NXP PRESENTS

AiM ARTIFICIAL INTELLIGENCE in MOBILITY

WEBINAR 3 :

CAMERA INTERFACE
OPENCV
QT VIEWER

IN COLLABORATION WITH

Time Of Sports®

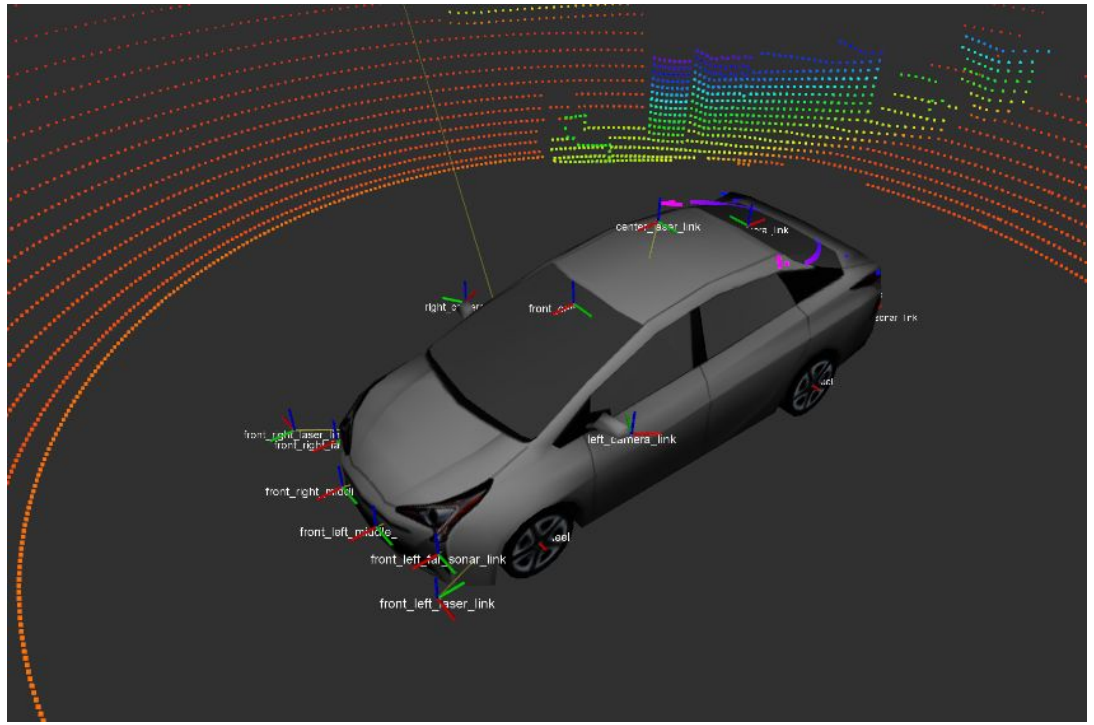
Agenda

- Sensors in Gazebo : Camera Sensor
- Open CV – An Overview
- QT Viewer – An Overview

SENSORS IN GAZEBO

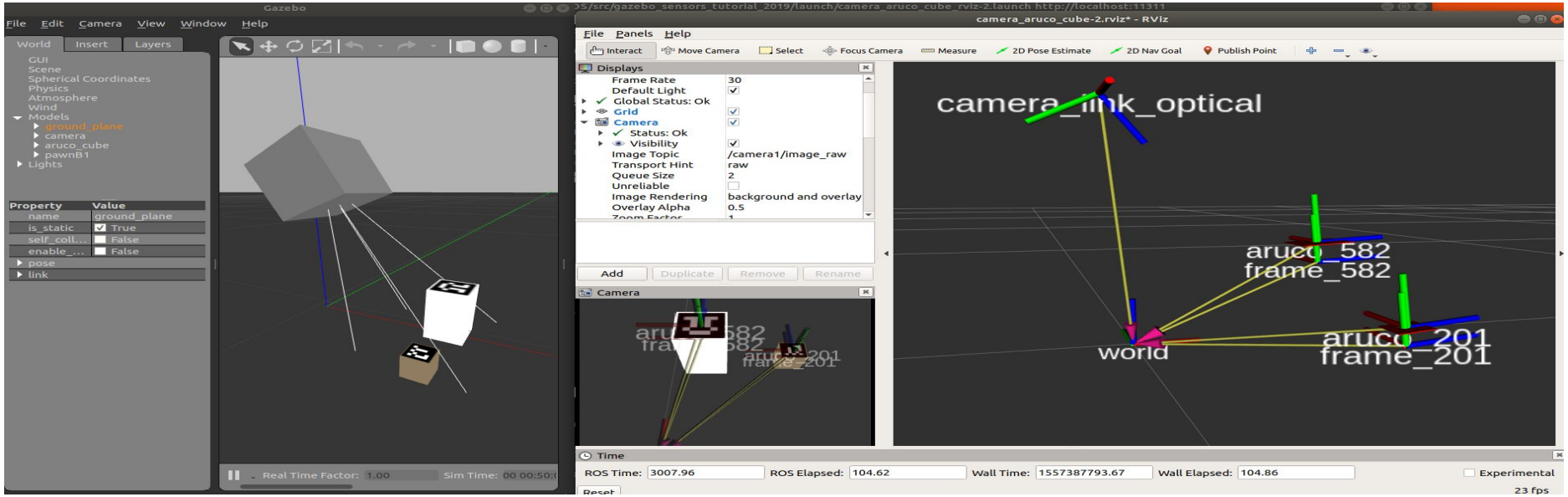
- Gazebo supports models of many common sensors :
  - Camera
  - Depth Camera
  - Contact Sensor
  - Force Torque Sensor
  - Laser
  - IMU (Inertial Measurement Unit)
  - RFID
  - Sonar
  - Wide Angle Camera
  - Wireless Receiver

