

# Putting Eyes and ARMs on Machines

Sun Huo  
Product Marketing  
Mass Market Group  
Marketing Strategy Center



# ARM

# Agenda

---

- I. Putting ARMs on machines
  - Renesas' ARM-based silicon portfolio
- II. Putting eyes on machines
  - Embedded vision concepts and applications
- III. Future outlook and summary



# I. Putting ARMs on Machines

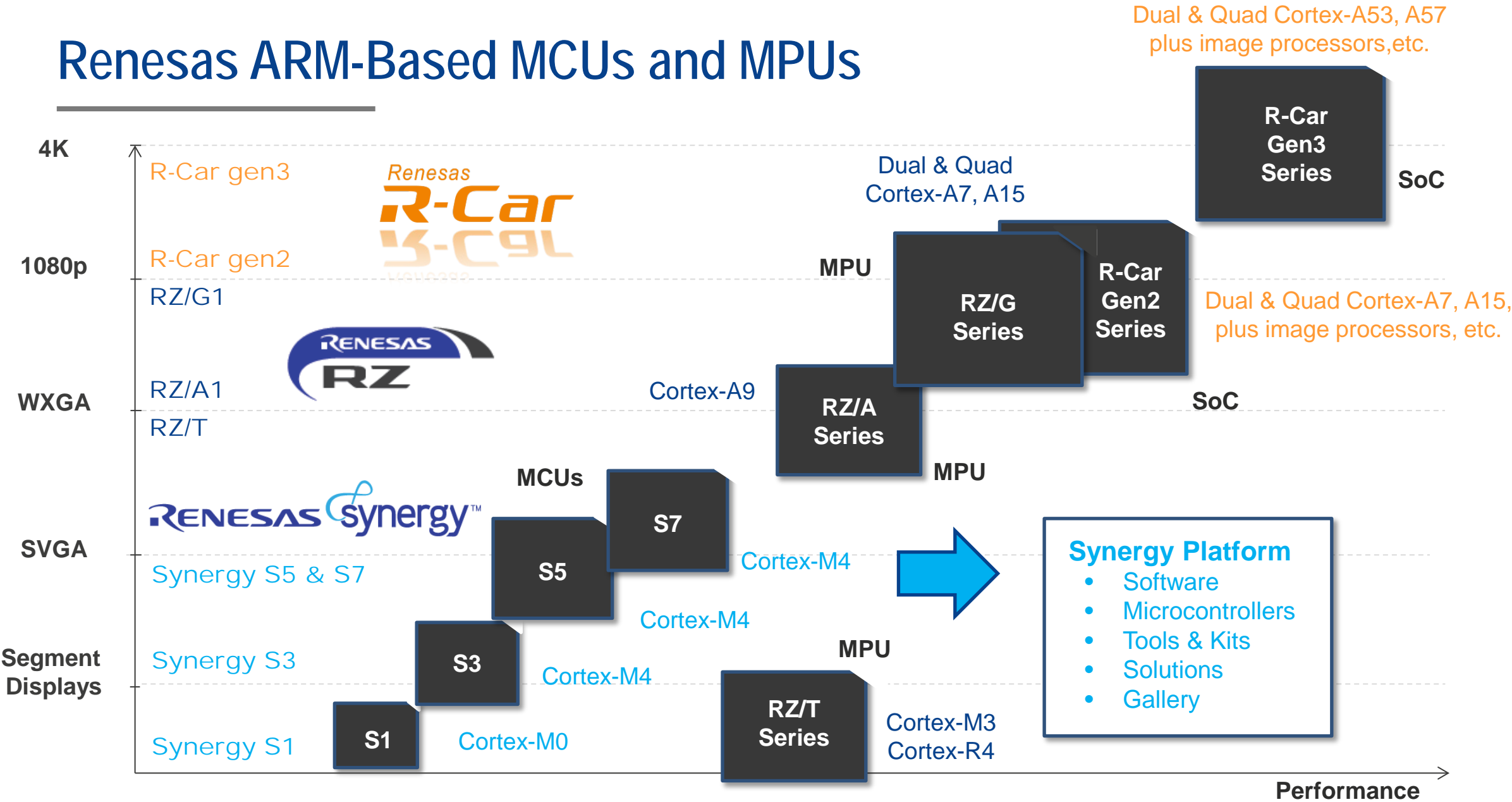
## Renesas' ARM-Based Silicon Portfolio



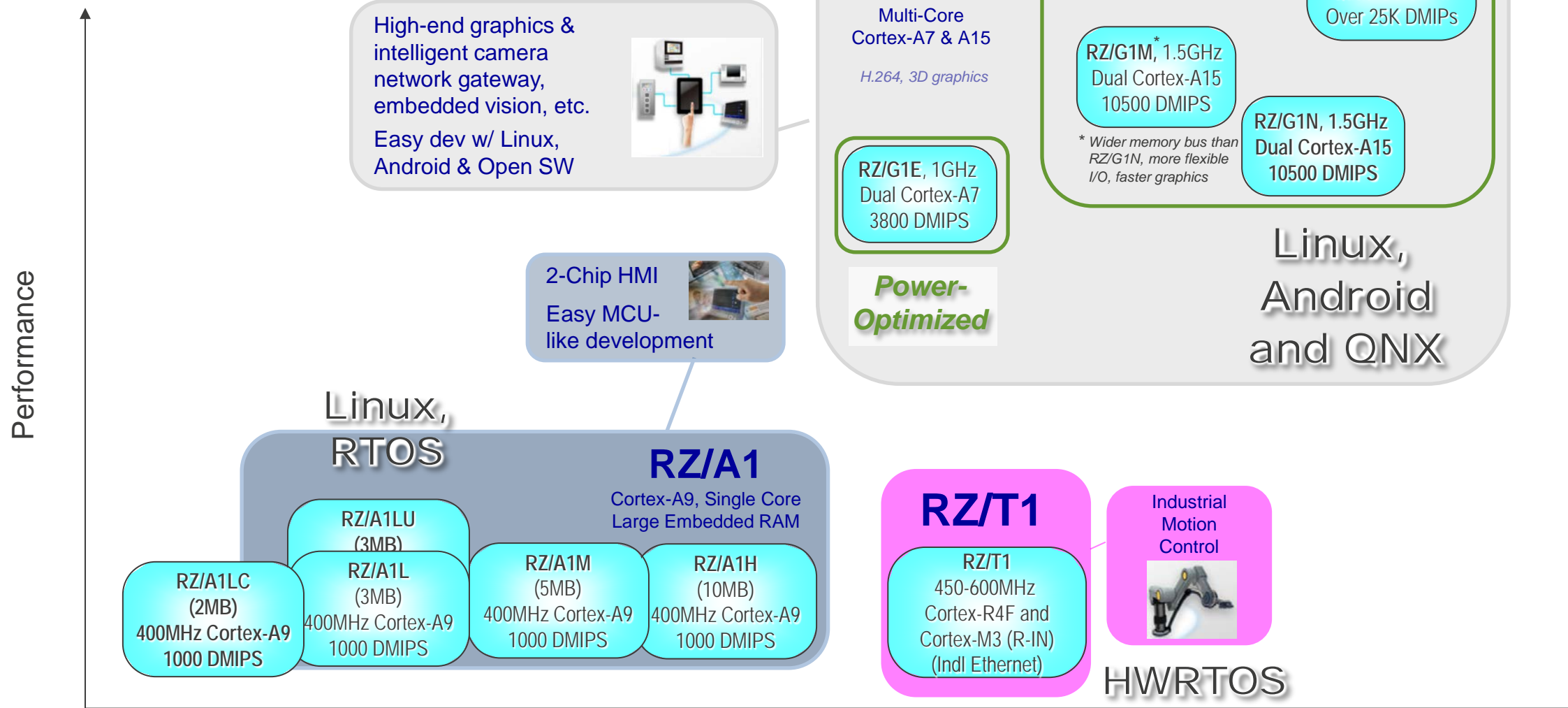
**ARM**



# Renesas ARM-Based MCUs and MPUs

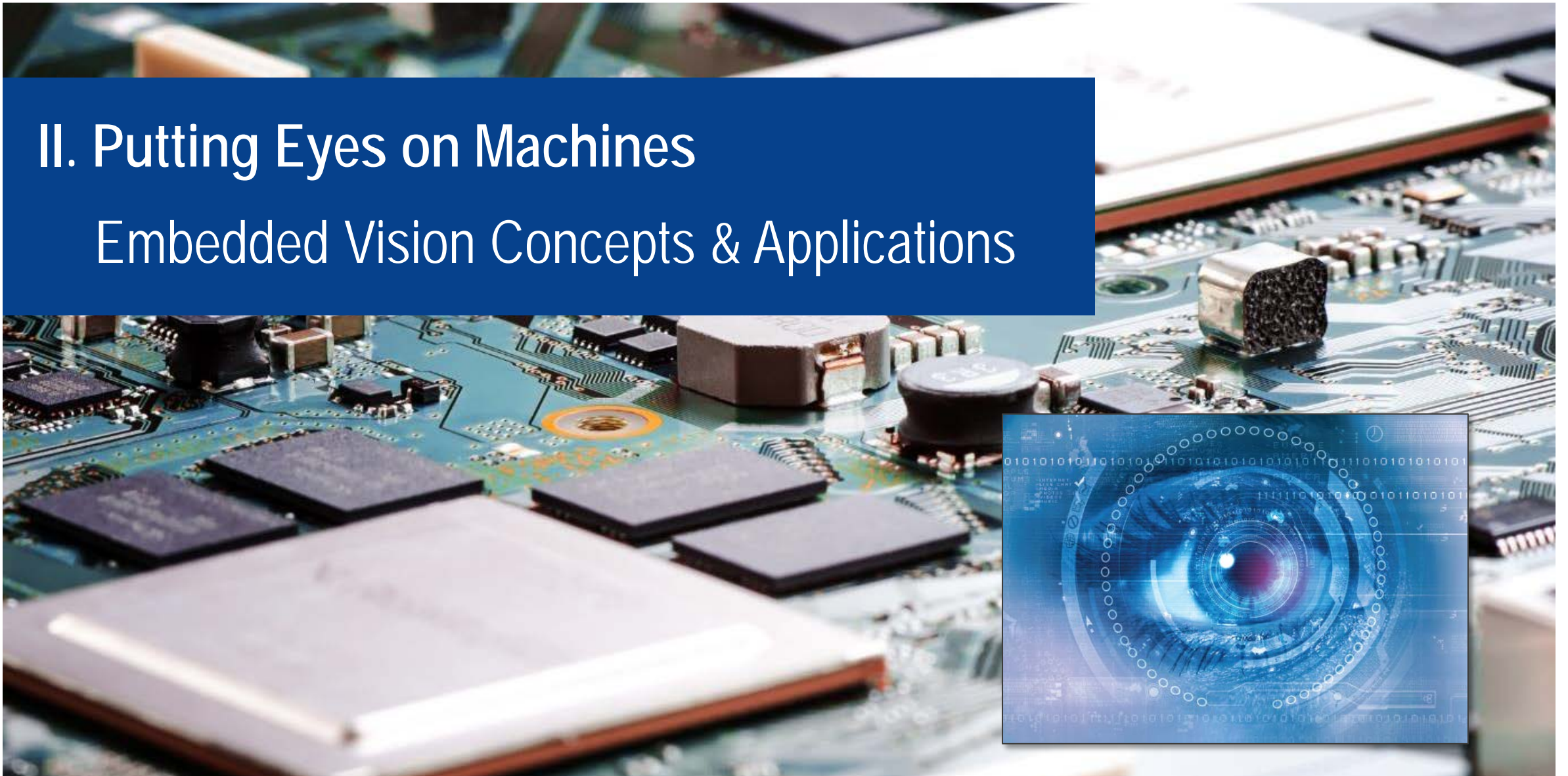


# RZ Family



## II. Putting Eyes on Machines

### Embedded Vision Concepts & Applications





# From the Electric-Eye to Modern Security Camera Systems

An **electric eye** is a [photodetector](#) used for detecting obstruction of a light beam

Modern photodetector systems use an infrared [light-emitting diode](#) modulated at a few kilohertz, which allows the detector to reject stray light and improves the range, sensitivity and security of the device.

