithms we performed.

To evaluate the GPU acceleration enabled power consumption and pose estimation accuracy of OpenVSLAM and original ORB-SLAM2, we used the Technical University of Munich (TUM) RGB-D SLAM dataset for the experiments. It contains sequences from RGB-D sensors grouped in several categories to evaluate indoor scene reconstruction and visual odometry methods under different texture, illumination, and structure conditions [7]. This benchmark system also provides synchronized ground truth camera poses for the camera, recorded by a precise motion capture system [7] [16]. Then the benchmark system is used to evaluate the accuracy. Detailed discussion of evaluation test is available in the experimental results section.

Experimental Results

In [13] shows the GPU acceleration performance on the original ORB-SLAM2 algorithm. As shown in in Figure 4, for this study, we have combined the performance results obtained by the ORB-SLAM2 algorithm with performance results obtained with OpenVSLAM. The real-time tracking FPS for each algorithm and each board are shown in Figure 4. The OpenVSLAM has a slightly better performance than ORB-SLAM2, but on Jetson Nano it is still below 10FPS. Note that the OpenVSLAM has optimization for the X86 processor, however, optimization is disabled since the Nvidia Jetson hardware are based on ARM processors [8].

From the test of one sequence in the TUM RGB-D SLAM dataset, we generated a plot, which is shown in Figure 5. The Figure 5 shows the energy cost of processing each frame on Jetson Nano, TX2, and Xavier. For a fair comparison, we only list the power efficiency chart with one sequence with the maximum power setting. When comparing the power efficiency, it is clear that the TX2 is the overall winner among all boards. It is worth mentioning that the TX2 developer kit we test boots from an external SSD, which consumes additional power, therefore the real energy cost of TX2 should be less.

With the evaluation of performance and power consumption, accuracy is also evaluated in our experiments [17]. The Abso-

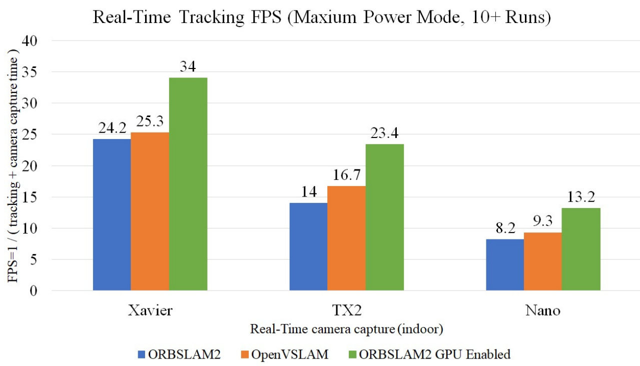


Figure 4. Real-Time Tracking FPS

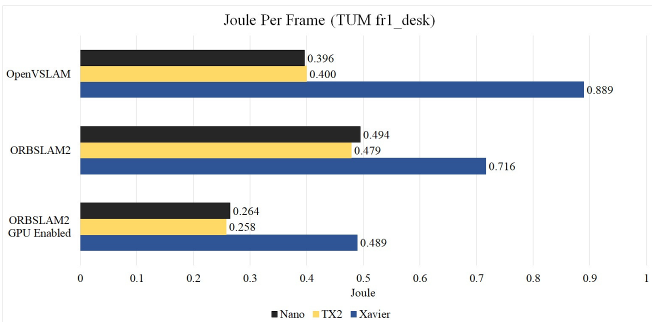
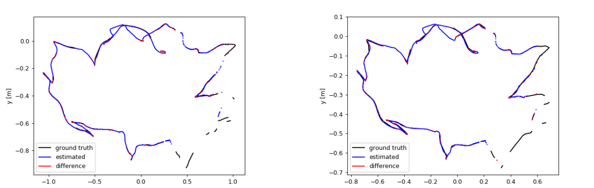
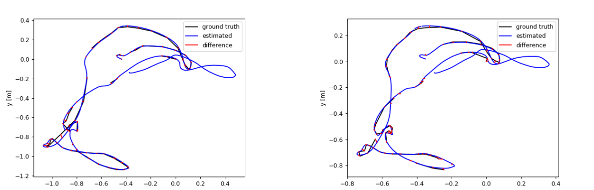


Figure 5. Power Efficiency

lute Trajectory Error is used for the evaluation. We did not find any significant difference in accuracy among the original ORB-SLAM2, ORB-SLAM2 with GPU acceleration, and OpenVSLAM. Also, in [8] it shows the accuracy of OpenVSLAM is similar to ORB-SLAM2 in most of the sequences of EuRoC MAV dataset (monocular). We also did the test on the TUM RGB-D SLAM dataset. The results is shown in Figure 6.



