

DevCAM: An Open-Source Multi-Camera Development System for Embedded Vision

Akhil M. Birlangi¹, Dominique E. Meyer¹, and Falko Kuester¹

¹Dronelab, University of California San Diego; La Jolla, CA

Abstract

Computer vision algorithms are often challenging to implement in hardware due to integration time and system complexity, while commercial systems often prevent low-level image processing customization and hardware optimization due to the largely proprietary nature of the algorithms and architectures, hindering research development by the larger community. This work presents DevCAM- an open-source multi-camera environment, targeted at hardware-software research for vision systems and specifically for co-located multi-sensor processor systems. DevCAM seeks to facilitate the integration of multiple late-generation sensors, abstracting interfacing difficulties to high-bandwidth sensors, enable user defined hybrid processing architectures on FPGA, CPU and GPU, and to unite multi-module systems with networking and high-speed storage storage. The system architecture can accommodate up to six 4-lane MIPI sensor modules which are electronically synchronized, alongside support for an RTK-GPS receiver and a 9-axis IMU. We demonstrate a number of available configurations that can be achieved for stereo, quadocular, 360, and light-field image acquisition tasks. The development framework includes mechanical, PCB, FPGA and software components for the rapid integration into any system. System capabilities are demonstrated with the focus on opening new research frontiers such as distributed edge processing, inter system synchronization, sensor synchronization, and hybrid hardware acceleration of image processing tasks.

Introduction

This work spans a set of evolving fields within the electronic imaging community: image signal processors (ISP), multi-view reconstruction, hardware acceleration, image/video compression, visual-inertial Simultaneous Localization and Mapping (SLAM), robotics, light-field arrays, multi-agent robotics, and robotic swarms. Across these fields, there has been a consistently growing need to evaluate the latest algorithms and techniques in embedded scenarios. However, most of these novel methods are often tested and developed on powerful desktop workstations[3, 14], unable to be easily run at real-time speeds on conventional embedded hardware, and generally not designed for embedded environments. This limitation is two-fold, algorithms are commonly implemented with the goal to achieve superior qualitative results compared to state of the art rather than to perform within a deployed system. As a result, high quality algorithms often must be altered and simplified to be used on conventional embedded hardware at the cost of accuracy, resolution, and speed.

This is reflected directly in the density of data in most embedded deployments of algorithms. In many prior works, embed-

ded camera systems are limited in resolution to about 720p, forced to lower frame-rate, or rely on image compression [3] due to hardware limitations. The lack of high bandwidth data offloading or dense local storage combined with the low compute capability of conventional embedded hardware forced many systems to utilize a separate laptop or desktop to offload and process data off device [13]. As such, there is a need for co-development of deployment oriented algorithms and hardware to facilitate algorithm deployment.

In this paper, we present DevCAM: an open source, application agnostic, multi-camera, field programmable gate array (FPGA) centered embedded vision platform. The platform spans both a hardware reference design set for printed circuit boards (PCBs) and opto-mechanical configurations, as well as a set of software configuration tools for facilitated porting of vision algorithms to the hardware. Inherently, this facilitates researchers and industry to leverage the flexibility inherent to FPGAs combined with generous programmable logic (PL) resources and high level synthesis (HLS) toolkits to enable a vast array of existing algorithms to be deployed on the system. The DevCAM open-source project provides a fully customizable stack of rich state of the art capabilities, enabling advanced embedded vision research into hardware optimized algorithms, synchronized multi-camera systems, and even multi-agent computation. The primary goals for the system are as follows:

- 1) **Hardware References:** Enabling rapid development of time-intensive designs for rapid prototyping and development.
- 2) **Extensibility:** Multi-camera support, IMU, GPS and syn-

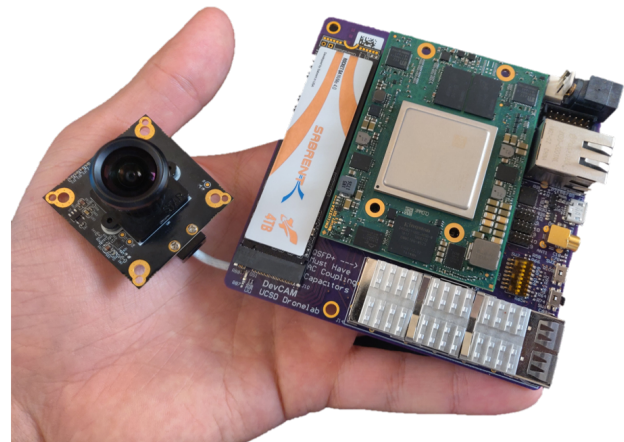


Figure 1: The DevCAM FPGA V1.0 hardware-software development system with a single IMX577 camera module

chronization enable extended capability in centralized and de-centralized multi-system visual-inertial development.

