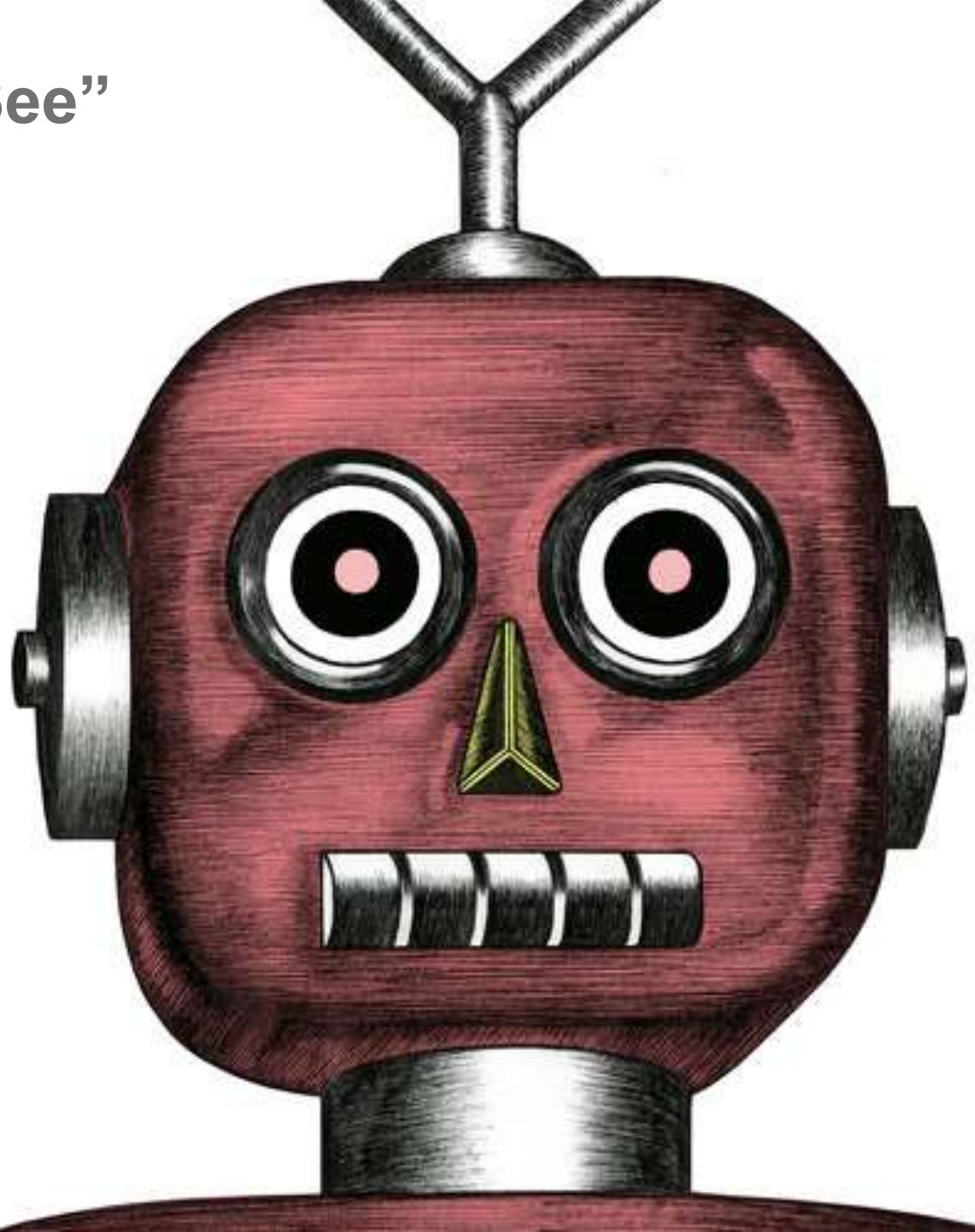
TEACHING THINGS TO SEE - INTRODUCTION TO I.MX 8 VISION ARCHITECTURE

AMF-DES-T1939

GABRIEL DAGANI
SENIOR ENGINEER, GRAPHICS & ARCHITECTURE
OCT 5TH, 2016

SECURE CONNECTIONS FOR A SMARTER WORLD

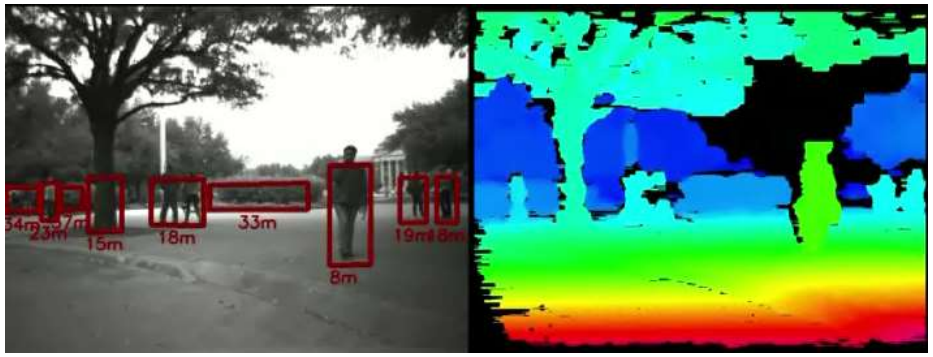
Teaching Machines to “See”