(http://osrf-distributions.s3.amazonaws.com/gazebo/api/8.2.0/group__gazebo__sensors.html)



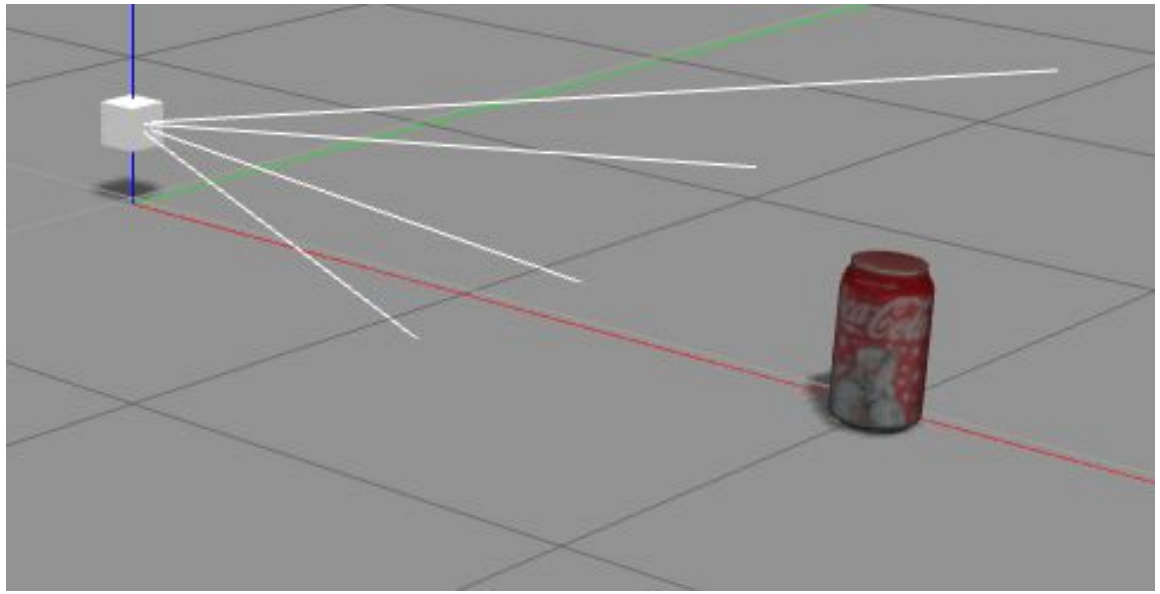
SENSORS IN GAZEBO

- Gazebo(gazebo_models repo) provides some SDF models of sensors that include geometric descriptions and sensor features like camera.
- Gazebo provides a set of sensor classes, like CameraSensor, that define the behavior of the sensors. All of them inherit from class physics::Sensor.
- A set of plugins, like the camera plugin, that interface the corresponding sensor classes.
- ROS sensor plugins are available to wrap the Gazebo sensor plugins.
- Plugins can be added to SDF sensor models or to sensor models defined using URDF(Unified Robotic Description Format).