- 3) **Customizability:** From ISP to multi-view and Convolutional Neural Network (CNN) algorithms, the platform allows for easy customization, acceleration and porting to hardware.
- 4) **Standardization:** Documentation and support for ease of Open-Source development.

Motivation and Novelty

Within the fields of electronic imaging, hardware platforms are often built top down as opposed to bottom up. Generally, systems are designed and built out to test and implement algorithms and as such tend to support just the functionality needed and not much more. This leads to embedded systems that are constrained to their specific application and unable to adapt to the rapidly changing research landscape. In order to bolster research capabilities and maximize hardware re-usability, we set out to develop a compact, application-agnostic, embedded vision research platform. DevCAM meets and expands upon many of the current state of the art requirements for an embedded vision platform.

When it comes to embedded vision, image quality and information density play a large factor in many modern algorithms. Higher resolution, frame-rate, and image quality provide more data for algorithms to utilize in order to bring about higher quality results and better algorithms. However, in embedded scenarios, power and compute availability often force deployed algorithms to reduce the quality and size of the data and thereby impacting the quality of results. DevCAM targets this issue by leveraging the flexibility and power efficiency of FPGAs. DevCAM leverages the XCZU15EG, a large, resource plenty FPGA System-on-chip (SoC). When compared to similar modern platforms [19], DevCAM has approximately 1.25x more Look Up Tables (LUTs) and Flip-Flops (FF), and nearly 1.8x the amount of RAM (BRAM/Block RAM and URAM/UltraRAM) available. This increase in PL resources enables DevCAM to support larger image resolutions, perform image preprocessing, and compute some algorithmic results all as part of the streaming capture pipeline. FPGAs also provide large benefits in terms of relative power efficiency when compared to GPU based embedded systems and embedded systems that rely on off-board computation via a laptop or desktop computers. All of this combined with a growing repository of software tools to easily port over algorithms to FPGA hardware makes DevCAM a highly application agnostic embedded vision platform.

Over the past few years, there has been a surge of multi-view algorithms that leverage images taken from multiple cam-

eras. Multi-camera vision systems have many advantages, such as higher combined field of view and greater 3D accuracy. Along with this, multiple *synchronized* cameras can provide large benefits for accurate 3D reconstruction. As such, the DevCAM system architecture supports six 4-lane MIPI CSI-2 cameras streaming directly into the programmable logic for custom capture pipelines and vision applications. All of the camera GPIO and interface pins are also routed into the Programmable Logic (PL). This enables gate level, tuneable, custom synchronization of all 6 camera ports via PL triggered external trigger pins or synchronization signals.

With six cameras worth of high bandwidth image data streaming in at high frame rates, it may be difficult for even a large FPGA to perform all of the computations needed. In this case, if the DevCAM system is primarily being used as a high density data capturing sensor platform, the most important function is to collect and quickly offload data as to not become the bottleneck for downstream processing resources. To facilitate this need for high output bandwidth, DevCAM is outfitted with a 40Gb Ethernet port wired directly into the PL to enable fast streaming off-board without any processor intervention in the pipeline allowing for a low bottlenecks system design. In scenarios where optical fiber is unavailable, DevCAM also has an onboard M.2 slot for NVMe SSDs to provide a fast local storage option.

The abundance of hardware computation resources plays an important role in visual-inertial systems. Specifically, accurate pose estimation and world-centered localization are of crucial concern for UAV and robotic agents. Applications such as SLAM, reconstruction, and many others rely on pose estimates of the cameras when images are taken to work properly. Along with this, for 3D reconstruction and photogrammetry applications, it can be very useful to work in world-centric coordinates and overlay scans with real world locations using GPS landmarks for AR visualization or accurate surveying. To enable research applications in these domains, DevCAM is fitted with a TDK ICM-20948 and a ZED-F9P RTK-GPS for obtaining sensor poses and accurate GPS locations.

