# ZHENYU YE

Email: zhenyu.z.ye@gmail.com ◇ Homepage: http://zhenyu-ye.net/
Location: Amsterdam, The Netherlands ◇ Phone: upon request.

## SUMMARY

◇ Commercial product development from prototyping to mass production.
◇ Optimization of computer vision and machine learning algorithms on custom hardware.
◇ Internet-of-things devices, edge and cloud platforms for machine learning applications.
◇ Implementation of electronics systems, from chip design to firmware/software stack.
◇ Integration of electronics systems into mechatronics and robotics systems.

## PROFESSIONAL EXPERIENCE

**Computer Vision Engineer, AM-Flow, Full Time** *2024 January - Current*
◇ Automate 3D printing factory with computer vision.

**Machine Vision Consultant, ams OSRAM (via TMC), Contract** *2020 July - 2023 Dec*
◇ Design algorithms (detect object by color/shape/etc.) and demonstrators for smart image sensor.
◇ Design hardware-friendly high-dynamic range algorithms for future image sensors.
◇ Analyze AR/VR applications and machine learning models on resource-constrained systems.
◇ Perform modelling and optimization of camera systems for various applications.
◇ Develop Linux kernel space drivers for image sensors and bring up sensor features.
◇ Bring up a new variant of intelligent LED chip and develop firmware for new features.

**Embedded Software Engineer, Bosch (via TMC), Contract** *2023 April - 2023 August*
◇ Integrate Linux kernel space driver for multimedia device in conference system.

**Machine Vision Expert, TMC, Full Time** *2020 July - 2023 Dec*
◇ Help clients create innovative machine vision systems.

**Hardware and Software Engineer, CASPAR.AI, Full Time** *2019 September - 2020 May*
◇ Bring up smart home device prototypes.
◇ Build edge computing platforms supporting deep learning applications in vision domain.

**Hardware and Devices Engineer, Connecterra, Full Time** *2016 July - 2019 August*
◇ Commercial product development, from prototyping to mass production.
◇ Electronics, firmware, and software for wireless Internet-of-things devices at industrial scale.
◇ Optimization of neural networks for motion analysis on battery-powered devices.
◇ Edge computing platform supporting deep learning applications.

**Firmware Engineer, Intel, Full Time** *2014 August - 2016 June*
◇ Implementation of firmware stack for Intel Image Signal Processors (ISP).

**Researcher, Embedded vision, TU Delft, Full Time** *2014 April - July*
Implementation of a proprietary computer vision algorithm on FPGAs in 4 months.

**PhD researcher, Embedded vision architecture, TU Eindhoven, Full Time** *2009 - 2014*
◇ Design of vision algorithms and electronic systems for 1000 frames-per-second vision processing.
◇ Implementation of vision based closed-loop precision motion control.
◇ Low-power image processor architecture (Xetal-Pro).
◇ Teaching GPU architecture and programming, and assisting hands-on labs of optimizing vision algorithms on GPUs: histogram equalization (2009), stereo vision (2010), natural feature detection (2011), face detection (2012), and neural networks for object recognition (2013).

**MSc project, GPU architecture & programming, TU Eindhoven, Full Time** *2008 - 2009*
◇ Modeling and simulating GPU architecture. Optimize linear algebra algorithms on GPUs.

## TECHNICAL SKILLS

◇ **Computer Vision**: design of hardware-friendly algorithms from scratch and from existing libraries (Matlab/OpenCV/etc.), image processing on CPU/DSP/GPUs/FPGAs and custom chip, full-system implementation of high-speed (1000 fps) imaging platform.
◇ **Machine Learning**: neural network in vision and motion domain using commercial tools such as TensorFlow(/Lite/uC) as well as custom codes, optimization of neural network on resource-constrained systems such as microcontrollers and micro-NPUs.
◇ **Internet-of-things**: battery-power wireless sensors, edge-computing platform supporting deep learning applications, firmware and software stack from device to cloud (Azure/AWS/Google).
◇ **Embedded software**: C/C++ targeting various embedded processors (microcontrollers/DSPs), single-instruction-multiple-data (SIMD) extensions of Intel/ARM/etc., programming GPUs with languages like CUDA/OpenCL/etc., real-time image processing with Python, multi-threading on multi-core processors with pthread/OpenMP/etc., real-time operating systems, scripting (Bash/etc.), Linux kernel space driver.
◇ **Electronic systems**: parallel architecture, register-transfer level (RTL) design using hardware description languages (VHDL/Verilog), high-level-synthesis (HLS) tools, FPGA toolchains, high-speed interfaces for image sensors and cameras (LVDS/CameraLink/etc.).
◇ **Mechatronics systems**: digital and analog interface (ADC/DAC), sensor and actuator interface, design of feedforward and feedback controllers, modeling and simulation (e.g., in Matlab/Simulink), performance tuning for measurement noise and delay of visual feedback.