## AUTOS ARE COUNTED BY ELECTRIC EYES

VEHICLES passing a given point are automatically counted by a new traffic-recording device just introduced. Two infra-red lamps, housed a short distance apart and mounted on one side of a road, cast invisible beams across the highway to a photo-electric receiving unit on the other side. Interruption of the two beams by an auto actuates an electrical counting device, which can be set to total the number of passing vehicles by the hour, day, week, or month. Pedestrians are not counted, since the apparatus is so constructed that it registers only when both beams are blocked at the same instant.

36

## Popular Science 1937



Passing vehicles intercept parallel beams of light and are counted by a photo-electric recording device

Copyrighted material  
POPULAR SCIENCE MONTHLY

## Outdoor Motion-Sensitive Light



## Laser Hallway



## Garage Door Safety Sensors

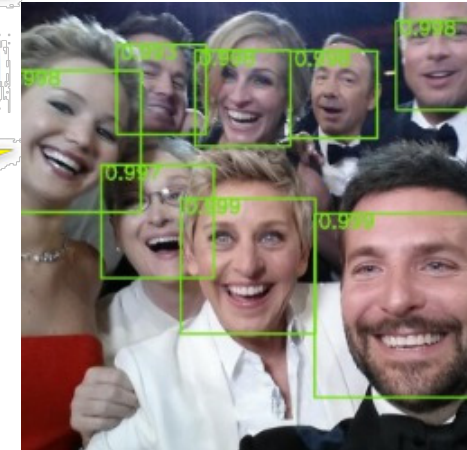
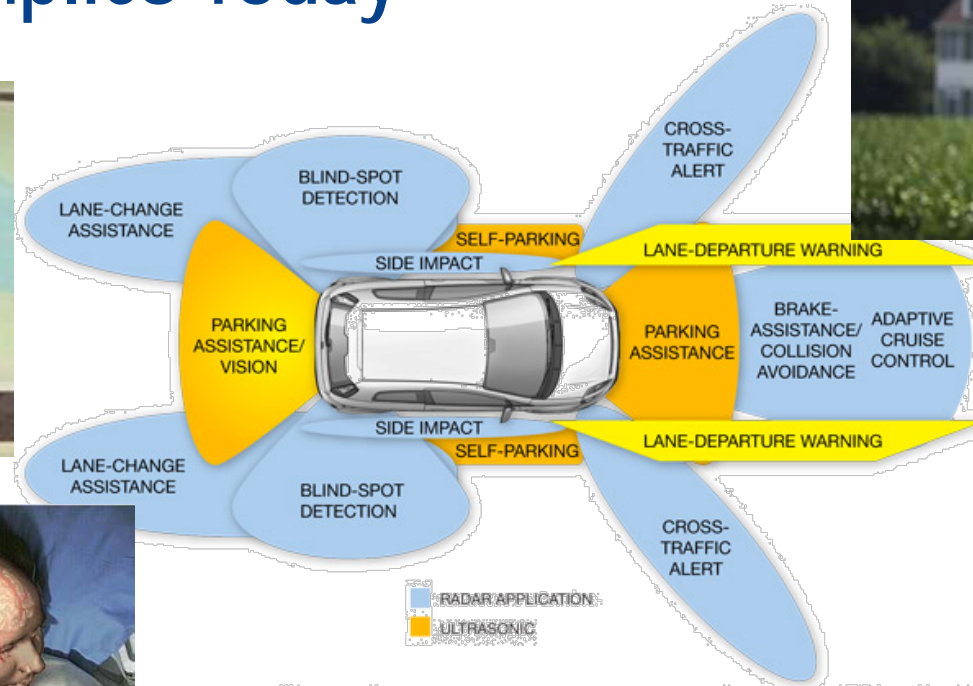


## Networked Security Systems





# What Embedded Vision Implies Today





# Embedded Computer Vision

---

## According to BDIT ([URL](#))

- “Embedded Vision” refers to the practical use of computer vision in machines that understand their environment through visual means, and the use of digital processing and intelligent algorithms to interpret meaning from images or video.