DevCAM serves as a synthesis of the primary functions used by many embedded vision projects and acts as a testbed that encourages novel interdisciplinary research opens the capability of dense high bandwidth data collection and computation to many fields in electronic imaging.

Technical Design Process

The DevCAM system implements a carrier board for the Enclustra Mercury+ XU1 SOMs. As such the system architecture uses Enclustra's own larger carrier boards as a reference design. However, for smaller footprint optimization, DevCAM targeted a smaller form factor of 100mm x 100mm with a 10 PCB layers.

Along with the high layer count increasing complexity, there are many differential pairs on the system. Each MIPI port contains a pair for each lane and one more for the clock. These pairs must be length matched with each-other in order to prevent skew between pairs and each line of each pair must be length matched to avoid skew between the P/N lines. The trace width and spacing between the P and N lines of each pair must also be carefully controlled to ensure that the route has a differential impedance of 100Ω to avoid noise on the line due to reflections. All of the above is true for the 40GbE as well adding on 8 pairs for its RX and TX

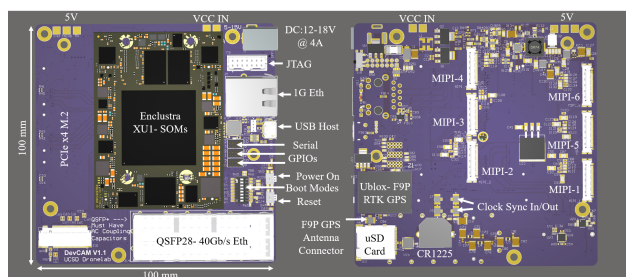


Figure 2: DevCAM FPGA V1.1 interface specifications

lines. For PCIe the lengths between each pair do not need to be matched as the interface has built in deskewing. Along with this there are 4 USB pairs in the system for general purpose USB connectivity. All in all, this results in a total of over 50 differential pairs and over 100 controlled impedance routes in the system.

The multitude of controlled impedance routes combined with the low routing surface area introduced a new issue for routing. In general, it is best practice to route differential pairs in external layers. This is primarily due to the complex factors that determine differential impedance. Routing guidelines state that even the specific weave of the fiberglass used in the PCB can significantly affect the differential impedance of routed lines. Along with this, to maintain proper impedance, a variety of rules must be followed if the lines are required to change layers at any point. This includes using ground stitching vias, aka inter layer connections, to provide a continuous current return path when changing the ground reference for the differential pair. The vias used for the differential pair can also behave as antennas for high speed protocols such as the 40GbE. This occurs when the pair does not go from the top to the bottom layer and instead goes into any internal layer. The remaining copper in the via that extends to the bottom layer acts as an antenna that can pick up noise and interfere with the transmission. To rectify this, the stub length should be minimized as much as possible and can also be backdrilled, where the PCB fab uses a larger drill size to drill out the excess copper.

Beyond all the complicated routes, there were still hundreds of single ended routes that had to be carefully designed and routed in order to minimize the number of routing layers. The size constraints also led to very tight component placements that complicated routing significantly resulting in a very densely packed PCB.

Platform Functionality Test Application

Upon receiving the fabricated and assembled boards. We began to test the capabilities to verify that that the design goals were successfully achieved. First was bring-up of basic systems and OS installation. After verifying functionality of the power systems, we began with installing Petalinux, the Xilinx Linux distribution. Following this, we set out to decide on a camera sensor and a capture pipeline to perform initial testing of image capture.

We began to integrate the Sony IMX577 Camera Module as the first camera for testing on DevCAM. This camera was selected

for multiple reasons. The IMX577 is a relatively recent, modern camera module with a high 12MP pixel array allowing for 4k image capture and provides a 12bit color output for very high data density. The IMX577 also supports multiple sensor synchronization and is configurable to provide synchronization signals that can be routed through the PL to other IMX577 sensors in other MIPI ports. A resolution of 4000x3040, 12 bits per pixel, at a frame rate of 30 frames per second (fps), yields an approximate data transfer of 1.6GB/s. With 6 cameras that is almost 10GB of data entering the FPGA per second.

For the programmable logic capture pipeline, we designed a simple capture pipeline consisting of a MIPI CSI-2 IP block leading into a Demosaic IP block and then sent to a framebuffer to be sent into the Processing System (PS) for local storage. This serves as a simple capture example that provides a means to verify if the image data is properly received. Figure 4 Shows the exact Xilinx Vivado block diagram configuration that is being used.