GAZEBO : CAMERA SENSOR

- Camera ROS Plugin : ROS interface for simulating cameras by publishing the CameraInfo and Image ROS messages as described in *sensor_msgs*.
- SDF Model : The gazebo_sensors_tutorial package contains a models folder that includes this SDF camera sensor model modified to include the plugin XML code
- URDF Model : URDF robot models may contain description of sensors with the associated plugins
- Plugin Implementation : class GazeboRosCamera implements the camera plugin



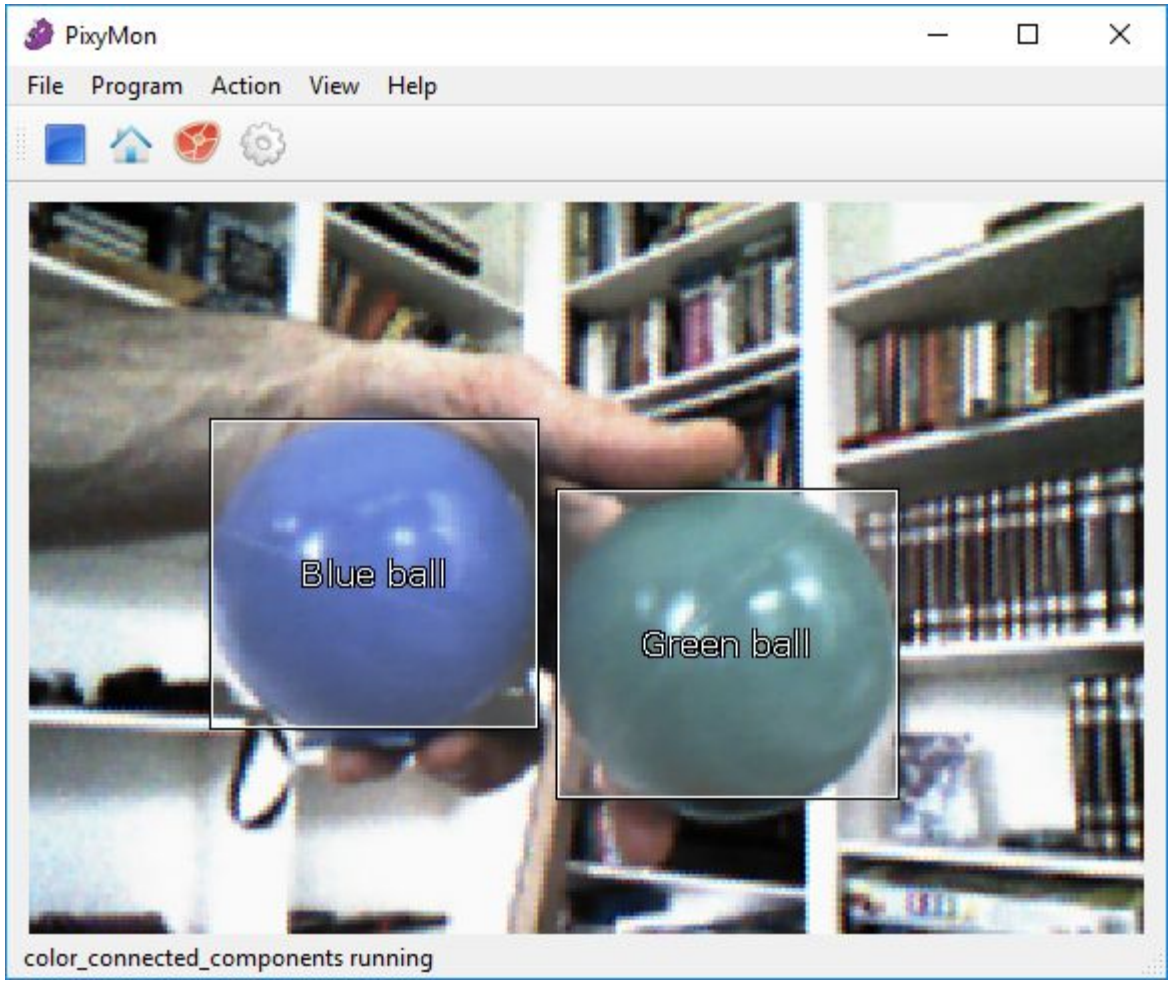
VISION AS A SENSOR

If one wants a robot to perform a task such as picking up an object, chasing a ball, locating a charging station, etc., and one want a single sensor to help accomplish all of these tasks, then **vision** is your sensor.

Vision (image) sensors are useful because they are so flexible. With the right algorithm, an image sensor can sense or detect practically anything.