## Some key concepts in computer vision

- Image acquisition
- Image processing
  - **Motion detection**
  - **Feature detection**
  - **Object identification**



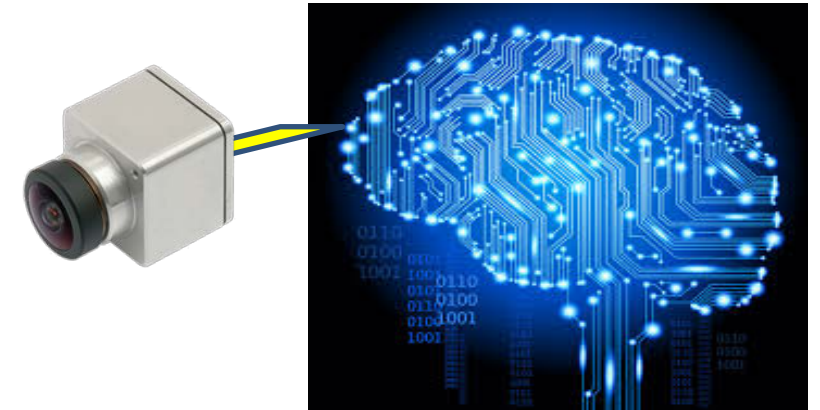
# Image Acquisition and Processing



# Image Acquisition

## Analog Cameras

- Used in cars, drones, FPV (first person view) controllers, etc.
- Max resolution ~720p due to wire bandwidth



## Digital camera sensors - CCD and CMOS

(2D arrays of tiny cells that convert light into electrons and often send raw digital data to semiconductor device for processing)

### CCD (Charge-Coupled Device)

- Move charge btwn. capacitive bins
- Very sensitive, very high-quality

### CMOS (Complementary Metal-Oxide Semiconductor)

- Simpler manufacturing and lower power
- Several transistors / cell, so less detection and more noise

## Serial digital cameras (MIPI CSI or LVDS)

- Send raw camera sensor data into MPU in high-speed serial format

## Serial digital camera module

- Send processed data in serial format to semiconductor device

## Parallel digital camera

- Send camera data to semiconductor device in RGB or YCbCR format

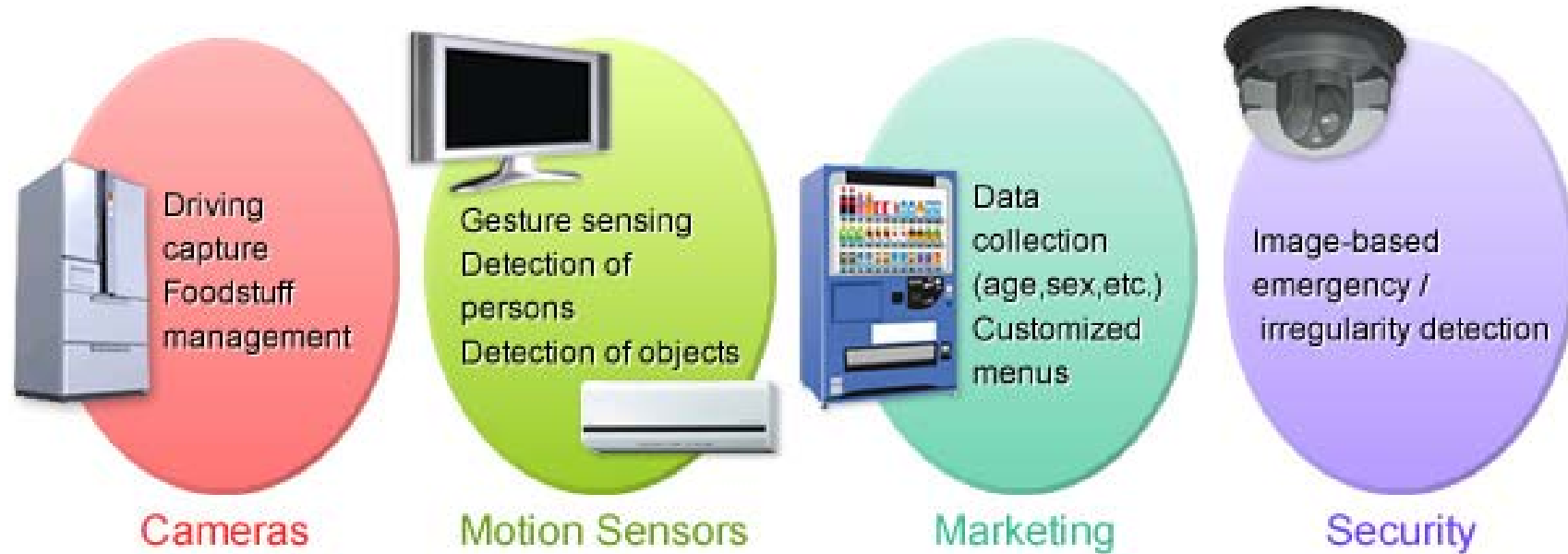


# Camera Modules Used in Industrial and Consumer Products



For GUI & Up to 4  
Digital Camera Inputs

Rapidly Growing CMOS-Camera Applications for HMLs





# Cameras Used in Consumer Drones

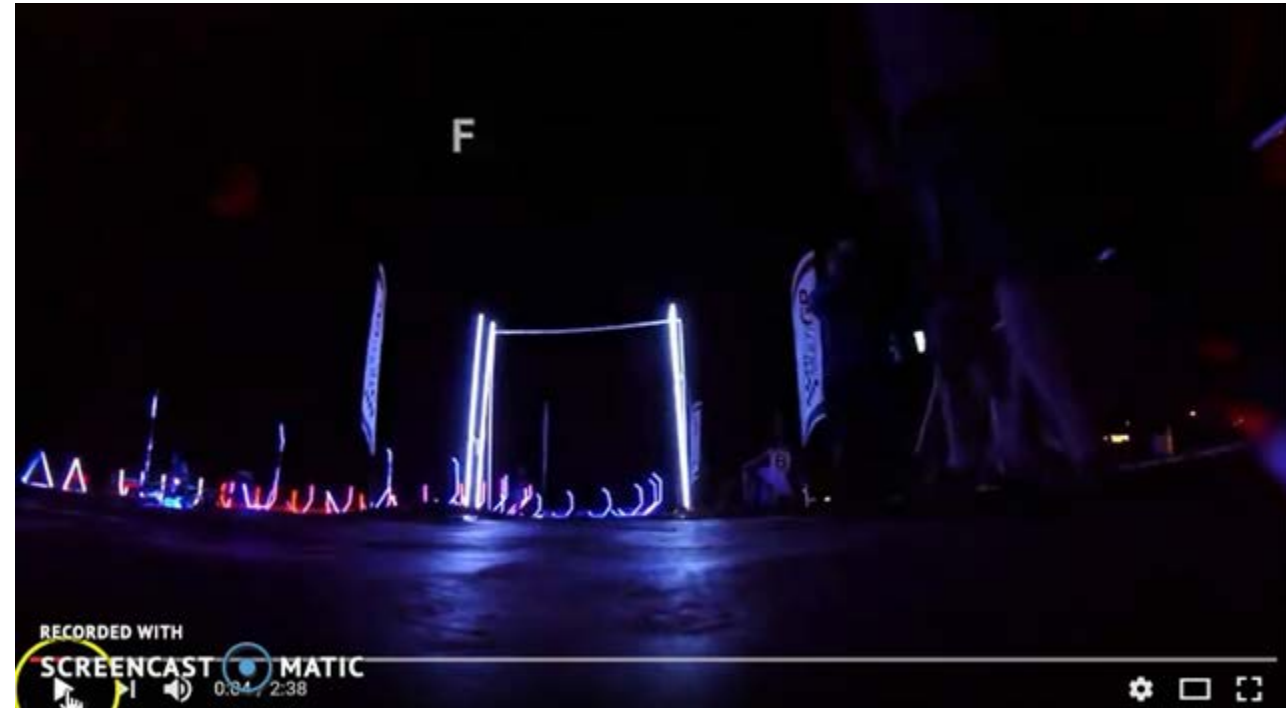
---

## Drones with auto-tracking & stabilization



Self-Flying Lily Drone ([URL](#))

## Drone racing



GoPro Awards: Epic Drone Race at Night ([URL](#))