## EDUCATION

**PhD (dr.) in Mechanical Engineering, TU Eindhoven, The Netherlands** *2009 - 2020*
Worked on the Embedded Vision Architecture project. My thesis, titled "Implementation, Modeling, and Exploration of Precision Visual Servo Systems", was successfully defended on 26 May, 2020.

**MSc. (ir.) in Embedded Systems, TU Eindhoven, The Netherlands** *2006 - 2009*
Funded by TU/e scholarship, with MSc. thesis on GPU architecture and programming.

**BSc., Electronic Engineering, Harbin Institute of Technology, China** *2002 - 2006*
BSc. project on the implementation and optimization of digital filters on DSPs.

## ADDITIONAL INFORMATION

**Language skills**: English, Chinese, Dutch (A2-level, genoeg voor koetjes-en-kalfjes).
**Practical matters**: Dutch permanent residence and work permit, Dutch driving license.

## PROJECT PORTFOLIO

**Project 1: Embedded Vision Architecture at TU Eindhoven**
**Summary: build prototype high-speed and high-precision machine vision system.**
◇ Design of vision algorithms and electronic systems for 1000 frames-per-second vision processing on FPGA.
◇ Implementation of vision based closed-loop control systems, a.k.a. "vision-in-the-loop" systems, for precision motion control (see the material below).
◇ Cooperation with multiple industrial partners and a multidisciplinary team consisting of electronic engineers, computer vision scientists, and control engineers.

**Used technologies**: C, assembly, C++, Python, Matlab, CUDA, OpenCL, OpenMP, SIMD, FPGA, RTL design, VHDL/Verilog, High-Level Synthesis, ADC/DAC interface, LVDS interface, CameraLink, high-speed (1000 fps) image processing, image processing algorithm, OpenCV, mechatronics, motion control, feedforward and feedback controller design and tuning.

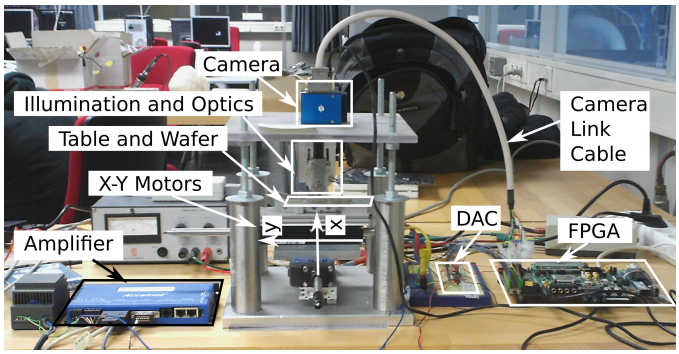


Figure 1: 1000 fps vision-in-the-loop system.

**Project 2: Real-time 2D-to-3D conversion on FPGAs at TU Delft**
**Summary: implement a proprietary single-image 2D-to-3D conversion algorithm on FPGA.**
◇ Simplify the 2D-to-3D conversion algorithms designed by image scientist such that it can be efficiently implemented on FPGAs.
◇ Rapid prototyping of the vision algorithm within 4 months for demonstration of feasibility.

**Used technologies**: RTL design, VHDL/Verilog, FPGA, C, image processing.

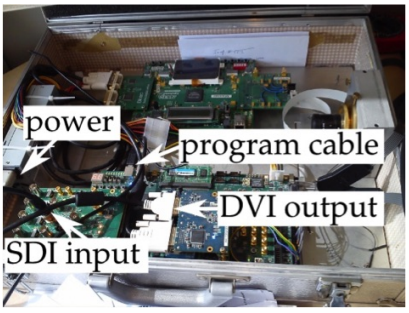


Figure 2: FPGA system running 1080p 2D-to-3D conversion at 30fps.