One major complication is designing a Linux driver for the IMX577 which requires considerable time and design effort to ensure proper device-tree setup and probing. Camera configuration is complicated due to incompatibility between the pixel bit packing used by Xilinx and existing Linux video frameworks. The Xilinx video streaming IP blocks utilize a unique 40bit format for transferring 12 bit RGB which required modification of V4L2 and other related drivers to add support for this formatting at the firmware and software level. Careful adjustment of the IMX577 pixel/clock rates combined with software to depackage the 40 bit format properly resulted in successful frame capture.

Next, the frame rate was uncapped and the pipeline was run without storage to validate the frame rate in the PL pipeline. The MIPI CSI-2 IP is able to detect corrupted and incorrect MIPI packets and will detect issues that would appear if the hardware routing was noisy or an imperfect connection. Following this successful test, we began integration of a second camera.

Integration of a second camera in the programmable logic involved making a copy of the original pipeline and modifying the I/O linkages of the copy to match the I/O pins of the second camera. Finally, the relevant synchronization pins are tied together to enable camera synchronization. Figure 3 explores how in an ideal scenario this data path can be expanded up to 6 cameras and integrate all of the available sensors/data on the platform.

Results

The evaluation of the DevCAM platform is broken down in two parts. Firstly system tests are completed and presented that validate the capabilities of the platform. Secondly, the state of performance of the system is analytically evaluated compared to other systems.

Platform Validation

Early imaging results of the DevCAM PCB system are shown in Figure 5, demonstrating the synchronous capture of two frames using the camera system. This image displays capture from two cameras pointed at a digital stopwatch. We can see from the stopwatch values that the frame capture between the two cameras is synchronized to within 1 millisecond. The development platform has been tested for general use and is under ongoing development for diverse algorithmic testing such as efficient image rectification and demosaicing. Furthermore, DevCAM camera ar-

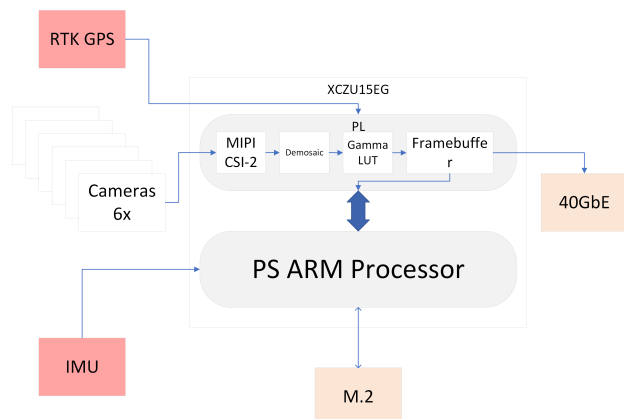


Figure 3: Example Datapath

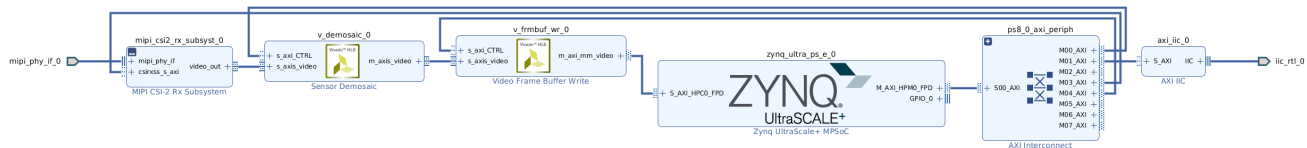


Figure 4: Single Camera Capture Pipeline Block Design

ray concepts have been built and are being expanded to include more hardware configurations.

Practical Applications

The DevCAM system architecture provides a platform to enable research into a vast array of fields. This includes embedded vision applications with single or multiple cameras, robotic agents, and hardware synchronization.

A single capture pipeline requires relatively little PL resources. As such, the remaining PL resources can be used to create a dense capture pipeline with many stages that can produce real-time qualitative results, such as monocular real-time depth extraction, feature detection, object localization, and pos-

sibly even monocular SLAM.

With multiple cameras, DevCAM has the resources to support many high bandwidth capture pipelines to act as a high density sensor data collection platform or be positioned as an edge computing device for fewer cameras or lower resolution images. In the high density case, the onboard QSFP+ 40GbE can support dense data to be passed on to other systems for dense reconstruction and high quality imaging. In the edge computing case, DevCAM can be used to perform preprocessing for many tasks, such as photogrammetry, 3D reconstruction, and others.