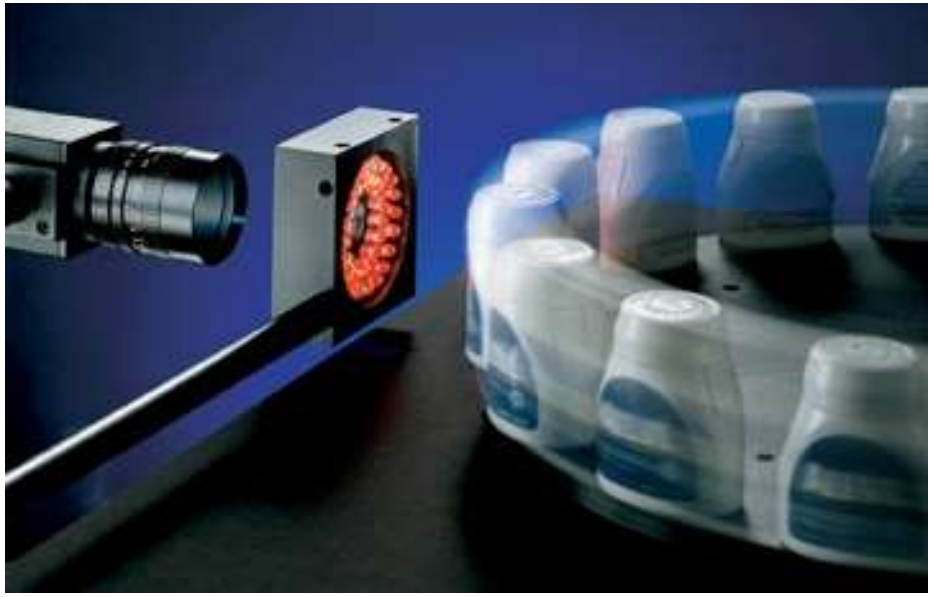


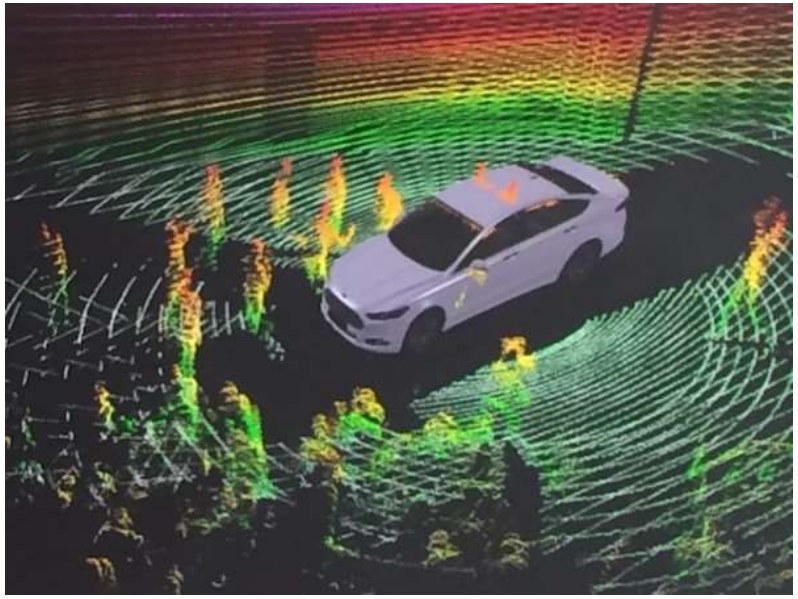
AGENDA

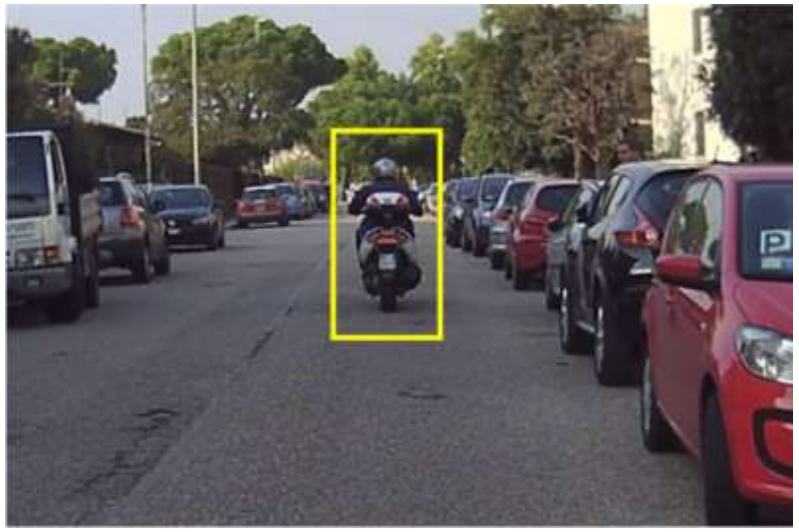
- Intro to Machine Vision:
  Automotive and Consumer Applications
- Enabling Machine Vision in an SoC
- Machine Vision on i.MX 8

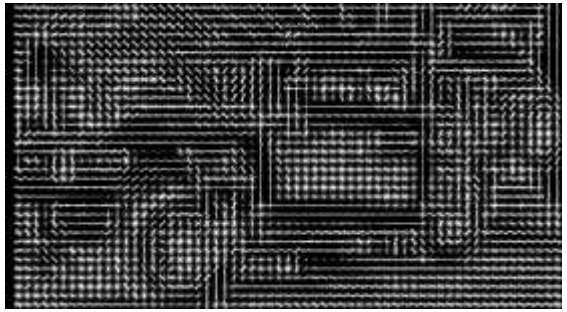
What Comes to Mind When You Think of Machine Vision?