# Motion Detection



# Miscellaneous Techniques Used in Motion Detection

---

Background subtraction

Temporal difference detection

Optical flow

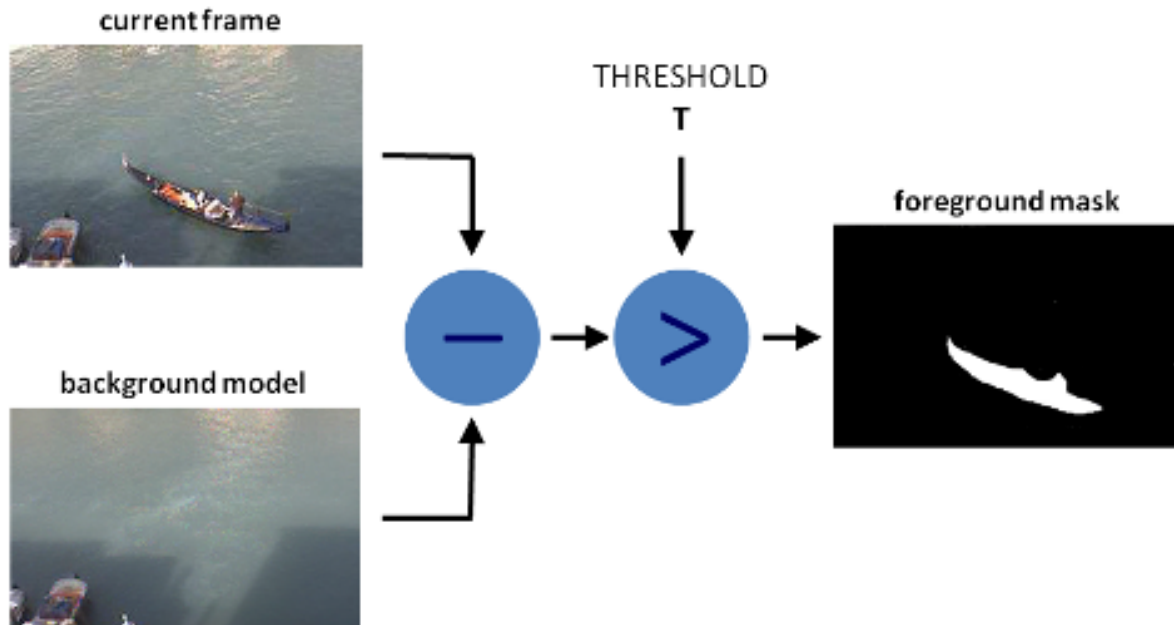
# Background Subtraction



# Background Subtraction

Motion is detected by comparing incoming frames to a known background image

- Foreground mask, which contains the pixels belonging to moving objects in the scene, is compared to a model of the background reference to determine if motion occurred



## Pros

- Reduces the amount of information needed to process in the algorithm
- Flexible and fast
- Good for stationary environments

## Cons

- It is only useful when camera is stationary
- With indoor scenes, reflections can lead to background changes.
- With outdoor scenes, wind, rain or lighting changes can trigger false detection



# Temporal Difference Detection



# Temporal Difference Detection (AKA Frame Differencing)

Algorithm that checks for the **difference** between two consecutive video **frames**

- If the pixels have changed, then something changed in the image

Often incorporates blur and threshold, to distinguish real movement from noise (e.g. when lighting conditions change, camera auto-focuses or corrects for brightness)

- Motion tracking occurs when change exceeds a certain threshold within “region of awareness”

## Pros

- This method is highly adaptive to dynamic (i.e. moving) environments

## Cons

- Does poor job of extracting complete shapes of moving objects



*Input first frame(a)*



*Input second frame(b)*



*Difference between two frame showing moving object*



*Binary image of difference image.*

# Renesas RZ/G1 Edge Blurring Example

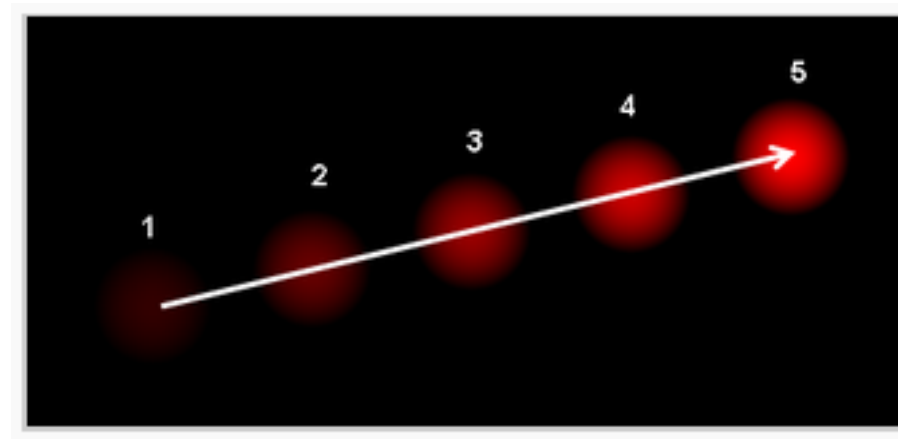
Deming the technique for making the image more stable, not the motion detection

<b>System</b> <ul style="list-style-type: none"><li>ARM Debugger (CoreSight)</li><li>DMAC</li><li>MMU</li><li>Interrupt Controller</li><li>3ch PLL/Module-standby</li></ul>	<b>RZ/G1M</b> Package : FC-BGA2727-831	<b>Interfaces</b> <ul style="list-style-type: none"><li>LBSC</li><li>Ex-BUS Interface (max16bit)</li><li>DBSC</li><li>DOR31-SDRAM BSC / 64bit</li><li>1ch USB2.0 host EHCI/OTG</li><li>1ch USB2.0 host/function High-speed module</li><li>3ch SDHI Support SDXC</li><li>1ch MMCIF</li><li>6ch I2C</li><li>6ch SCIF</li><li>3ch MSIOF</li><li>QSPI Single/Dual/Quad-SPI</li><li>3ch HSCIF</li><li>GPIO</li></ul>	
<b>Timers</b> <ul style="list-style-type: none"><li>Watchdog Timer</li><li>Timer pulse Unit 4ch / output PWM</li><li>Compare match Timer0 2ch / 16/32bit selectable</li><li>Compare match Timer1 8ch / 16/32/48bit selectable</li><li>Timer Unit 4sets of 3ch 32bit timer</li><li>7ch PWM timer</li></ul>	<b>CPU</b> <ul style="list-style-type: none"><li>ARM Cortex-A15 1.5GHz<ul style="list-style-type: none"><li>L1 I\$ 32KB</li><li>L1 D\$ 32KB</li></ul></li><li>ARM Cortex-A15 1.5GHz<ul style="list-style-type: none"><li>L1 I\$ 32KB</li><li>L1 D\$ 32KB</li></ul></li><li>L2 cache : 1024KB</li><li>NEON/VFPv4</li></ul>		
<b>Network</b> <ul style="list-style-type: none"><li>2ch CAN</li><li>Ethernet AVB 3 100 and 1000Mbps</li><li>Ethernet MAC 3 10 and 100Mbps</li></ul>	<b>Memory</b> <ul style="list-style-type: none"><li>RAM0 72KB</li><li>RAM1 4KB</li><li>RAM2 256KB</li></ul>		
	<b>Graphic IPs</b> <ul style="list-style-type: none"><li>520 MHz 3DGE PowerVR SGX544MP2</li><li>FDP1 De-interlacing module</li><li>VSP1 Input Format Converter Image Processor Output Format Converter</li><li>2D-DMAC Image extraction Image rotation/Inversion</li><li>VCP3 H.264 1920x1080@60x1ch</li><li>VIN 3ch Video Inputs</li><li>TSIF</li><li>DU Digital RGB 2ch outputs</li></ul>	<ul style="list-style-type: none"><li>1ch USB3.0 host</li><li>Serial-ATA</li><li>1lane PCI-Express</li><li>LVDS dot clock~148.5MHz</li><li>THS/TSC</li><li>Thermal Sensor</li></ul>	<b>Audio IPs</b> <ul style="list-style-type: none"><li>10ch SSI Serial Sound Interface</li><li>10ch SRC Sampling Rate Converter</li><li>ADG Audio clock generator</li></ul>





# Optical Flow



# Optical Flow

Form of temporal difference (i.e. it looks at successive frames)

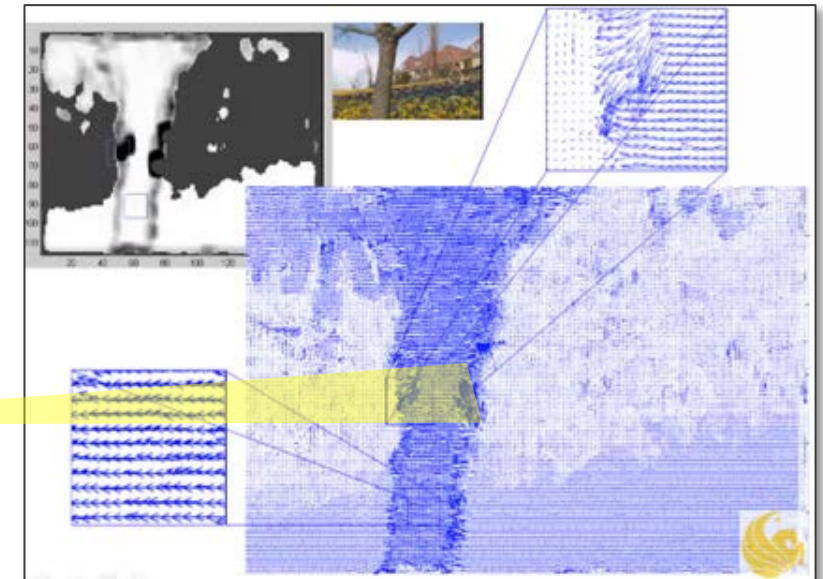
Algorithm determines displacement or “optical flow” vector (i.e. direction, gradient, and magnitude) for every pixel