For robotic agents, DevCAM can act as a rich sensor suite of imaging, motion, and localization sensors and behave as a platform that simplifies data capture. Using the 1GbE port onboard, DevCAM can easily integrate into existing robotic systems and provide sensor data.

An interesting avenue for DevCAM research includes exploration of tighter timing and synchronization along with global shutter cameras and the IMU for gate level synchronized multi-image capture. Since DevCAM is a custom open-source hardware system, we can know the exact trace lengths between MIPI lanes and interface lanes for the IMU. This combined with the fully customizable programmable logic, we can design systems that can have extremely tight synchronization between external trigger signals for the cameras and pose information extracted from the IMU. This can be tuned using the anticipated electrical propagation delay in signal routing to improve synchronization. This may yield better qualitative and quantitative results for many multi-camera synchronization applications.

This concept can be extended further using the onboard RTK-GPS. The GPS timepulse signal is routed directly into the PL and can be used to create a GPSDO and also be used to possibly synchronize between multiple DevCAM modules. This can open up the possibility of multi-camera, multi-agent, gate level synchronization. Theoretically, this level of synchronization across devices can provide a very strong mathematical guarantee for many computer vision applications.

System Configurations

DevCAM has been deployed in a multitude of form-factors for the previously discussed research applications. Examples are illustrated in Figure 7. Depth estimation has at large been a field where occlusion handling and low-texture matching have been some of the key challenges. Light-field and multi-view stereo techniques have required that the scene is observed from mul-



Figure 5: Synchronous Dual Camera Capture



Figure 6: Sample epipolar array capture using 6 cameras in a linear configuration

System	Year	Horizontal Resolution (px)	Vertical Resolution (px)	Frame-rate (fps)	Mega-pixel disparities per second (MPd/s)	Power (W)	Hardware Architecture				Layout
							Rectification	Matching	3D Projection	Interfacing	
[20]	2007	512	480	200	49.15	1	FPGA	ASIC	FPGA	CPU	Stereo
[3]	2019	1080	476	30	24.49	N/A	FPGA	FPGA	FPGA	CPU	Quad
[2]	2010	640	480	103	31.64	N/A	FPGA	FPGA	N/A	CPU	Stereo
[18] D400	2018	1280	720	90	82.94	1.44	ASIC	ASIC	N/A	ASIC	Stereo
[18] D420	2018	1280	720	90	82.94	1.12	ASIC	ASIC	N/A	ASIC	Stereo
[5]	2009	750	480	25	9.00	3	FPGA	FPGA	FPGA	CPU	Stereo
[12]	2011	640	480	60	18.43	N/A	GPU	GPU	GPU	CPU	Stereo
[1]	2019	1280	960	60	73.73	5	FPGA	CPU	CPU	CPU	Epipolar
[15]	2002	659	494	2.5	0.81	N/A	CPU	CPU	CPU	CPU	Trinocular
[9]	2004	640	480	30	9.22	N/A	FPGA	FPGA	N/A	CPU	Trinocular
[16]	2007	320	240	150	11.52	N/A	FPGA	FPGA	FPGA	CPU	Stereo
[6]	2010	100	100	75	0.75	N/A	FPGA	FPGA	FPGA	CPU	Stereo
[8]	2010	450	375	7.74	1.31	5	N/A	DSP	DSP	CPU	Stereo
[10]	2009	640	480	64	19.66	N/A	FPGA	FPGA	FPGA	CPU	Stereo
[17]	2019	1280	1024	20	26.21	N/A	FPGA	GPU	GPU	CPU	Stereo
[11]	2003	256	192	50	2.46	N/A	ASIC	ASIC	ASIC	CPU	Stereo
[4]	2003	256	360	30	2.76	N/A	FPGA	FPGA	FPGA	CPU	Stereo
Ours	2020	4000	3040	30	365	30	FPGA	FPGA	FPGA	CPU	Stereo

Table 1: This table summarizes embedded vision systems in the industry, to which we can evaluate DevCAM against. It illustrates the disparity estimation rate achieved by the respective systems, the sensor layout and the hardware acceleration pairing for the respective parts of the algorithms

multiple perspectives, which these configurations address. Firstly, a quadnocular system is presented whereby the sensors are coplanar, with an equidistant baseline in horizontal and vertical. This lends itself to left-right and bottom-top consistency verification and depth estimation improvement, all while also providing occlusion understanding. Next, a linear array is presented, which consists of 6 cameras that are coplanar, and have a single 3D line that defines the co-shared optical centers. This is a configuration often used in light-field research and enables rapid epipolar line search across all images, as is demonstrated in Figure 6. The third configuration we share is a trinocular panoramic system composed of 3 DevCAM units, each with 6 cameras. This lends itself to depth estimation across a full panoramic field of view. Finally, a single DevCAM system with 6 cameras can constitute a monocular panoramic view, for which a stitched sample can be seen in Figure 8.