Computer Vision Hierarchy

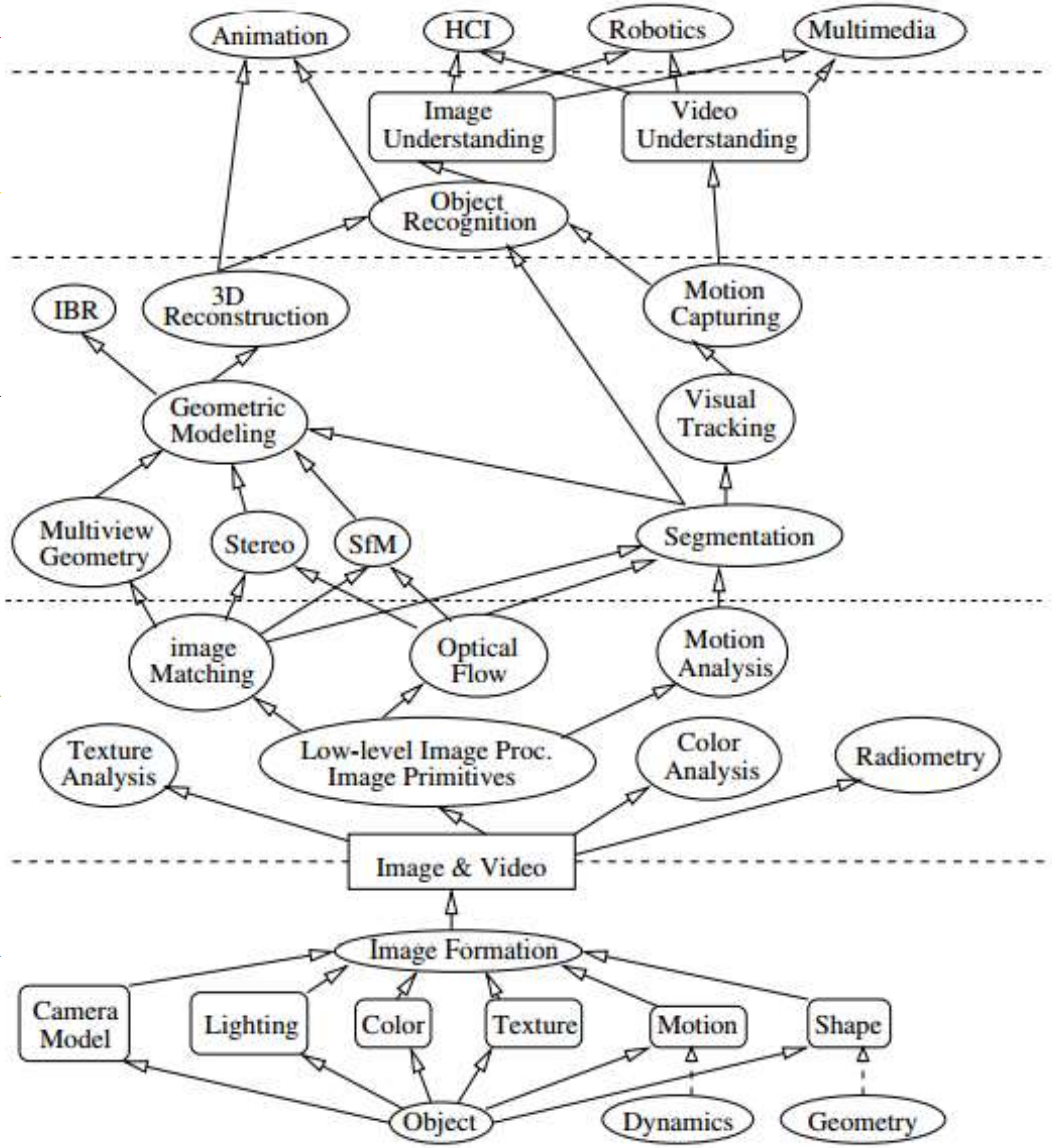
High Level (Complex) Vision →

Mid Level (Multi-layer) Vision →

Low Level (Single-layer) Vision →

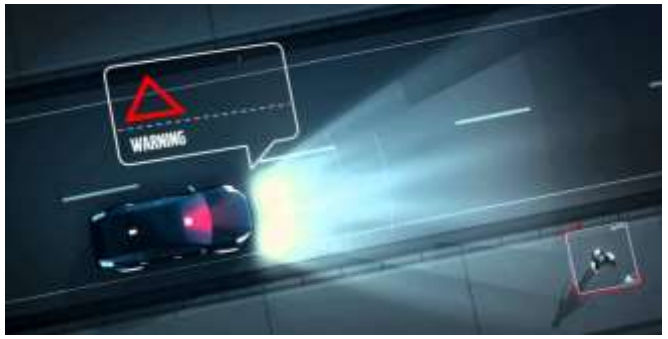
Image Processing →

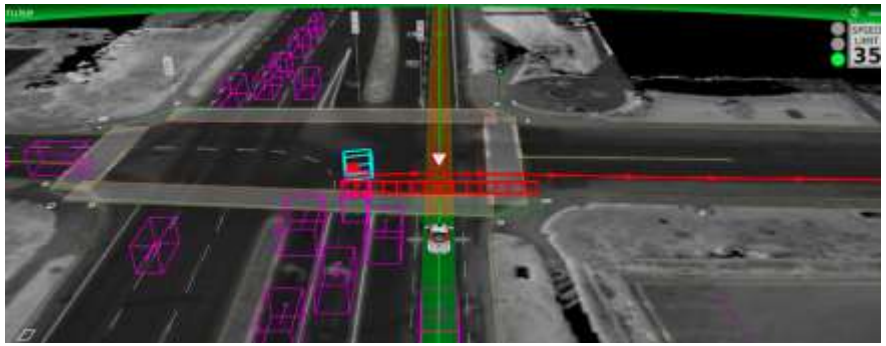
Image Formation / Collection →



What Are the Vision Use-Cases Involved in Automotive?

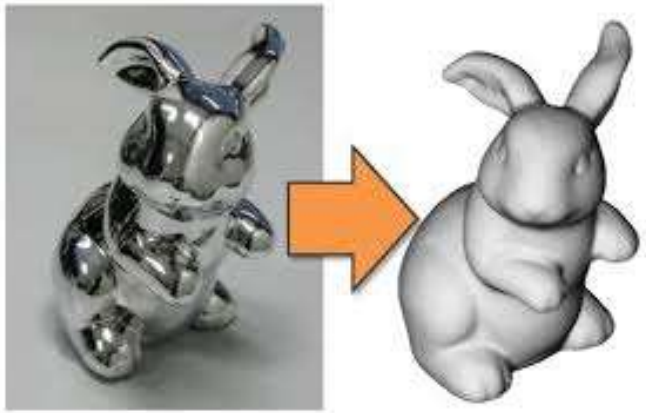
- Gesture control interaction
- Augmented reality
- Traffic sign / license plate detection
- Automobile safety
- Autonomous driving

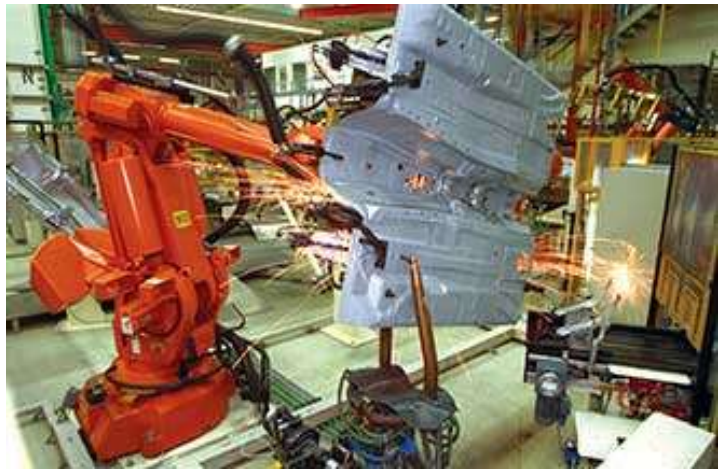



What Are the Vision Use-Cases for Industrial / Commercial?

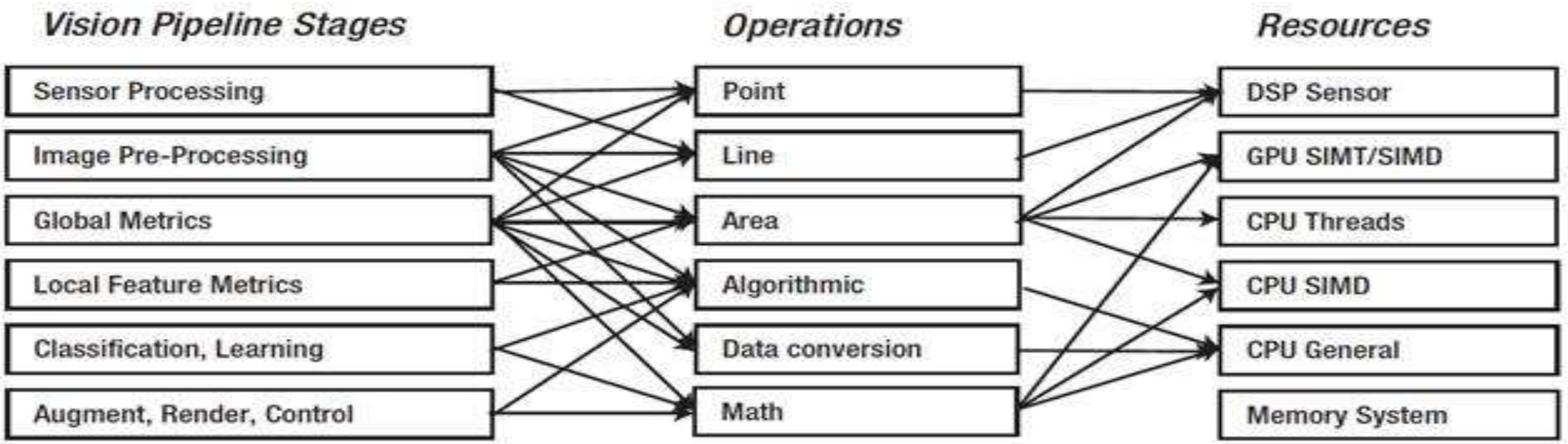
- 3D Survey
- Single / Multi Camera Surveillance
- Manufacturing defect recognition
- Robotics
- Drones
- Sensor Fusion