- Gives you displacement of each pixel vs. previous frame.
- Lets you interpolate and extrapolate motion to predict position

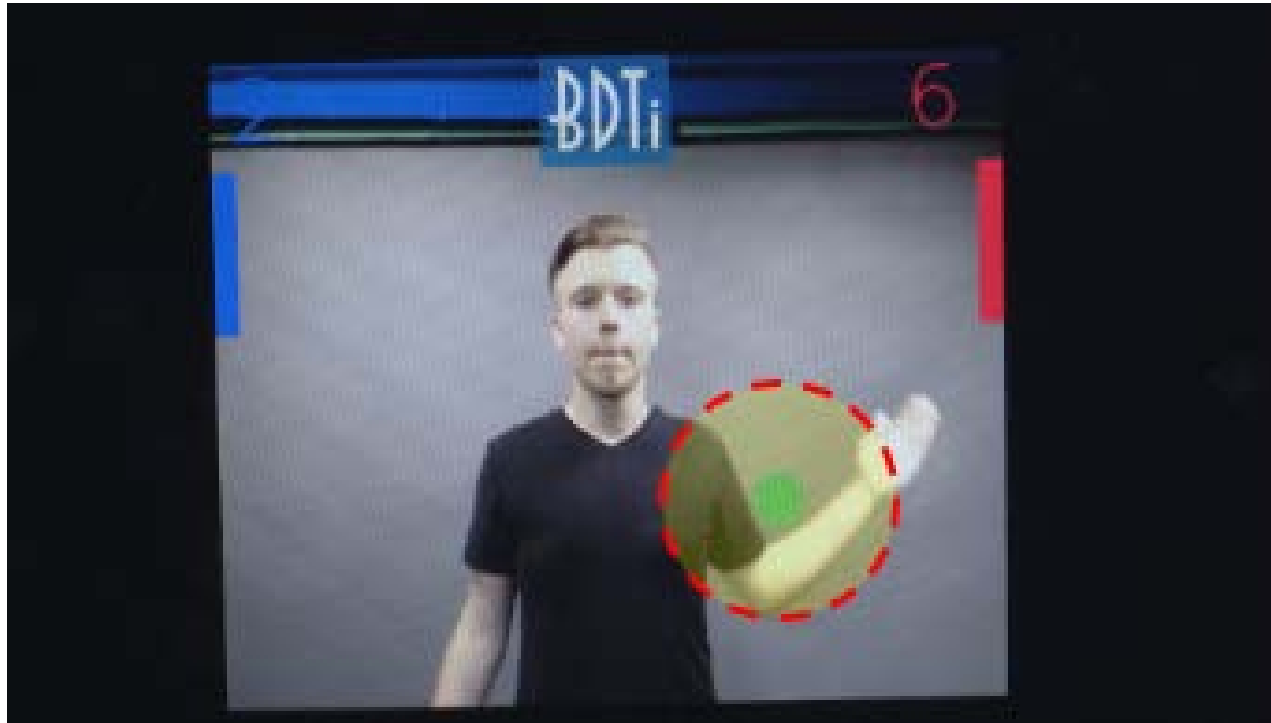
Relies on “Brightness and Constancy” assumption

- Corresponding points in two successive frames should not move more than a few pixels
  - Not great for very fast moving video

Interfaces between objects can cause confusion



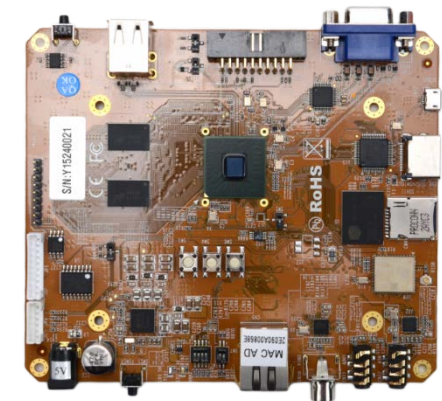
# Example – Optical Flow



Localize region of interest around the ball to minimize computational load, and achieve optical flow on low-cost, low-power RZ/G1E MPUs

(see also [URL](#))

<b>System</b> <ul style="list-style-type: none"><li>ARM Debugger (CoreSight)</li><li>DMAC</li><li>MMU</li><li>Interrupt Controller</li><li>3ch PLL/Module-standby</li></ul>	<b>RZ/G1E</b> Package : FC-BGA2121-501	<b>Interfaces</b> <ul style="list-style-type: none"><li>LBSC Ex-BUS interface (max16bit)</li><li>DBSC DDR3-SDRAM BSC / 32bit</li><li>1ch USB2.0 host EHCI/OHCI</li><li>1ch USB2.0 host/function High-speed module</li><li>3ch SDHI Support SDXC</li><li>1ch MMCIF</li><li>6ch I2C</li><li>6ch SCIF</li><li>3ch MSIOF</li><li>QSPI Single/Dual/Quad-SPI</li><li>3ch HSCIF</li><li>GPIO</li></ul>
<b>Timers</b> <ul style="list-style-type: none"><li>Watchdog Timer</li><li>Timer pulse Unit 4ch / output PWM</li><li>Compare match Timer0 2ch / 16/32bit selectable</li><li>Compare match Timer1 8ch / 16/32/48bit selectable</li><li>Timer Unit 4sets of 3ch 32bit timer</li><li>7ch PWM timer</li></ul>	<b>CPU</b> <ul style="list-style-type: none"><li>ARM Cortex-A7 1.0GHz</li><li>ARM Cortex-A7 1.0GHz</li><li>L1 I\$ 32KB</li><li>L1 D\$ 32KB</li><li>L1 I\$ 32KB</li><li>L1 D\$ 32KB</li><li>L2 cache : 512KB</li><li>NEON/VFPv4</li></ul>	
<b>Network</b> <ul style="list-style-type: none"><li>2ch CAN</li><li>Ethernet AVB 100 and 100Mbps</li><li>Ethernet MAC 10 and 100Mbps</li></ul>	<b>Memory</b> <ul style="list-style-type: none"><li>RAM0 72KB</li><li>RAM1 4KB</li><li>RAM2 256KB</li></ul>	<b>Audio IPs</b> <ul style="list-style-type: none"><li>10ch SSI Serial Sound Interface</li><li>6ch SRC Sampling Rate Converter</li><li>ADG Audio clock generator</li></ul>
	<b>Graphic IPs</b> <ul style="list-style-type: none"><li>260 MHz 3DGE (PowerVR SGX540)</li><li>FDP1 De-interlacing module</li><li>VSP1 Input Format Converter Image Processor Output Format Converter</li><li>2D-DMAC Image extraction Image rotation/inversion</li><li>VCP3 H.264 1920x1080@60x1ch</li><li>VIN 2ch Video inputs</li><li>DU Digital RGB 2ch outputs</li></ul>	





# Feature Detection

# Miscellaneous Techniques and Tools Used in Feature Identification

---

## Calculation-based techniques

- Like edge detection

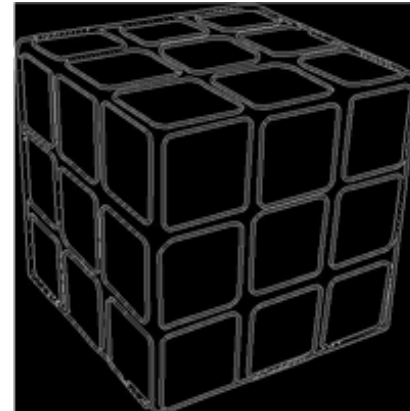
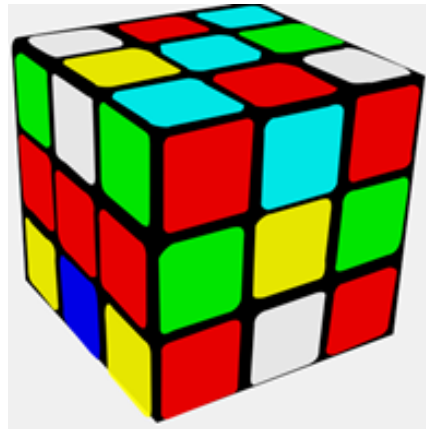
## Template matching

- Simple shapes and more complex cascade classifiers

## What do I mean by “feature”?

- Edges, corners, points, faces, codes, text, etc.