Comparative Evaluation

DevCAM's capabilities were evaluated based on performance for data throughput, features, and efficiency. Some notable publications and systems that are compared against include [17], [18], [3], [7], for which DevCAM demonstrably improves in the following ways:

- 1) DevCAM provides up to 2x the available PL logic resources.
- 2) DevCAM integrates key auxiliary devices (i.e. RTK-GPS, IMU, clock input/output, etc) onboard and provides direct FPGA connection for custom, synchronized data collection.

- 3) DevCAM enables 5x-40x faster data offloading via M.2 NVMe SSD storage or 40GbE Fiber link.
- 4) DevCAM supports higher resolution from more cameras than above systems and accepts nonstandard sensor configurations.

Resource	Single Camera	Dual Camera
LUTs	23077 (6.76%)	54477 (15.96%)
FF	27073 (3.97%)	64919 (9.51%)
BRAM	170 (22.85%)	269.50 (36.22%)
DSP	18 (0.51%)	36 (1.02%)

Table 2: PL Resource Utilization

Interface	Bandwidth
MIPI per lane	3200Mbps
QSFP+	40Gbps
PS-PL interface per port	40Gbps
M.2 PS PCIe 2.1	266Mbps

Table 3: Maximum Interface Bandwidth

Resource	Tri Camera	Quad Camera
LUTs	86,000 (27%)	120,000 (38%)
FF	105,000 (17%)	145,000 (25%)
BRAM	370 (50%)	480 (65%)
DSP	54 (1.53%)	72 (2.04%)

Table 4: Approximate PL Resource Utilization



Figure 7: A set of DevCAM multi-camera configurations with their respective CAD designs. From left to right: quadocular, linear light-field, trinocular panoramic and panoramic

Resource	5 Cameras	6 Cameras
LUTs	150,000 (50%)	180,000 (63%)
FF	180,000 (35%)	215,000 (45%)
BRAM	600 (80%)	700 (95%)
DSP	90 (2.55%)	108 (3.06%)

Table 5: Approximate PL Resource Utilization

Discussion and Conclusion

Designing a ground-up embedded system for high-speed, high bandwidth processing posed a multitude of challenges. Hardware design issues plagued early revisions of DevCAM. For future designs it is crucially important to have meticulous and detailed planning for I/O placement. Xilinx FPGA I/Os are grouped into banks, each bank must share the same voltage and not all I/Os are clock capable for high speed interfacing. A large portion of hardware issues in the early DevCAM revisions were voltage level related due to IO banks and clock groupings for MIPI lanes. Along with this, there is considerable effort required for proper bring-up of all electrical, programmable logic, and software systems. Given proper hardware functionality, device tree configuration presents a time-consuming task for many subsystems to ensure correct driver probing during Linux boot. Creating functional camera drivers for MIPI cameras such as the IMX577 that supported dual cameras and frame level synchronization was also very time consuming. Proper support for various image sensors can prove to be labor intensive and hinder research progress if Linux drivers and device tree stubs for the sensor is not already present or easy to find. Finally even with properly configured firmware for sensors, software integration into existing APIs such as V4L2 and GStreamer was also arduous and slow.

However, in spite of the challenges involved in hardware design and firmware/software bring-up, DevCAM clearly shows the benefit of a fully custom hardware design through its flexibility in

application, and performance throughput. We compare the DevCAM system to other state of the art systems over the previous decades, and summarize the comparison in Table ???. The notable improvement in overall performance is the disparity rate that is able to be achieved due to the increased throughput of higher-resolution image data. This is imperative when it comes to supporting new applications which require many sensors, and higher spatial resolution. Furthermore, the FPGA architecture provides unique advantages in terms of flexibility, while maintaining an increased throughput. Unfort . The platform also opens up further research in deeply synchronized frame capture with the capability for GPSDO driven clocks fed into the FPGA to be used for phase-locked clock synthesis. This will enable GPS coordinated frame capture and lock-step computation across multiple devices. This can be leveraged for extremely tight frame synchronization guarantees leading to greater accuracy in reconstruction.