But there are two drawbacks with image sensors:

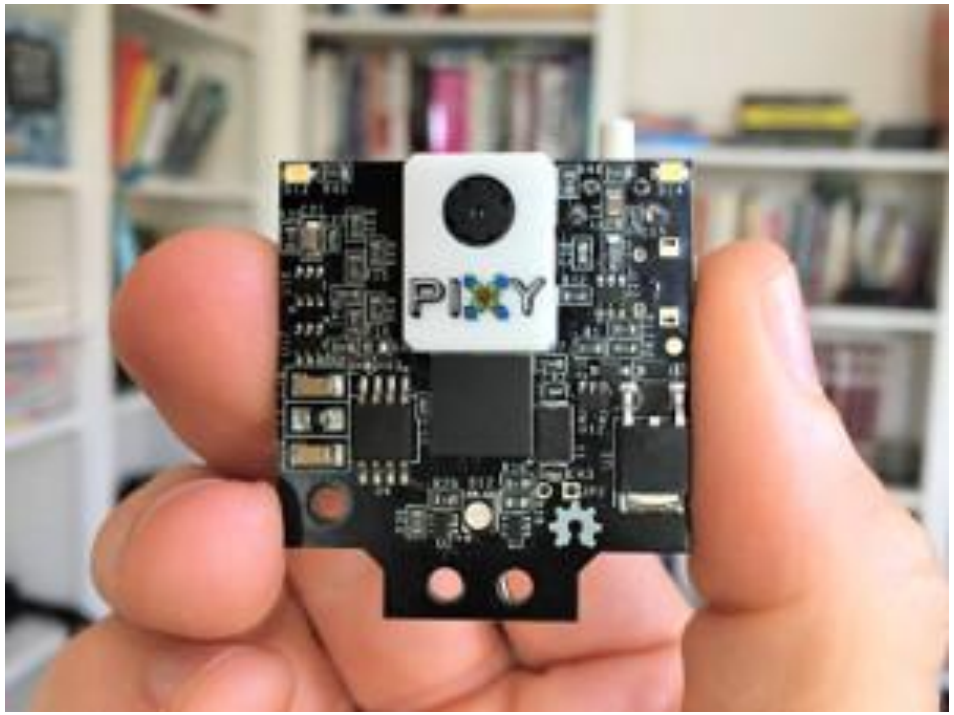
1) they output lots of data, dozens of megabytes per second
2) processing this amount of data can overwhelm many processors.



PIXY 2 CAMERA

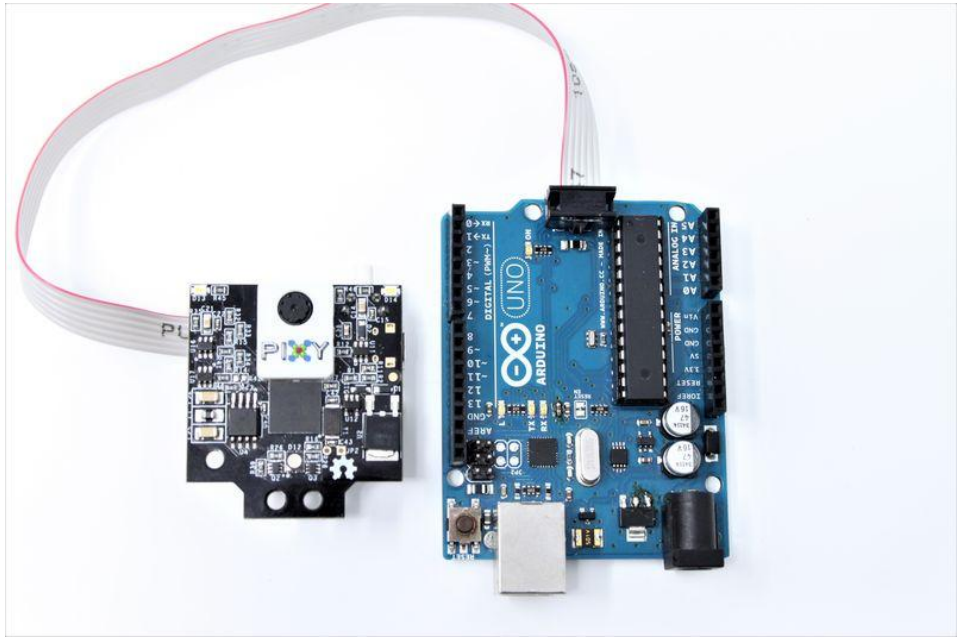
MACHINE VISION MADE EASY

- One of the best options to made Robot Vision easier
- Small, fast, easy-to-use, low-cost, readily-available vision system
- Learn to detect objects that you teach it
- Detect and track lines for use with line-following robots
- Detect intersections and “road signs” as well.
- Frame Rate 60 frames-per-second, so robot can be fast too
- Integrated light source
- C/C++ and Python are supported
- Communicates via one of several interfaces: SPI, I2C, UART, USB or analog/digital output
- Configuration utility runs on Windows, MacOS and Linux
- All software/firmware is open-source GNU-licensed



PIXY 2 CAMERA

- Can be plugged directly into an Arduino and a USB cable to plug into a Raspberry
- Have powerful dedicated processor with the image sensor. It processes images from the image sensor and only sends the useful information to the microcontroller.
- Multiple Pixys can be hooked up to microcontroller — for example, a robot with 4 Pixy2's and omnidirectional sensing. Or use Pixy2 without a microcontroller and use the digital or analog outputs to trigger events, switches, servos, etc.