# Edge Detection





# Edge Detection

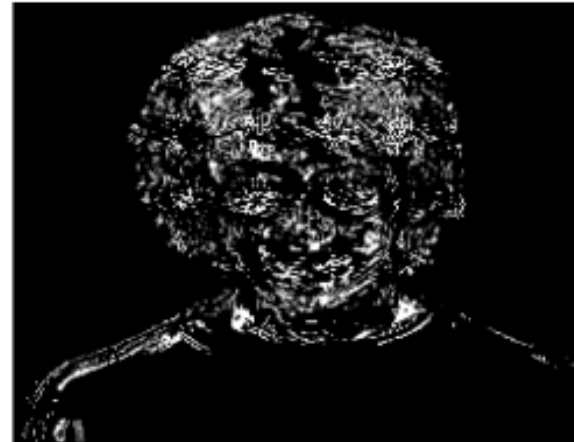
---

**Edge detection** is an image processing technique for finding the boundaries of objects within images

- It works by **detecting** discontinuities in brightness
- Used for image segmentation and data extraction in image processing and computer vision
- Often used as an intermediate step to reduce the amount of data you need to process

Common edge detection algorithms include Sobel, Canny, Prewitt, Roberts, and fuzzy logic

- Canny (multi-stage) operation shown below



# Edge Detection Renesas RZ/G1M

Edge detection running on 1080p 30  
real-time video playback on RZ/G1M

System	RZ/G1M		Interfaces	
ARM Debugger (CoreSight)	Package : FC-BGA2727-831		LBSC	
DMAC			Ex-BUS Interface (max16bit)	
MMU			DBSC	
Interrupt Controller			DDR3L-SDRAM BSC / 64bit	
3ch PLL/Module-standby			1ch USB2.0 host EHCI/OHCI	
Timers	CPU		1ch USB2.0 host/function High-speed module	
Watchdog Timer	ARM Cortex-A15 1.5GHz		3ch SDHI Support SDXC	
Timer pulse Unit 4ch / output PWM	L1 I\$ 32KB L1 D\$ 32KB		1ch MMCIF	
Compare match Timer0 2ch / 16/32bit selectable	L2 cache : 1024KB		6ch I2C	
Compare match Timer1 8ch / 16/32/48bit selectable	NEON/VFPv4		6ch SCIF	
Timer Unit 4sets of 3ch 32bit timer	Memory		3ch MSIOF	
7ch PWM timer	RAM0 72KB RAM1 4KB RAM2 256KB		QSPI	
Network	Graphic IPs		Single/Dual/Quad-SPI	
2ch CAN	520 MHz 3DGE PowerVR SGX544MP2		3ch HSCIF	
Ethernet AVB 8x3 100 and 1000Mbps	VSP1 Input Format Converter Image Processor Output Format Converter		GPIO	
Ethernet MAC 8x3 10 and 100Mbps	VCP3 H.264 1920x1080@60x1ch		1ch USB3.0 host	
	TSIF		Serial-ATA	
	2D-DMAC Image extraction Image rotation/Inversion		1lane PCI-Express	
	VIN 3ch Video Inputs		LVDS dot clock~148.5MHz	
	DU Digital RGB 2ch outputs		THS/TSC Thermal Sensor	
			Audio IPs	
			10ch SSI Serial Sound Interface	
			10ch SRC Sampling Rate Converter	
			ADG Audio clock generator	



# vSLAM - Visual Simultaneous Localization and Mapping

Autonomous robots must be able to explore their environment without user-intervention or external data, build a reliable map, and localize themselves within it

- No initial map needed
- Determine appropriate reference points, correct for slippage, collisions, relocation
- Handles dynamic changes in the environment (lighting, moving objects and/or people)



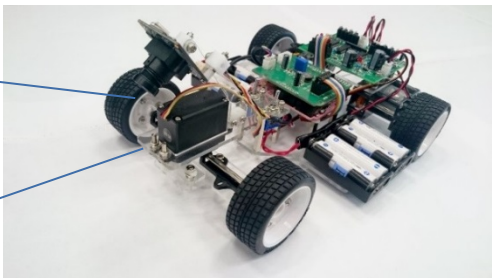
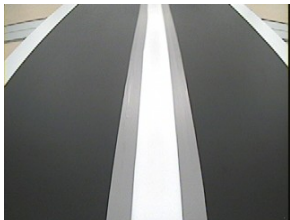
Roomba SDK available for hobbyists and inventors

vSLAM utilizes edge detection for localization for this purpose

- Vision and odometry-based, enabling low-cost navigation in cluttered / populated environments



# Renesas RZ/A Autonomous Car

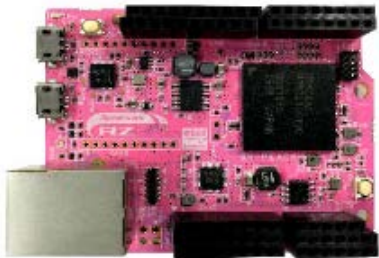
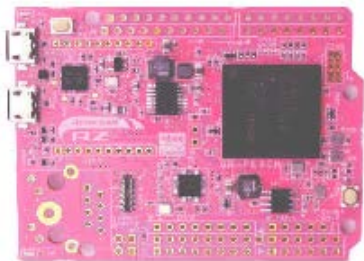


## RZ/A1

<b>ARM™ Cortex A9</b> 400MHz 1000DMIPS <b>Floating point Unit</b> <b>NEON</b> <b>MMU</b> <b>On-Chip Debug</b>	<b>Imaging</b> CMOS Camera Imager Interface (Video In) LCD Controller Interface 2 channels RGB, LVDS Video out OpenVG 1.1 2D Graphics Engine JPEG Codec Unit IMR-LS NTSC/PAL Decoder	<b>External Memory Interface</b> Bus State Controller SDRAM, NOR Flash NAND Flash I/F SPI Flash Interface SPI Multi I/O 2 channels MMC I/F SD Card Host Interface 2 channels CDROM Decoder	<b>Communication</b> USB 2.0 H/D 2 high-speed ch. SCI 10 channels I2C 4 channels SPI 5 channels QSPI 2 channels 10/100 Ethernet SSI 6 channels SPDIF 1 channel IE Bus 1 channels CAN 5 channels LIN 2 channels	<b>Timers</b> Motor Timer MTU2 16 bit, 5 channels OS Timer 32 bit, 2 channels PWM Timer 10 bit, 16 channels Watchdog Timer Real-Time Clock <b>Misc</b> Asynchronous Audio sampling rate converter Sound Generator Display Output Compare IrDA	<b>System</b> ADC 12-bit, 8 channels Encryption Engine (Optional) Unique Customer ID (Optional) Standby modes Sleep/Software/Deep JTAG Clock Generation with Spread Spectrum Clocking Interrupt Controller DMA 16 channels
	<b>Internal Memory</b> SRAM (10MB, 5MB) L1 I-Cache 32KB L1 D-Cache 32KB L2 Cache 128KB	<b>Package</b> 324 BGA 19mmx19mm, 0.8mm pitch 256 QFP 28mmx28mm, 0.4mm pitch 256 BGA 11mmx11mm, 0.5mm pitch			

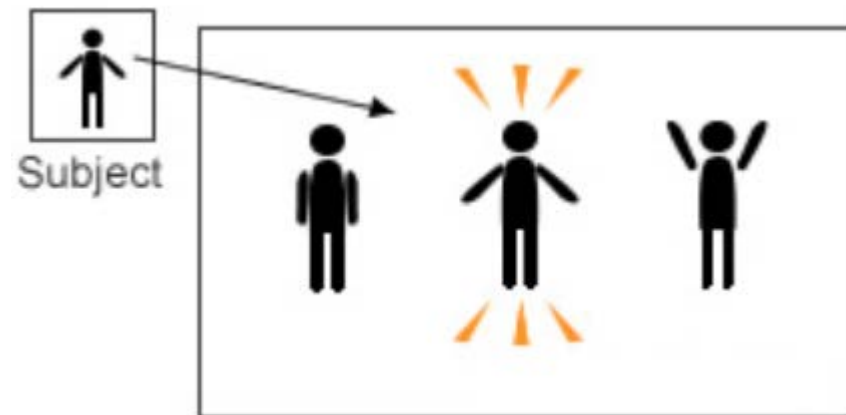
(See also [URL](#))