As an open-source hardware, firmware, and software project, DevCAM has room for growth through further development and revisions. Currently the hardware functionality is only capable of dual camera operation. This is due to I/O planning and routing issues on the FPGA on 4 of the MIPI lanes. With a minor redesign to incorporate logic level shifting, 3 of the 4 problematic MIPI lanes can be fixed and the last one can be fixed with some simple I/O rerouting. The next major revision of DevCAM can rectify all of the outlying issues with the MIPI lanes and incorporate interface logic level conversions on all crucial interfaces such as the MIPI I2C lanes, ZED-F9P SPI lanes and etc. In a possible future revision it may be useful to consider a major redesign of the I/O planning to group all I/O pins by their voltage level and not by their proximity to the SOM connector as is currently done. This change would allow all MIPI 1.2V LVDS signals to be grouped into one I/O bank, thereby eliminating the I/O bank voltage issue.

Overall, DevCAM provides a robust framework for research



Figure 8: Sample stitched panorama capture from 6 camera system radial configuration

and experimentation on gate level synchronization, multi-camera imaging, multi-agent operations, and high density computer vision tasks.

Acknowledgments

This publication is based on work supported by the US Army Corps of Engineers under research Cooperative Agreement W912HZ-17-2-0024, NIST Award #70NANB17H211, as well as NSF award #CNS-1338192, MRI: Development of Advanced Visualization Instrumentation for the Collaborative Exploration of Big Data, under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (submission LLNL-CONF-842032), under the 2020 Qualcomm Innovation Fellowship, and by the LLNL-LDRD Program under Project No. 20-SI-005. We thank all collaborators at the Dronelab, Qualcomm Institute, the Contextual Robotics Institute, Autonomous Vehicle Laboratory, Engineers for Exploration, UC San Diego, as well as all other contributors to ideas, suggestions and comments. Opinions, findings, and conclusions from this study are those of the authors and do not necessarily reflect the opinions of the research sponsors.