ENABLING MACHINE VISION IN AN SOC

Acceleration Strategies for Vision

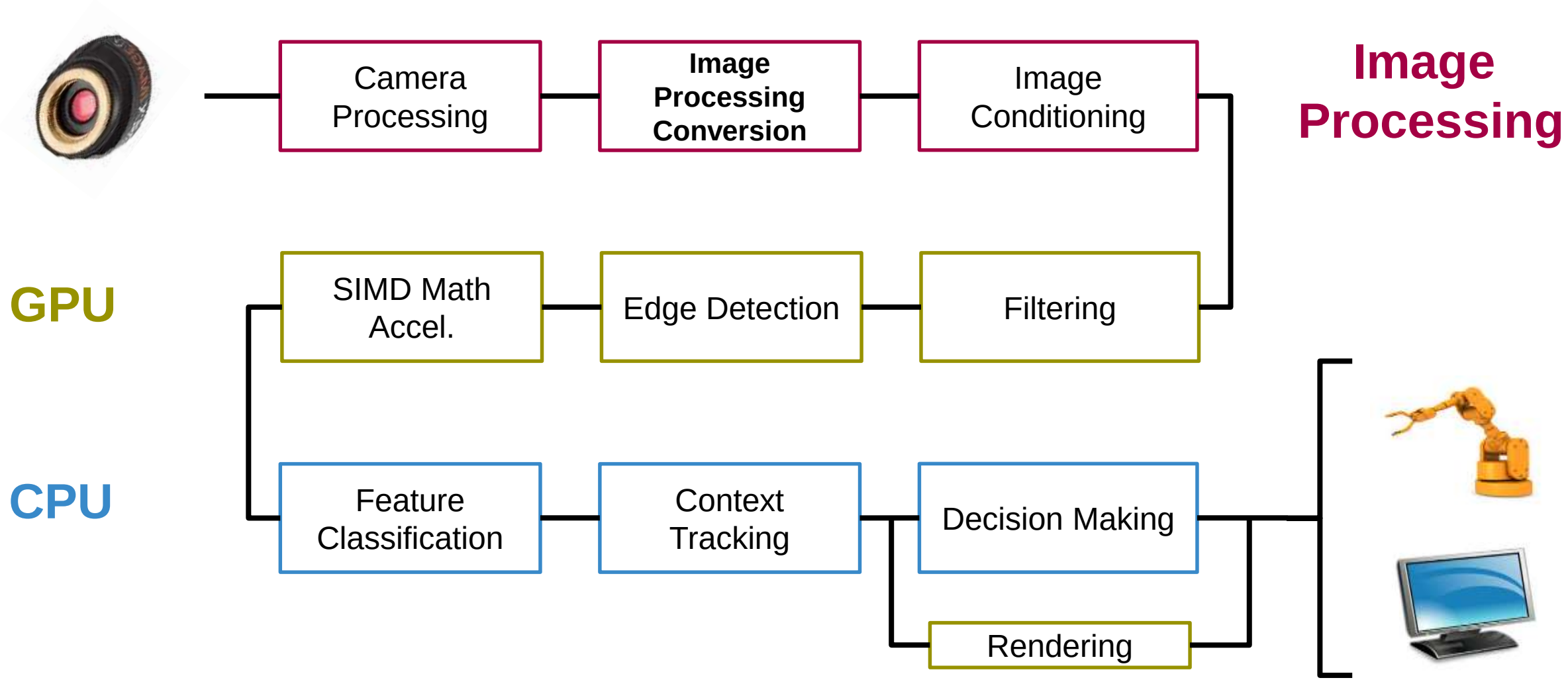


[1] Krig, Scott, Computer Vision Metrics: Survey, Taxonomy and Analysis, Springer Apress, 2014.

Acceleration Strategies for Embedded

Resource	Description	Pros	Cons	Notes
Fixed Function HW	Specialty Logic that performs a single set of vision functions	Small / Fast	Not flexible. Algorithm stuck in time.	Vision algorithms evolve very quickly
Proprietary Vision Engine	Group of DSP devices that perform 2D calculates quickly	Scalable	Algorithms are not portable, Architecture is usually also proprietary	Initial algorithm optimization commitment forces long term investment in a single HW Vendor
GPU	Leverage the ~10x compute power of a GPU for general compute	High Performance Availability Code portability	Power Consumption Not efficient for Sparse compute	Usually requires a CPU to perform control and classification tasks
CPU	Quickly leverage the CPU to perform the vision algorithm from top to bottom.	Code portability Time to market Well understood	Lower compute throughput Shared with OS and other apps	Performance bound and not ideal for dense compute
CPU SIMD	CPU SIMD instruction set utilized for the higher compute performance	Available Speed increase in performance	SIMD setup overhead is high Performance is still bound by CPU being shared by OS and other apps	Useful, but reduces algorithm flexibility because SIMD instruction intrinsics are not well supported in general compilers

Vision Pipeline Example



Vision Pipeline Example (revisited)

Camera Processing

Exposure, Color Balance, Dead Pixel Corr,

Image Processing Conversion

De-mosaic Bayer
RGB -> YUV

Image Conditioning

Gamma Corr.
HDR, Denoise
Dewarping

Image Processing

GPU

SIMD Math Accel.

HOG, Optical Flow, FAST, SURF, BRIEF, SIFT

Edge Detection

Canny Edge, Harris Corner, Kirsh

Filtering

Sobel, FIR, smoothing, sharpening, Median, Histogram, Integral

CPU

Feature Classification

SVM, Neural Nets, Boosted Decision Trees

Context Tracking

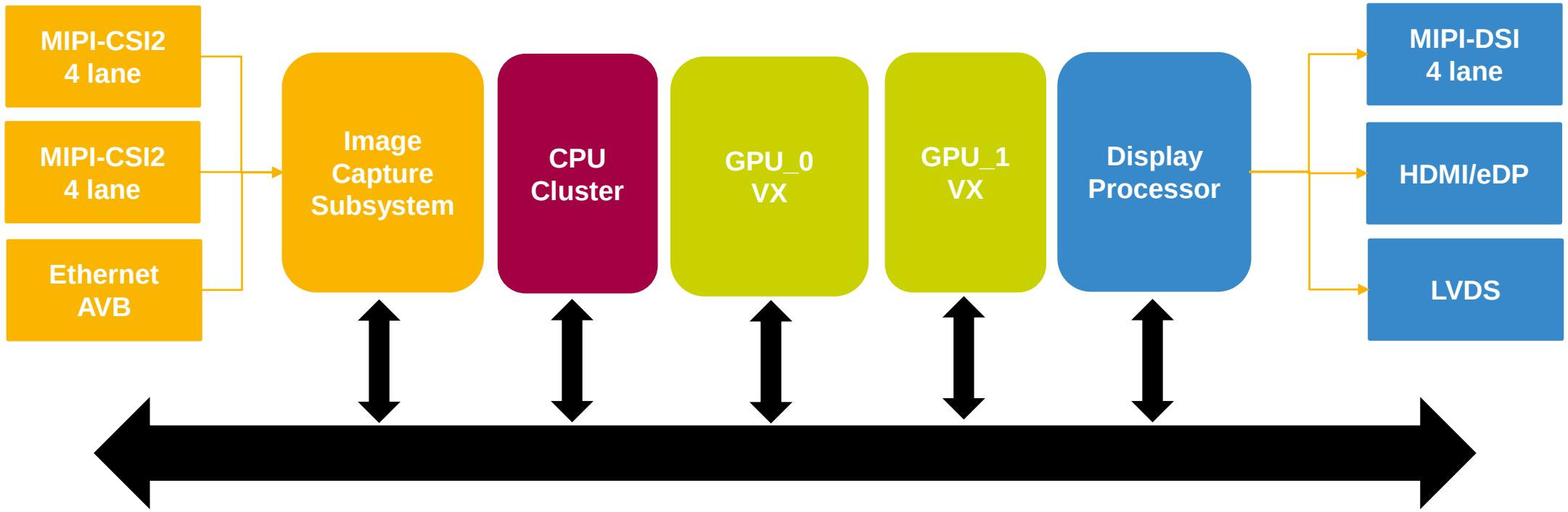
High level contour tracking, object occlusion strategies