## ARM MBED “GR Peach” Board



World’s only Cortex A-Class MBED product

# Template Matching



# Template Matching

Digital image processing technique for finding small parts of an image, which **match** a pre-defined **template**

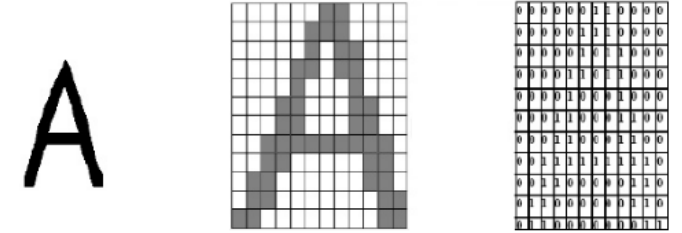
- A template is a small image inside a larger image.
- There is a similarity criteria (metric) – an equation of correlation that your algorithm follows to perform that matching

Flexible and relatively easy to use, which makes them very popular for object localization, especially in machine vision

Can be used in motion detection, robot navigation, object-detection and in manufacturing as a part of quality control

- Applicability is limited by computational power available
- Big and complex templates can be time-consuming to identify

## Character Recognition



## Handwriting Recognition



## UPC Code



## QR Code



# Running Z-Bar on RZ/A

## Renesas RZ/A running Z-Bar Code



(See also [URL](#) – from 1 min, 30 sec onward)



“Open source software suite for reading bar codes from various sources, such as video streams, image files and raw intensity sensors

It supports many popular symbologies (types of bar codes) including EAN-13/UPC-A, UPC-E, EAN-8, Code 128, Code 39, Interleaved 2 of 5 and QR Code

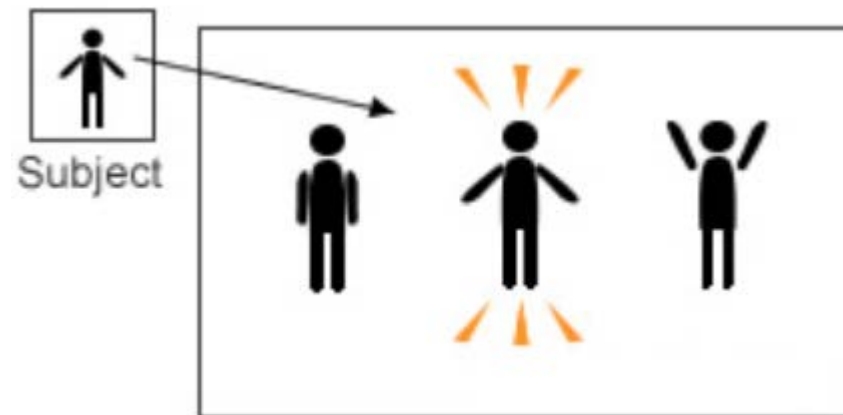
ZBar is licensed under the [GNU LGPL 2.1](#) to enable development of both open source and commercial projects.”

<http://zbar.sourceforge.net/>



# Object Identification

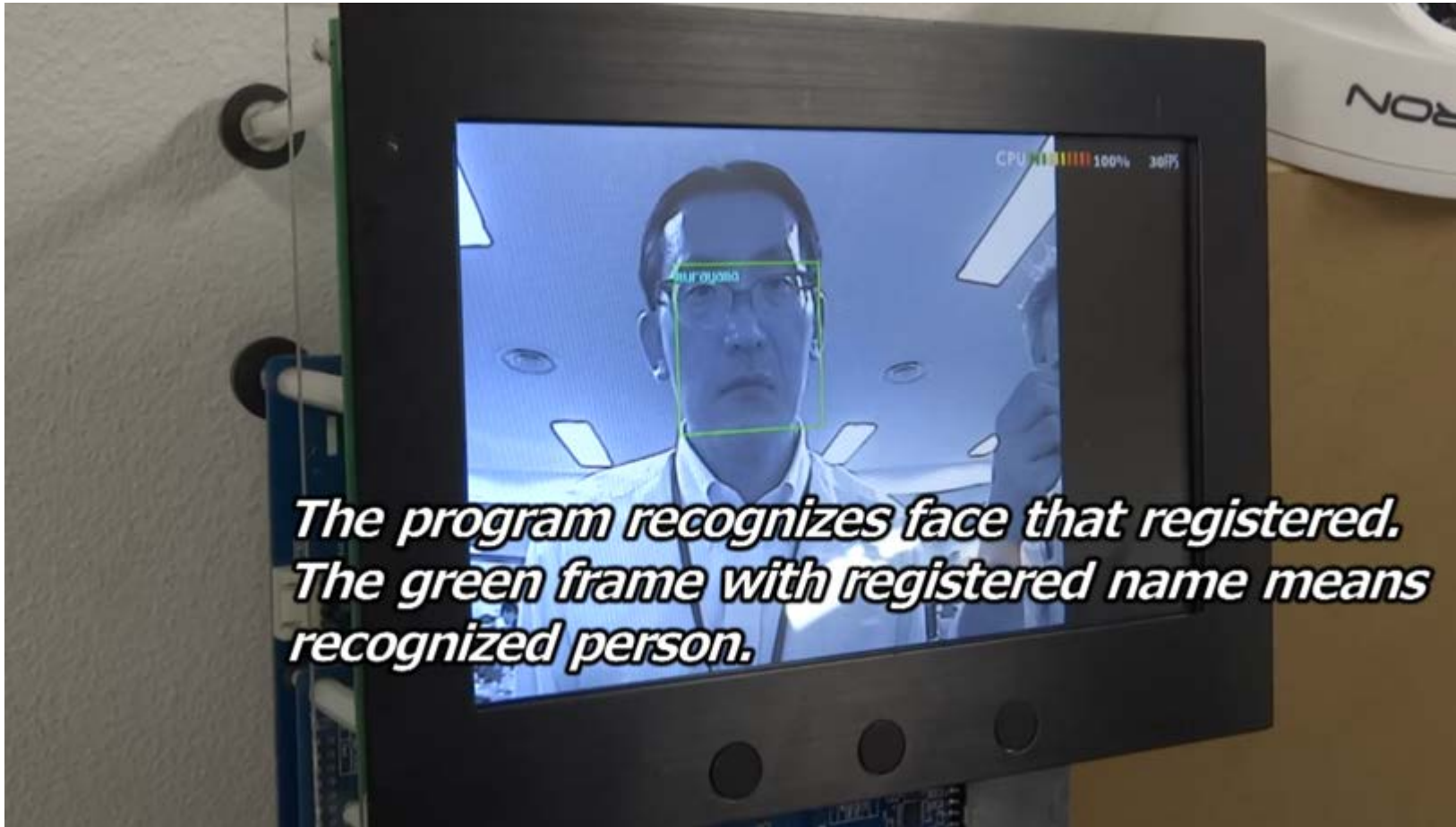
# More Complex Template Matching



# Renesas RZ/A Face, Gender and Age Identification Example

Face detection  
by template-  
matching

Limited to  
frontal view,  
relatively  
stationary  
subject



(See also [URL](#))

# Face Identification Using Constellation Model

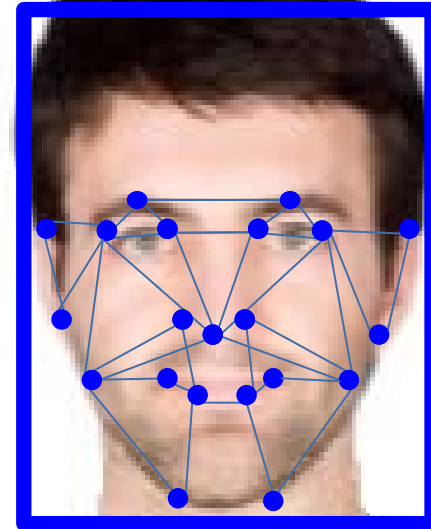
---

**Constellation model** performs category-level object recognition

A “**part-based**” model

- It attempts to represent classes of objects by a set of parts that are related by geometric constraints
- It models relative location, relative scale, and appearance of the parts in a given object category