References

- [1] H Harlyn Baker, Gregorij Kurillo, Allan Miller, Alessandro Temil, Tom Defanti, and Dan Sandin. Epimodules on a geodesic: Toward 360° light-field imaging. *Electronic Imaging*, 2019(3):636–1, 2019.
- [2] Christian Banz, Sebastian Hesselbarth, Holger Flatt, Holger Blume, and Peter Pirsch. Real-time stereo vision system using semi-global matching disparity estimation: Architecture and fpga-implementation. In *2010 International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation*, pages 93–101. IEEE, 2010.
- [3] Long Chen, Qin Zou, Ziyu Pan, Danyu Lai, Liwei Zhu, Zhoufan Hou, Jun Wang, and Dongpu Cao. Surrounding vehicle detection using an fpga panoramic camera and deep cnns. *IEEE Transactions on Intelligent Transportation Systems*, 21(12):5110–5122, 2020.
- [4] Ahmad Darabiha, Jonathan Rose, and JW Maclean. Video-rate stereo depth measurement on programmable hardware. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 1, pages I–I. IEEE, 2003.
- [5] Stefan K Gehrig, Felix Eberli, and Thomas Meyer. A real-time low-power stereo vision engine using semi-global matching. In *International Conference on Computer Vision Systems*, pages 134–143. Springer, 2009.
- [6] Stavros Hadjitheophanous, Christos Ttofis, Athinodoros S Georghides, and Theocharis Theocharides. Towards hardware stereoscopic 3d reconstruction: a real-time fpga computation of the disparity map. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 1743–1748. European Design and Automation Association, 2010.
- [7] Lionel Heng, Benjamin Choi, Zhaopeng Cui, Marcel Geppert, Sixing Hu, Benson Kuan, Peidong Liu, Rang Nguyen, Ye Chuan Yeo, Andreas Geiger, et al. Project autovision: Localization and 3d scene perception for an autonomous vehicle with a multi-camera system. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4695–4702. IEEE, 2019.
- [8] Martin Humenberger, Christian Zinner, Michael Weber, Wilfried Kubinger, and Markus Vincze. A fast stereo matching algorithm suitable for embedded real-time systems. *Computer Vision and Image Understanding*, 114(11):1180–1202, 2010.
- [9] Yunde Jia, Xiaoxun Zhang, Mingxiang Li, and Luping An. A miniature stereo vision machine (msvm-iii) for dense disparity mapping. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 1, pages 728–731. IEEE, 2004.
- [10] Seunghun Jin, Junguk Cho, Xuan Dai Pham, Kyoung Mu Lee, Sung-kee Park, Munsang Kim, and Jae Wook Jeon. Fpga design and implementation of a real-time stereo vision system. *IEEE transactions on circuits and systems for video technology*, 20(1):15–26, 2009.
- [11] Michael Kuhn, Stephan Moser, Oliver Isler, Frank K Gurkaynak, Andreas Burg, Norbert Felber, Hubert Kaeslin, and Wolfgang Fichtner. Efficient asic implementation of a real-time depth mapping stereo vision system. In *2003 46th Midwest Symposium on Circuits and Systems*, volume 3, pages 1478–1481. IEEE, 2003.
- [12] Sang Hwa Lee and Siddharth Sharma. Real-time disparity estimation algorithm for stereo camera systems. *IEEE transactions on Consumer electronics*, 57(3):1018–1026, 2011.
- [13] Peidong Liu, Marcel Geppert, Lionel Heng, Torsten Sattler, Andreas Geiger, and Marc Pollefeys. Towards robust visual odometry with a multi-camera system. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1154–1161, 2018.
- [14] DE Meyer, H Wang, D Sandin, C McFarland, E Lo, G Dawe, J Dai, T Nguyen, H Baker, MD Brown, et al. Starcam-a 16k stereo panoramic video camera with a novel parallel interleaved arrangement of sensors. *Electronic Imaging*, 2019(3):646–1, 2019.
- [15] Jane Mulligan, Volkan Isler, and Kostas Daniilidis. Trinocular stereo: A real-time algorithm and its evaluation. *International Journal of Computer Vision*, 47(1-3):51–61, 2002.
- [16] Chris Murphy, Daniel Lindquist, Ann Marie Rynning, Thomas Cecil, Sarah Leavitt, and Mark L Chang. Low-cost stereo vision on an fpga. In *15th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM 2007)*, pages 333–334. IEEE, 2007.
- [17] Morgan Quigley, Kartik Mohta, Shreyas S Shivakumar, Michael Watterson, Yash Mulgaonkar, Mikael Arguedas, Ke Sun, Sikang

- Liu, Bernd Pfrommer, Vijay Kumar, et al. The open vision computer: An integrated sensing and compute system for mobile robots. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1834–1840. IEEE, 2019.
- [18] Intel Realsense. D400 and d420 from the intel realsense depth and tracking camera series.
- [19] Zishen Wan, Yuyang Zhang, Arijit Raychowdhury, Bo Yu, Yanjun Zhang, and Shaoshan Liu. An energy-efficient quad-camera visual system for autonomous machines on fpga platform. *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 1–4, 2021.
- [20] John Iselin Woodfill, Ron Buck, Dave Jurasek, Gaile Gordon, and Terrance Brown. 3d vision: Developing an embedded stereo-vision system. *Computer*, 40(5):106–108, 2007.

Author Biography

Akhil M. Birlangi is a PhD candidate at the University of California San Diego in the department of Computer Science and Engineering, where he previously received his M.S and B.S. in Electrical Engineering. He is graduate student researcher at the UC San Diego Dronelab working on embedded system design of multi-camera systems and multi-agent deployable systems. His development experience spans the full stack development from electrical PCB design through RTL, firmware, and embedded software.

Dominique E. Meyer holds executive positions as CEO at Looq AI (an infrastructure construction diagnostics platform), CTO of CamerEye (Swimming pool safety through computer vision), and acts as industry consultant across startups and companies. He received his Ph.D. in Computer Science and Engineering in 2021, from the University of California San Diego, where he previously also received his B.S. in Physics in 2017 and his M.S. in 2019.

Falko Kuester is a Professor in the Jacobs School of Engineering at UC San Diego and the director of the UC San Diego DroneLab, advancing research in robotics, remote imaging, large-scale visual analytics and virtual reality. He received his Ph.D. in Computer Science from UC Davis in 2001 and a M.S.E. in Mechanical Engineering as well as Computer Science and Engineering from the University of Michigan, Ann Arbor, in 1994 and 1995 respectively.