**Project 3: Image Signal Processor at Intel**
**Summary: image processing on custom circuits and vector processors.**

◇ Implementation of firmware stack for customized circuits and image signal processors.
◇ Support Windows and Android driver teams for bringing up and troubleshooting device features
◇ Cooperate with image algorithm designers for implementation and interface of firmware.

**Used technologies**: C, microcontrollers, DSP, SIMD.

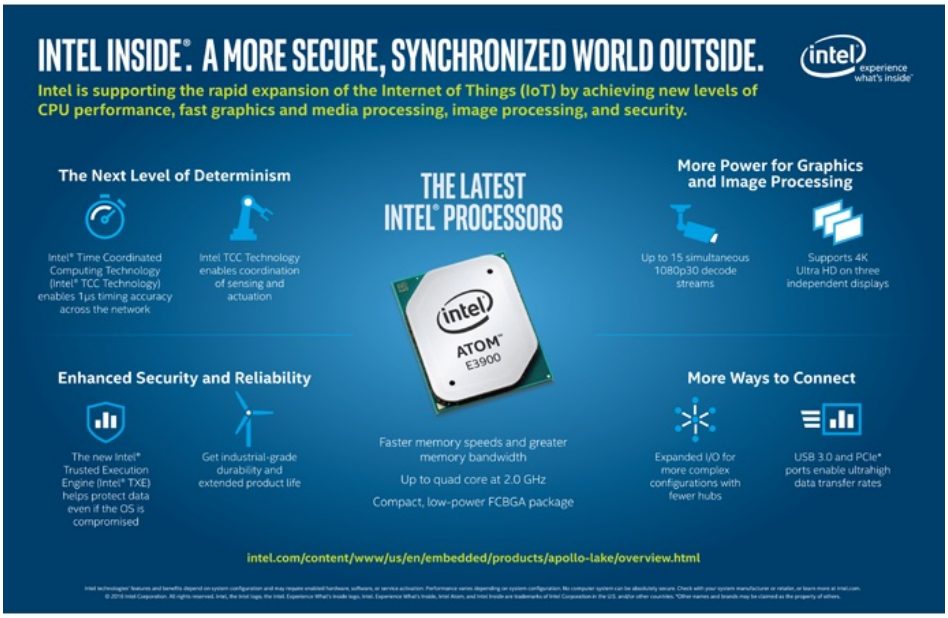


Figure 3: Image signal processor of Intel (top right part).

**Project 4: Deep Learning on IOT sensors and edge computers at Connecterra**
**Summary: perform motion analysis of animals on IOT devices and edge computers.**
◇ Commercial product development, from prototyping to mass production.
◇ Electronics, firmware, and software for wireless Internet-of-things devices at industrial scale.
◇ Optimisation of machine learning algorithms on battery-powered devices.
◇ Building the technology behind IDA: Intelligent Dairy farmer's Assistant.

**Used technologies**: C, C++, Python, microcontroller (Arm Cortex-M), IOT wireless protocols, RF hardware, RFID, edge computing, deep learning, TensorFlow, cloud platform (Azure/Google).

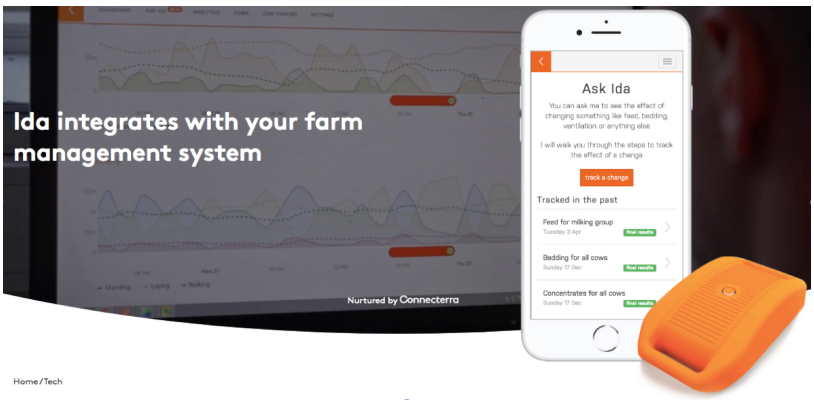


Figure 4: IDA sensor. More info on `http://ida.io`

**Project 5: Intelligent Smart Building at CASPAR.AI**
**Summary: develop smart home devices and on-premises edge computing infrastructure.**
◇ Prototyping and production of smart sensor devices for smart home.
◇ Build infrastructure to support machine learning on edge computers.