Decision Making

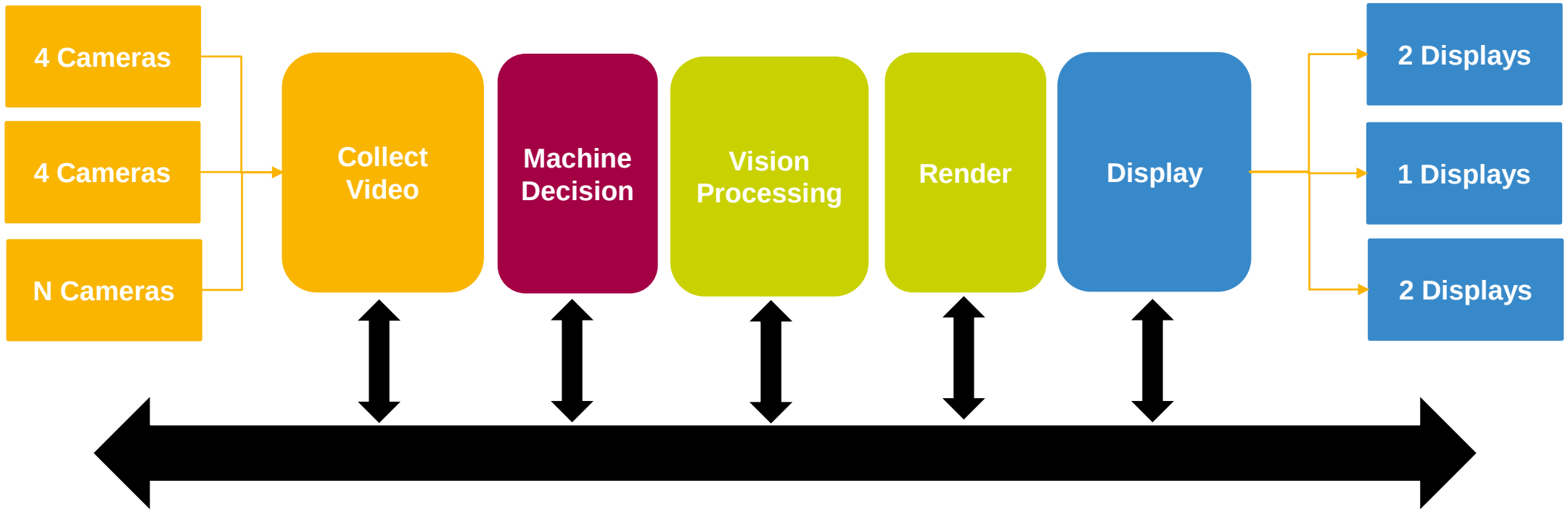
Rendering

Indicators of Detection, Augmented Reality

i.MX 8QuadMax Pixel Pipe



i.MX 8QuadMax Pixel Pipe



MACHINE VISION ON i.MX 8

Open Source APIS that Accelerate Machine Vision and Compute

GPU Compute Shaders (OpenGL 4.X and OpenGL ES 3.1)

Pervasively available on almost any mobile device or OS
Easy integration into graphics apps - no vision/compute API interop needed
Program in GLSL not C
Limited to acceleration on a single GPU

General Purpose Heterogeneous Programming Framework

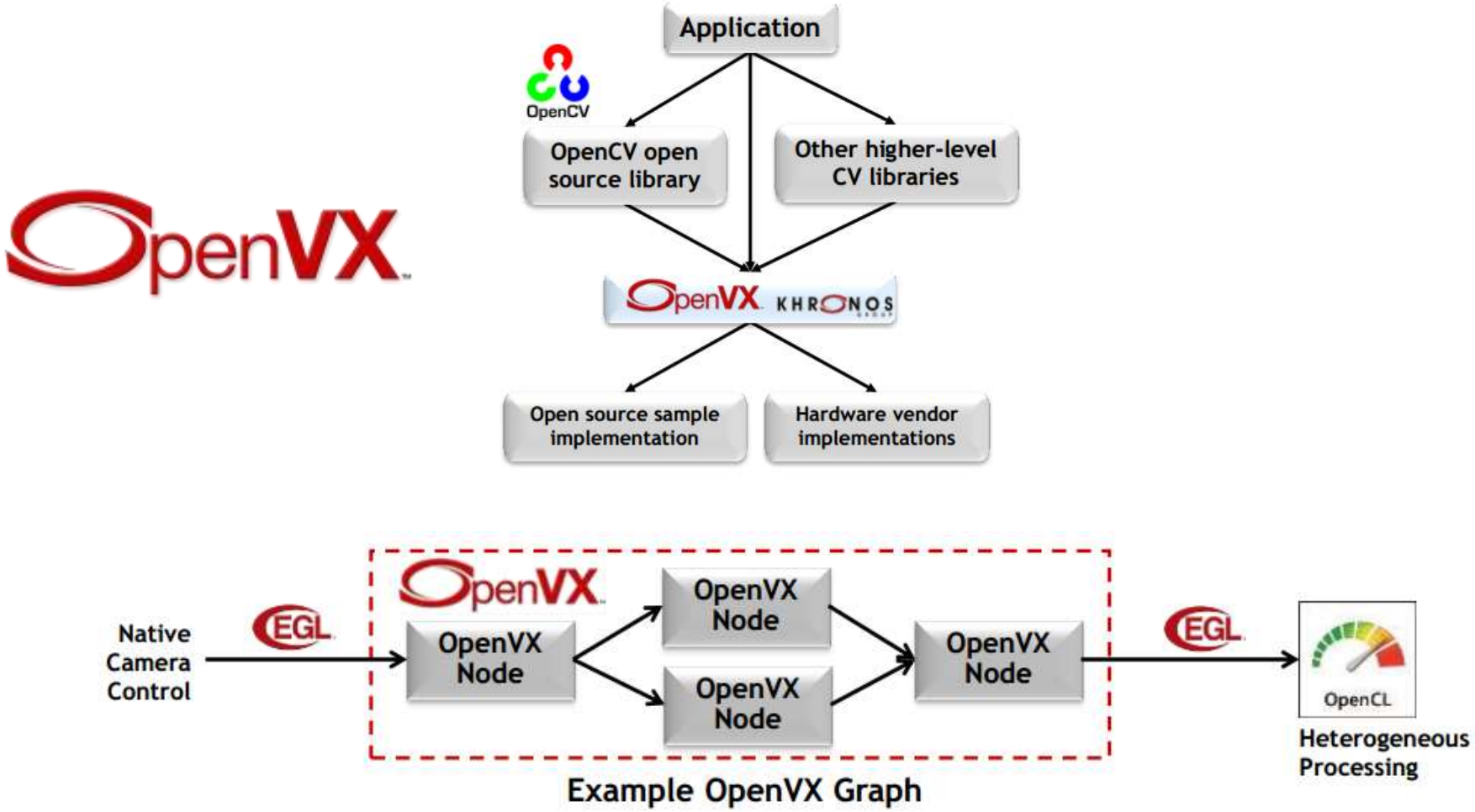
Flexible, low-level access to any devices with OpenCL compiler
Single programming and run-time framework for CPUs, GPUs, DSPs, hardware
Open standard for any device or OS - being used as backed by many languages and frameworks
Needs full compiler stack and IEEE precision

Out of the Box Vision Framework - Operators and graph framework library

Can run some or all modes on dedicated hardware - no compiler needed
Higher-level abstraction means easier performance portability to diverse hardware
Graph optimization opens up possibility of low-power, always-on vision acceleration
Fixed set of operators - but can be extended

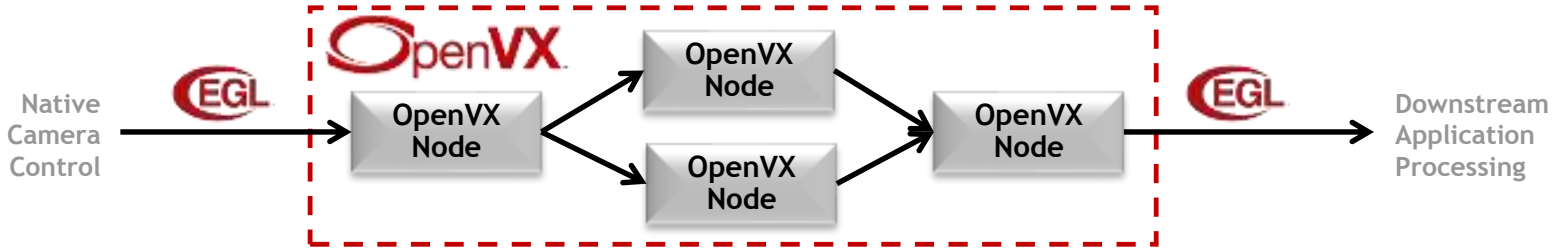
It is possible to use OpenCL or GLSL to build OpenVX Nodes on programmable devices!