(a) fr2_desk_ORB_SLAM2_CPU (b) fr2_desk_ORB_SLAM2_GPU



(c) fr1_desk_ORB_SLAM2_CPU (d) fr1_desk_OpenVSLAM_CPU

Figure 6. Estimated trajectory (blue) and groundtruth (black) in TUM RGB-D dataset

Conclusion and Future Work

We presented the study of the integration for ORB-SLAM2 and OpenVSLAM on the embedded GPU systems: Nvidia Jetson Nano, Jetson TX2, and Jetson Xavier. Both of the algorithms work well on desktops but not on these low-power embedded systems, especially Jetson Nano. The Jetson Nano can not reach our real-time embedded SLAM goal (at least 10fps) without proper tuning by enabling GPU parallelism. The improvement from using GPU to offload tracking thread from CPU is more significant for Jetson Nano than others because it has a relatively weak CPU on it. The OpenVSLAM has better performance compare to ORB-SLAM2 because of its optimization based on ORB-SLAM2.

In this project we did not test the performance of enabling SIMD (Single instruction, multiple data) on Jetson-family boards with ARM processors, from our experiments on PC with X86 CPU we have a 2-3% improvement in performance. However, the current simulation result shows the outstanding potential of starting new Visual SLAM applications with OpenVSLAM.

From our results, we consider Jetson TX2 has an overall best balance for performance and power consumption for Visual SLAM applications in industrial use. Nvidia also released TX2i 8GB, which is a rugged version TX2 for industrial robot use. Also, it comes with 8GB RAM (with ECC support). The operation life of TX2i is 10 years 24/7, which is twice as TX2. As an entry-level Jetson platform, Jetson Nano can barely handle the real-time Visual SLAM applications without proper tuning. On the other hand, the Jetson Xavier is capable of handling basic Visual SLAM applications but with a high cost of power consumption.

For future work, ARM NEON (a packed SIMD architecture for ARM-based processors) optimizations could also help to improve the performance. Similar to GPU acceleration of the ORB-SLAM2, re-implementation of feature extraction, feature matching, PnP pose estimation for OpenVSLAM can boost the system. We are currently testing a fisheye camera setup with OpenVSLAM.

References

[1] S. Uttendorf, B. Eilert, and L. Overmeyer, "A fuzzy logic expert system for the automated generation of roadmaps for automated guided vehicle systems," in *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Dec 2016, pp. 977–981.

[2] A. Research. (2019, aug) Slam software paving the path for a new generation of industrial robots - amrs. [Online]. Available: https://www.abiresearch.com/press/slam-software-paving-path-new-generation-industrial-robots-amrs/

[3] P. Beinschob and C. Reinke, "Graph slam based mapping for agv localization in large-scale warehouses," in *2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, Sep. 2015, pp. 245–248.

[4] A. Pfrunder, P. V. K. Borges, A. R. Romero, G. Catt, and A. Elfes, "Real-time autonomous ground vehicle navigation in heterogeneous environments using a 3d lidar," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 2601–2608.

[5] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, Dec 2016.

[6] M. Filipenko and I. Afanasyev, "Comparison of various

slam systems for mobile robot in an indoor environment," in *2018 International Conference on Intelligent Systems (IS)*, Sep. 2018, pp. 400–407.

[7] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, Oct 2017.

[8] S. Sumikura, M. Shibuya, and K. Sakurada, "OpenVSLAM: A Versatile Visual SLAM Framework," in *Proceedings of the 27th ACM International Conference on Multimedia*, ser. MM '19. New York, NY, USA: ACM, 2019, pp. 2292–2295. [Online]. Available: http://doi.acm.org/10.1145/3343031.3350539

[9] O. Ulusel, C. Picardo, C. B. Harris, S. Reda, and R. I. Bahar, "Hardware acceleration of feature detection and description algorithms on low-power embedded platforms," in *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*, Aug 2016, pp. 1–9.

[10] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, Nov 2007, pp. 225–234.

[11] D. Bourque, "Cuda-accelerated orb-slam for uavs," 2017.

[12] M. N. Rud and A. R. Pantiykchin, "Development of gpu-accelerated localization system for autonomous mobile robot," in *2014 International Conference on Mechanical Engineering, Automation and Control Systems (MEACS)*, Oct 2014, pp. 1–4.

[13] *Evaluating the Power Efficiency of Visual SLAM on Embedded GPU Systems*, 2019.