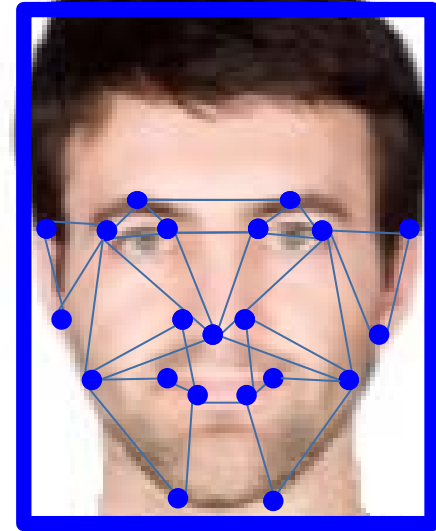
Model parameters are estimated using **unsupervised learning** algorithm, so visualization of an object class can be extracted from an unlabeled set of training images



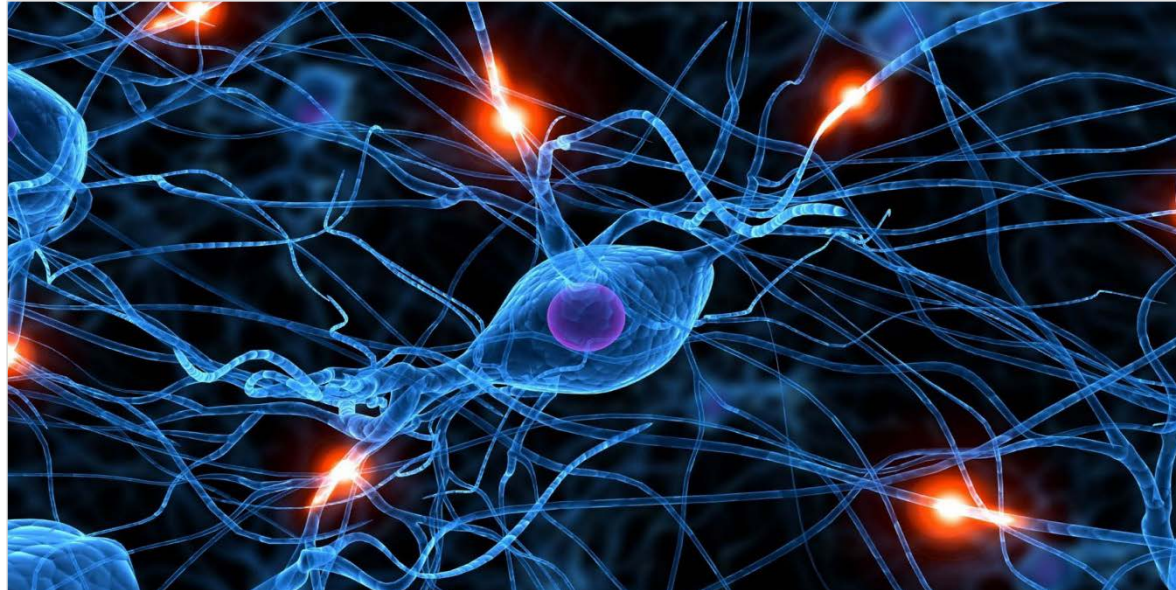


# Face Identification Using Constellation Model

But how do we make it rotation-invariant and less sensitive to temporal fluctuations in subjects' appearance?



# Neural Networks



# Artificial Neural Networks (ANN)

## A type of machine learning algorithm

## Loosely modelled after how the brain works

# A framework for solving predictive and analytical problem

Has exhibited success in solving hard, big problems that rely on a lot of data, including games, natural language interpretation, Big Data trend analyses, and vision

It uses **supervised** regressions (vs. unsupervised regressions). This process is called training



*“Deep Learning  
Machine Teaches Itself  
Chess in 72 Hours.  
Plays at International  
Master Level!” ([URL](#))*



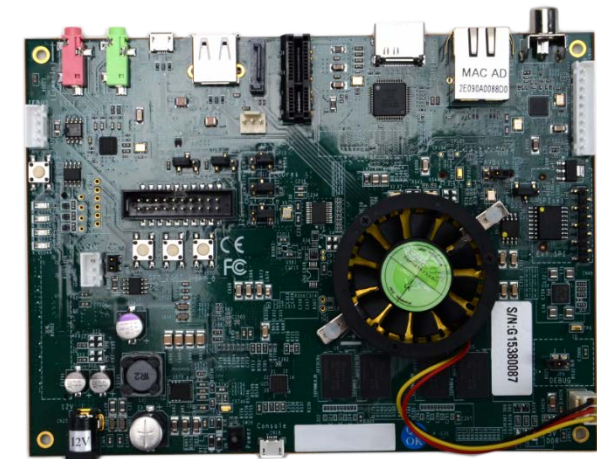
# Example – Image Identification

Uses partner software

<b>System</b> <ul style="list-style-type: none"><li>ARM Debugger (CoreSight)</li><li>DMAC</li><li>MMU</li><li>Interrupt Controller</li><li>3ch PLL/Module-standby</li></ul>	<b>RZ/G1M</b> Package : FC-BGA2727-831	<b>Interfaces</b> <ul style="list-style-type: none"><li>LBSC</li><li>Ex-BUS Interface (max16bit)</li><li>DBSC</li><li>DOR3L-SDRAM BSC / 64bit</li><li>1ch USB2.0 host EHCI/OHCI</li><li>1ch USB2.0 host/function High-speed module</li><li>3ch SDHI Support SDXC</li><li>1ch MMCIF</li><li>6ch I2C</li><li>6ch SCIF</li><li>3ch MSIOF</li><li>QSPI Single/Dual/Quad-SPI</li><li>3ch HSCIF</li><li>GPIO</li></ul>
<b>Timers</b> <ul style="list-style-type: none"><li>Watchdog Timer</li><li>Timer pulse Unit 4ch / output PWM</li><li>Compare match Timer0 2ch / 16/32bit selectable</li><li>Compare match Timer1 8ch / 16/32/48bit selectable</li><li>Timer Unit 4sets of 3ch 32bit timer</li><li>7ch PWM timer</li></ul>	<b>CPU</b> <ul style="list-style-type: none"><li>ARM Cortex-A15 1.5GHz<ul style="list-style-type: none"><li>L1 I\$ 32KB</li><li>L1 D\$ 32KB</li></ul></li><li>ARM Cortex-A15 1.5GHz<ul style="list-style-type: none"><li>L1 I\$ 32KB</li><li>L1 D\$ 32KB</li></ul></li><li>L2 cache : 1024KB</li><li>NEON/VFPv4</li></ul>	
<b>Network</b> <ul style="list-style-type: none"><li>2ch CAN</li><li>Ethernet AVB 100 and 1000Mbps</li><li>Ethernet MAC 10 and 100Mbps</li></ul>	<b>Memory</b> <ul style="list-style-type: none"><li>RAM0 72KB</li><li>RAM1 4KB</li><li>RAM2 256KB</li></ul>	
	<b>Graphic IPs</b> <ul style="list-style-type: none"><li>520 MHz 3DGE PowerVR SGX544MP2</li><li>VSP1 Input Format Converter Image Processor Output Format Converter</li><li>VCP3 H.264 1920x1080@60x1ch</li><li>TSIF</li><li>FDP1 De-interlacing module</li><li>2D-DMAC Image extraction Image rotation/Inversion</li><li>VIN 3ch Video Inputs</li><li>DU Digital RGB 2ch outputs</li></ul>	<b>Audio IPs</b> <ul style="list-style-type: none"><li>10ch SSI Serial Sound Interface</li><li>10ch SRC Sampling Rate Converter</li><li>ADG Audio clock generator</li></ul>
		<ul style="list-style-type: none"><li>1ch USB3.0 host</li><li>Serial-ATA</li><li>1lane PCI-Express</li><li>LVDS dot clock~148.5MHz</li><li>THS/TSC Thermal Sensor</li></ul>



(See also [URL](#))





# “Fog Computing” Approach – Sushi Demo



(See also [URL](#))

## Example IoT application

- Object detection at endpoint
- CNN in cloud (host)

By partner **Uncanny Vision**  
on **GR Peach Board**





### III. Future Outlook and Summary



# Summary and Future Outlook

---

Interest in embedded machine vision is growing - it is dragging us into the realm of AI  
Motion, feature, and gesture recognition can fit into industrial and automotive products

Image **detection** vs. **identification** have different computational requirements

Renesas can enable your embedded vision needs

What we do with this technology can help make the world a better, safer, more efficient place

Use it! But use it responsibly! ☺



VS.









Renesas Electronics America Inc.

© 2016 Renesas Electronics America Inc. All rights reserved.