OpenVX Standard Framework



OpenVX Programming Framework

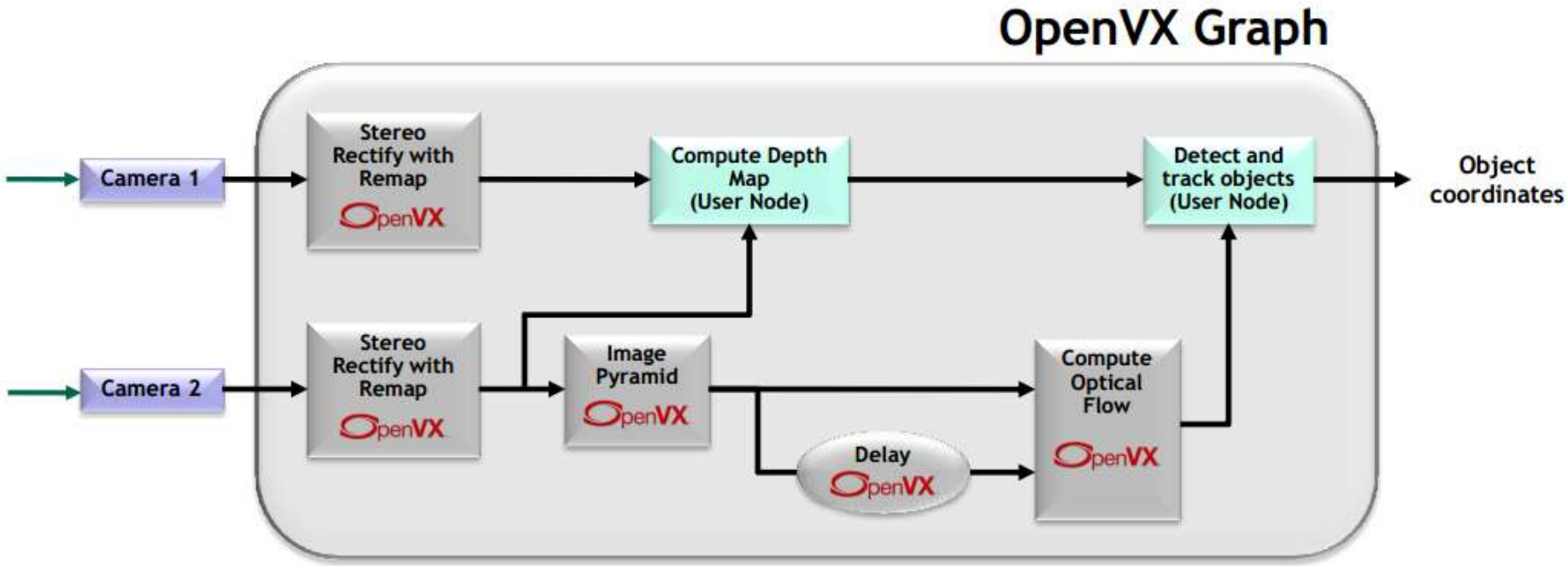
- Directed Acyclic Graph (DAG) Framework Pipeline
  - Optimized precompiled kernels of commonly used vision processes
    - A subset of OpenCV that lends itself to HW Acceleration
      - HW Vendor can create hardened / silicon aware specialized kernels
      - App Developer can create unique shader-based kernels using OpenCL or OpenGL APIs
  - OpenVX Graphs can split, join, delay, and produce callbacks depending on heuristics.
    - OpenVX Primitives include: Images, Image Pyramids, Process Graphs, Kernels, Control Parameters



Example OpenVX Graph

Accelerating Vision Using OpenVX

Example Graph - Stereo Machine Vision

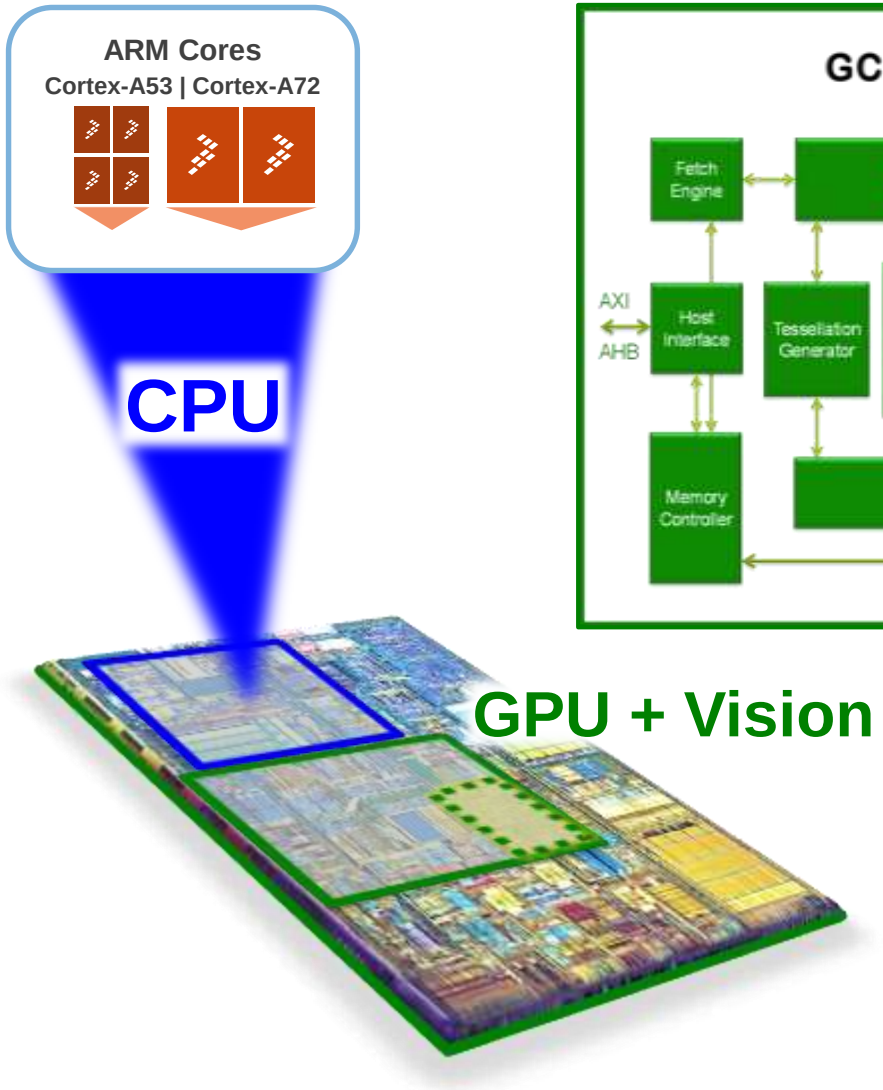


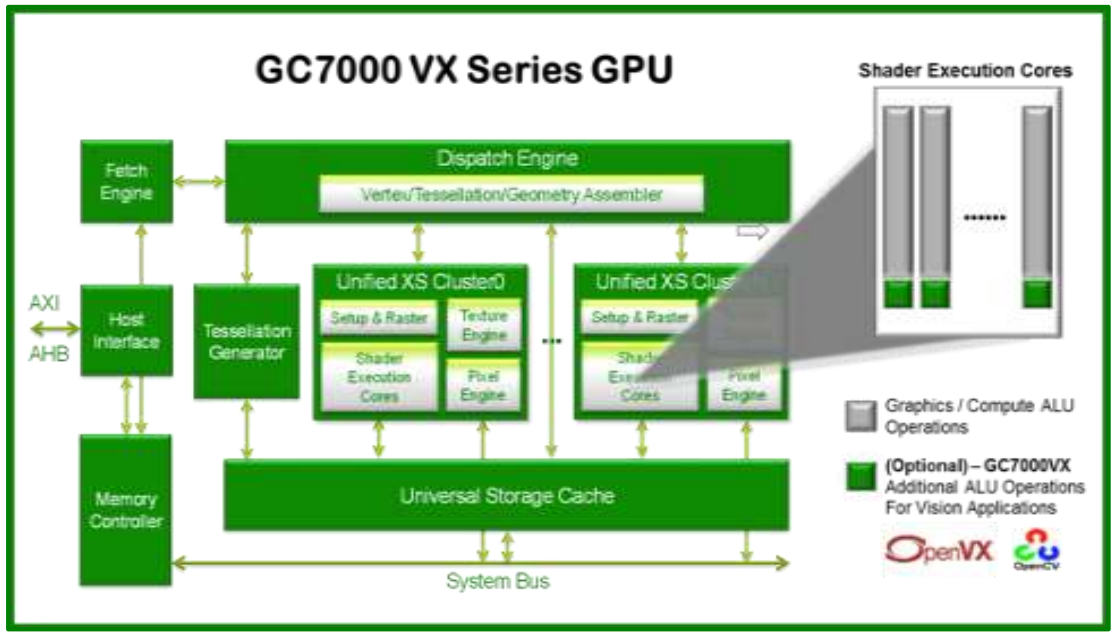
Tiling extension enables user nodes (extensions) to also optimally run in local memory