[14] N. Corp. (2020, feb) Embedded systems for next-generation autonomous machines. [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/

[15] R. Liu, T. Peng, V. K.Asari, and J. S. Loomis, "Real-time 3d scene reconstruction and localization with surface optimization," in *NAECON 2018 - IEEE National Aerospace and Electronics Conference*, July 2018, pp. 280–285.

[16] N. Otterness, M. Yang, S. Rust, E. Park, J. H. Anderson, F. D. Smith, A. Berg, and S. Wang, "An evaluation of the nvidia tx1 for supporting real-time computer-vision workloads," in *2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, April 2017, pp. 353–364.

[17] L. Nardi, B. Bodin, M. Z. Zia, J. Mawer, A. Nisbet, P. H. J. Kelly, A. J. Davison, M. Luján, M. F. P. O'Boyle, G. Riley, N. Topham, and S. Furber, "Introducing slambench, a performance and accuracy benchmarking methodology for slam," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 5783–5790.

Author Biography

Tao Peng received his BS in Photoelectric Information Science and Engineering from Nanjing University of Science and Technology (2011) and his MS in Electrical Engineering from University of Dayton (2013), he is pursuing his PhD in Electrical Engineering in University of Dayton. His research interests are Application of Simultaneous Localization and Mapping (SLAM), computer vision.

Dingnan Zhang received his BS in Automation Engineering at Shenyang Ligong University (2012) and his MS in Electrical Engineering from University of Dayton (2015). He is pursuing his PhD in Electrical Engineering in University of Dayton. His research interest in computer vision.

Don Lahiru Nirmal Hettiarachchi received his BSc special degree in Engineering Physics from University of Colombo, Sri Lanka (2014) and his MS in Electrical Engineering from University of Dayton (2016). Currently, he is pursuing PhD in Electrical Engineering in University of Dayton. His research interests are hardware acceleration using FPGA and GPGPU devices, computer vision, machine learning, and deep learning.

John Loomis received his BS in Physics from Case Western Reserve University (1966), his MS in Physics from the University of Illinois (1968) and MS in Optical Sciences from the University of Arizona (1977), and his PhD in Optical Sciences from University of Arizona (1980). He is a Professor Emeritus at the University of Dayton.

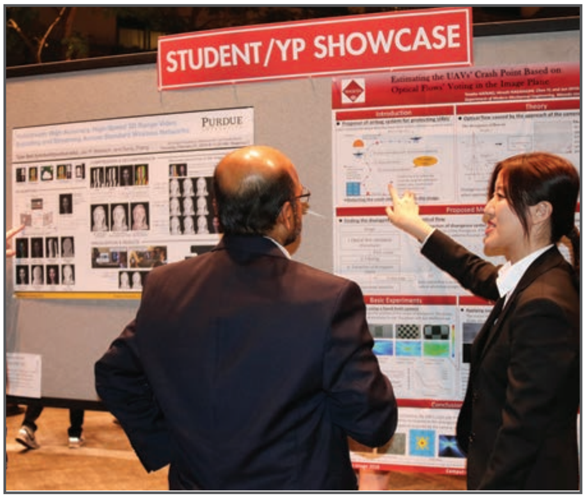
JOIN US AT THE NEXT EI!

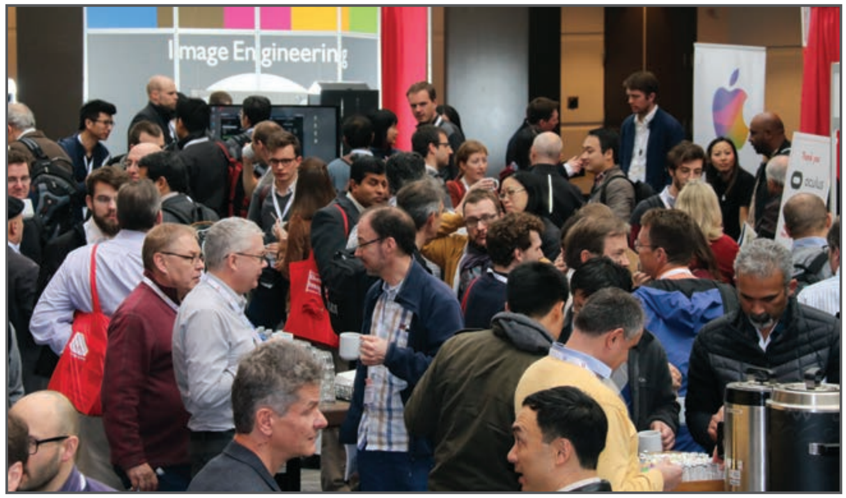
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org