**Used technologies**: C, Python, OpenCV, deep learning, on-premises edge computing, cloud platform (AWS).

Figure 5: Caspar smart home device.

**Project 6: New Image Sensor and Compute Platform at ams OSRAM**
**Summary: prototyping of next generation image sensor and processing platform.**
◇ Design classic computer vision algorithms (detecting objects by color/shape/motion etc.) on resource-constrained embedded vision platform.
◇ Design hardware friendly High Dynamic Range (HDR) algorithms.
◇ Benchmark and optimize deep learning models (object classification, human/face detection, etc.) on resource-constraint platforms.
◇ Evaluate various off-the-shelf and custom-built hardware accelerators for computer vision and deep learning applications.
◇ Develop Linux kernel space drivers for image sensors and bring up sensor features.

**Used technologies**: C, C++, Python, Python GUI (Qt), bare-metal, RTOS (Mbed), Linux, docker, Linux kernel space driver, microcontrollers (Arm Contrex-M), SIMD, RISC-V, microNPU (Arm U55), TensorFlow (/Lite/microController), Caffe, Arm CMSIS NN, deep learning, neural network quantization, OpenCV, object/shape detection, HDR.

Figure 6: ams OSRAM image sensor connected to an evaluation kit.

**Project 7: End-to-end Modeling of Vision Systems for AR/VR Use Cases at ams OSRAM**

**Summary: model illuminator, optics, image sensors, and scene for AR/VR applications.**
◇ Integrate in-house models (in Matlab and Python) into a framework for camera system simulation.
◇ Implement tools for design parameter optimization of camera system.
◇ Co-optimize vision algorithms and deep learning models with camera system.

**Used technologies**: C, C++, Python, Matlab, bare-metal, microNPU (Arm U55), TensorFlow (/Lite/microController), deep learning, neural network quantization, OpenCV, AR/VR applications.

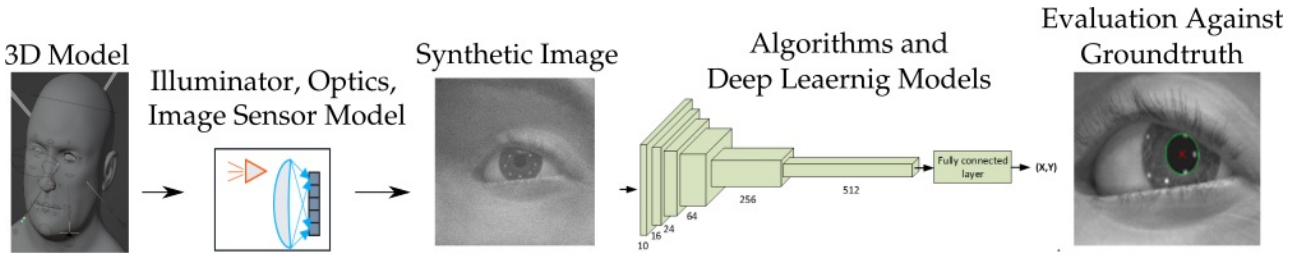


Figure 7: End-to-end modeling framework for AR/VR applications.

**Project 8: New Device Bring Up and Driver Integration at Bosch**
**Summary: bring up a new variation of multimedia device in conference system.**
◇ Customize Linux kernel driver for new chips in device.
◇ Bring up and troubleshoot new device prototypes.

**Used technologies**: C, Linux kernel space driver, I2C, oscilloscope, new device bring up.

Figure 8: Current generation of Bosch DICENTIS multimedia device in conference system.

**Project 9: New Chip Bring Up and Firmware Development at ams OSRAM**
**Summary: bring up a new variant of intelligent LED for automotive ambient lighting.**
◇ Bring up initial samples of new chip, and develop firmware for new features.

**Used technologies**: C, Arm Cortex-M, bare-metal, SPI, I2C, oscilloscope, logic analyzer.


Figure 9: Current generation of intelligent LED OSIRE E3731i for automotive ambient lighting.