OpenCV and OpenVX Are Complimentary

Governance	Community driven open source with no formal specification	Formal specification defined and implemented by hardware vendors
Conformance	No conformance tests for consistency and every vendor implements different subset	Full conformance test suite / process creates a reliable acceleration platform
Portability	APIs can vary depending on processor	Hardware abstracted for portability
Scope	Very wide 1000s of imaging and vision functions Multiple camera APIs/interfaces	Tight focus on hardware accelerated functions for mobile vision Use external camera API
Efficiency	Memory-based architecture Each operation reads and writes memory	Graph-based execution Optimizable computation, data transfer
Use Case	Rapid experimentation	Production development & deployment

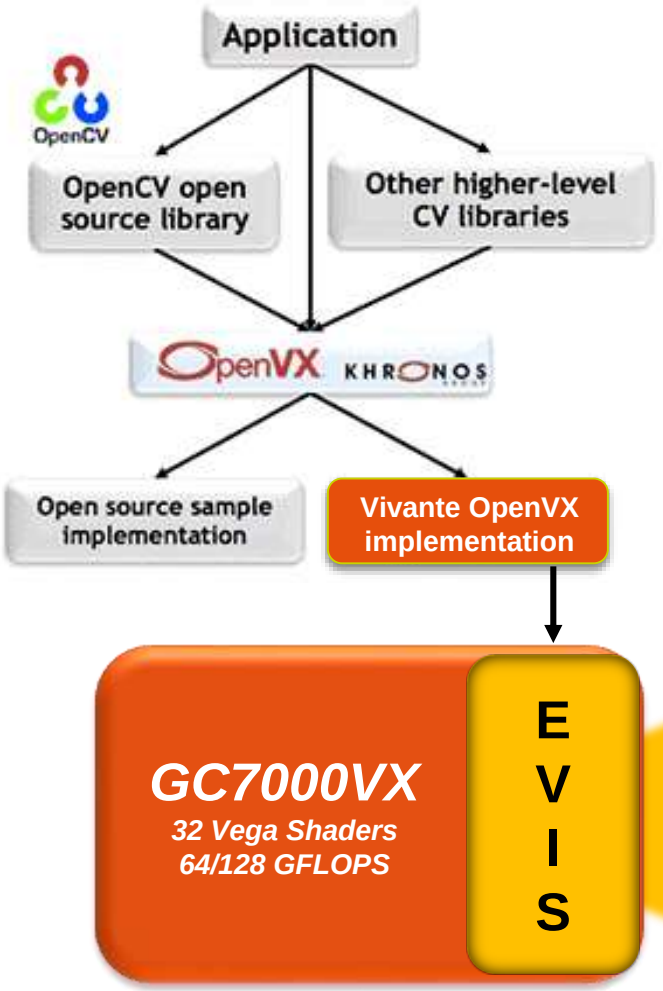
Vision Acceleration on i.MX 8





* Vision Optimized Dynamic VLIW Architecture
* Interleaved Multi-Threading Unified Cache
* Stream Interconnect to Hardwired Vision Functions
* Intelligent Switching Power Management
* Extended Vision Instruction Set (EVIS)

i.MX Graphics with Vision Extensions

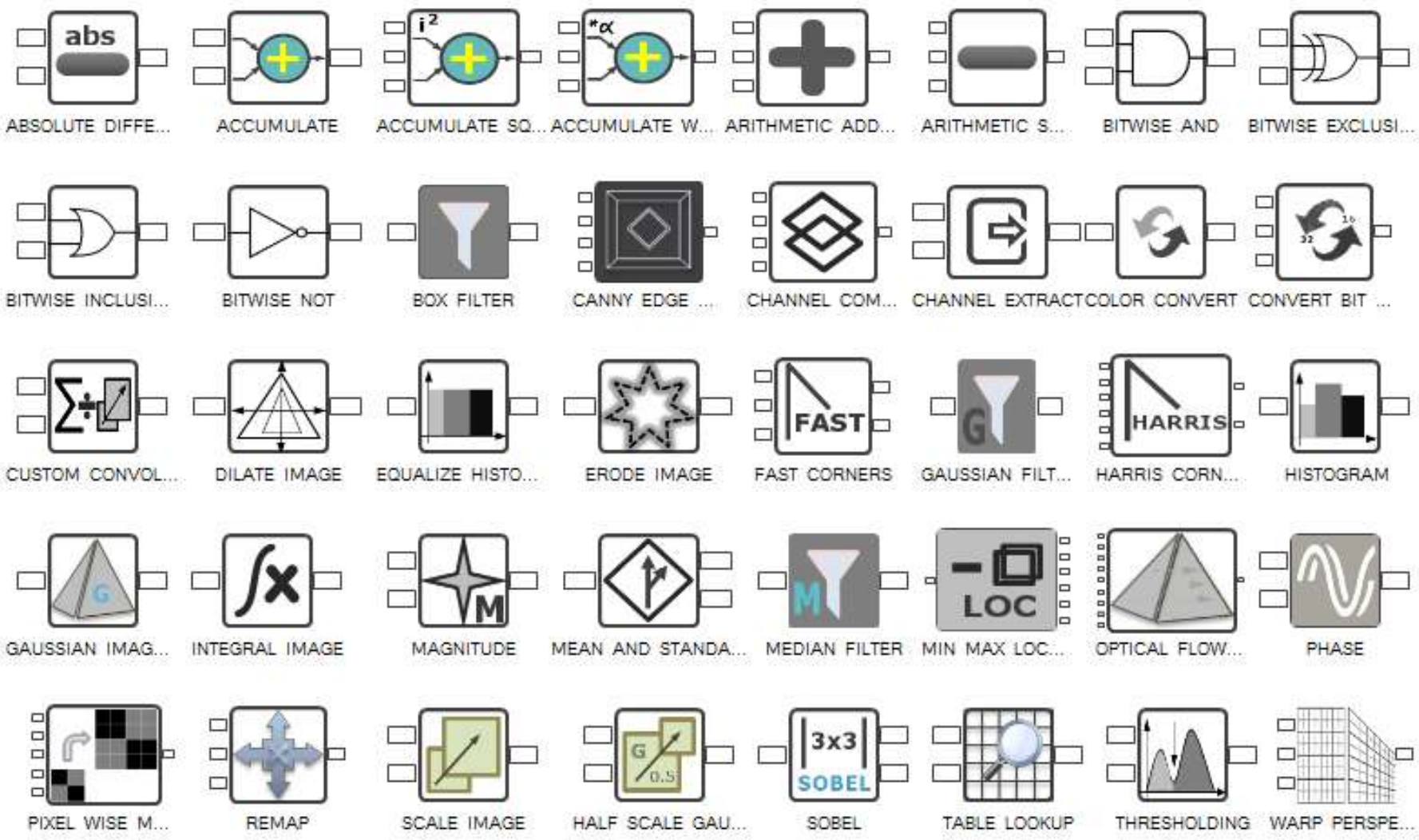


OpenVX *is a Khronos royalty free vision acceleration API* ***designed to map OpenCV*** *and other higher level vision libraries into optimized hardware implementations*

GC7000VX GPUs have EVIS™ *hardware optimized GC7000 implementations and hardware extensions to* ***optimize the throughput of vision*** *acceleration via OpenVX* ***without an intermediate API layer***

GC7000VX → ***17x performance improvement*** *vs standard GPU (non VX)* ***implementations for vision algorithms***

Optimized Kernels for OpenVX



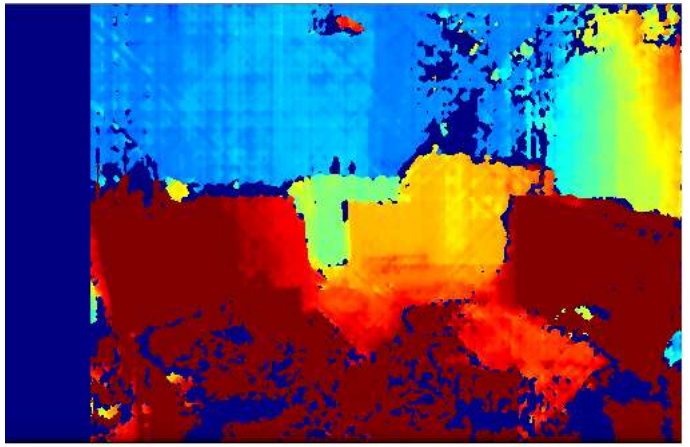
Future Optimized OpenVX Kernels to Be Provided

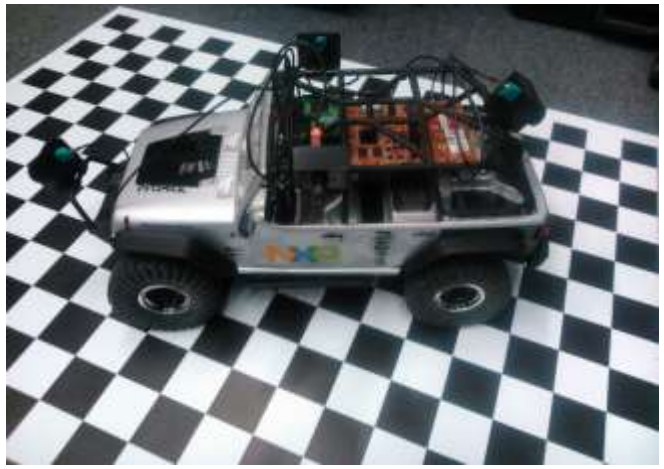
Filter Kernel 9x9
Sobel Filter XY
Grid Fusion
Haar Classifier
Convolutional Neural Network
ORB (FAST9 / BRIEF31)
Harris Corner
HoG + SVM

- Flexible DMA Descriptor
  - Up to 32 different ways to access image pixels (sliding window, row skips, column skips, etc.)
  - Ultimate flexibility: can program pixel-by-pixel coordinate sequence in local memory for DMA.
  - Per-thread conditional jumps => ideal for implementing classifier decision trees
- Enhanced Vision Instruction Set (EVIS)
  - High precision fixed point processing
  - Various dot product (DP) instructions

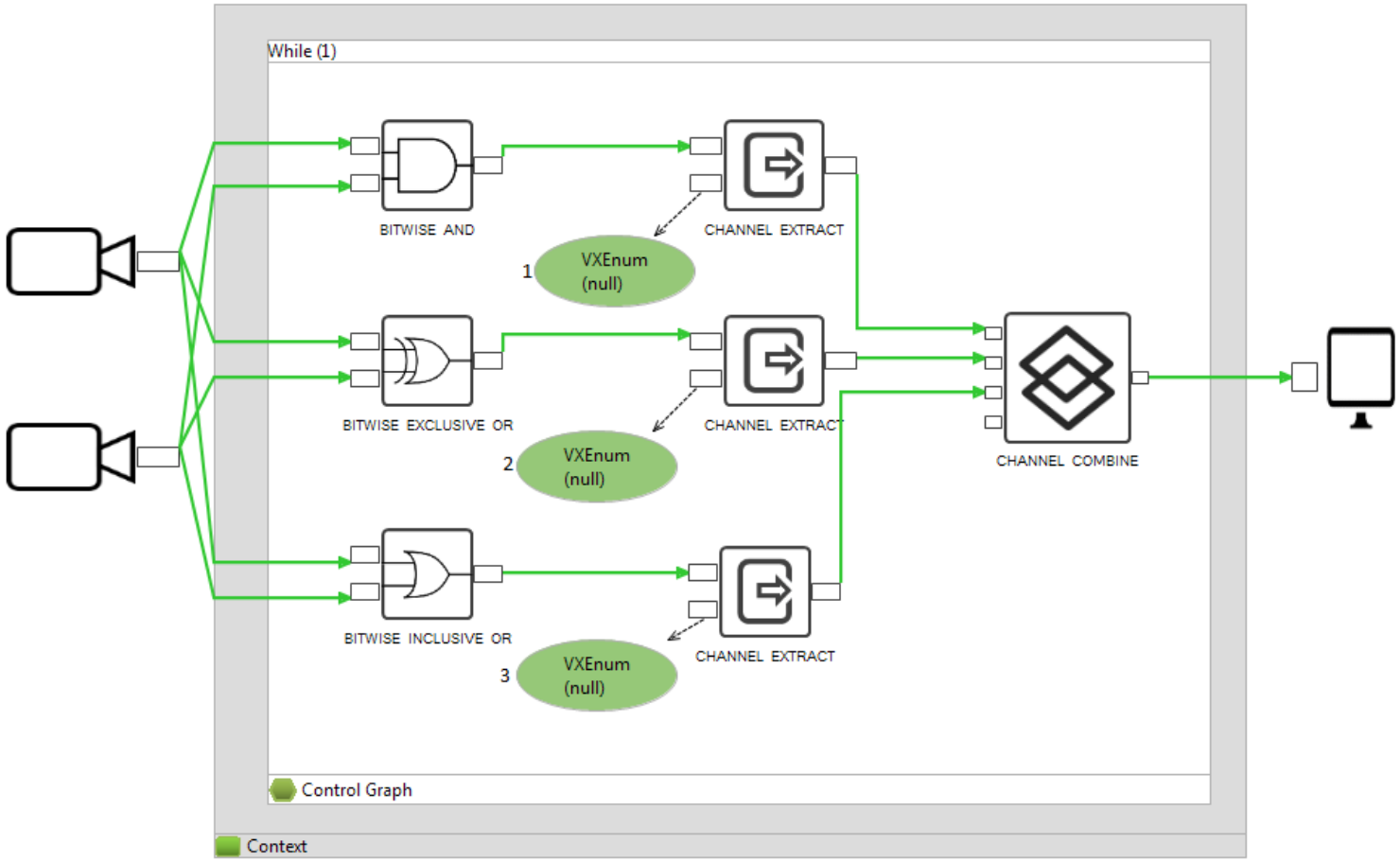
NXP Vision & Compute Enablement

- Algorithm R&D
  - Vision
  - Compute
  - Language processing
- SDK development
  - Enable rapid prototyping
  - Development & profiling tools
- Creating tools and demos to promote the capabilities of the i.MX 6 and i.MX 8





NXP – Drag and Drop Tool for Creating OpenVX Workflows



1: Enum Type = VX_CHANNEL_E, Enum Value = VX_CHANNEL_R
2: Enum Type = VX_CHANNEL_E, Enum Value = VX_CHANNEL_G
3: Enum Type = VX_CHANNEL_E, Enum Value = VX_CHANNEL_B

SECURE CONNECTIONS
FOR A SMARTER WORLD