MAJOR ADVANTAGES OF USING PIXY2 CAMERA

- **Teach it the objects you're interested in**

Pixy2 is unique because you can physically teach it what you are interested in sensing

- **Pixy2 "tracks" each object it detects**

Once Pixy2 detects a new object, it will add it to a table of objects that it is currently tracking and assign it a tracking index. It will then attempt to find the object (and every object in the table) in the next frame by finding its best match. Each tracked object receives an index between 0 and 255 that it will keep until it either leaves Pixy2's field-of-view, or Pixy2 can no longer find the object in subsequent frames (because of occlusion, lack of lighting, etc.)

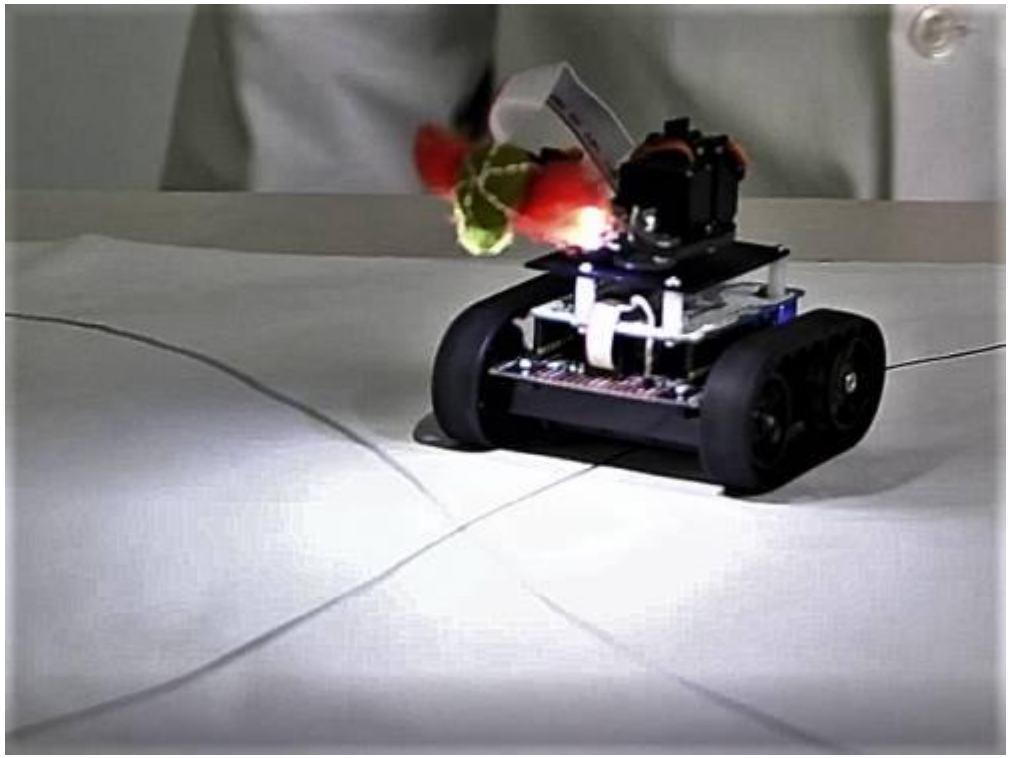
Tracking is useful when you want your program to keep tabs on a certain instance of an object, even though there may be several other similar objects in the frame.

ADVANTAGES OF USING PIXY2 CAMERA

- **Line tracking for line-following**

Pixy2 has added the ability to detect and track lines. Line-following is a popular robotics demo/application because it is relatively simple to implement and gives a robot simple navigation abilities. Most line-following robots use discrete photosensors to distinguish between the line and the background.

Pixy2's camera frames takes in information about the line being followed, its direction, other lines, and any intersections that these lines may form. Pixy2's algorithms take care of the rest. Pixy2 can also read simple barcodes, which can inform your robot what it should do – turn left, turn right, slow down, etc.

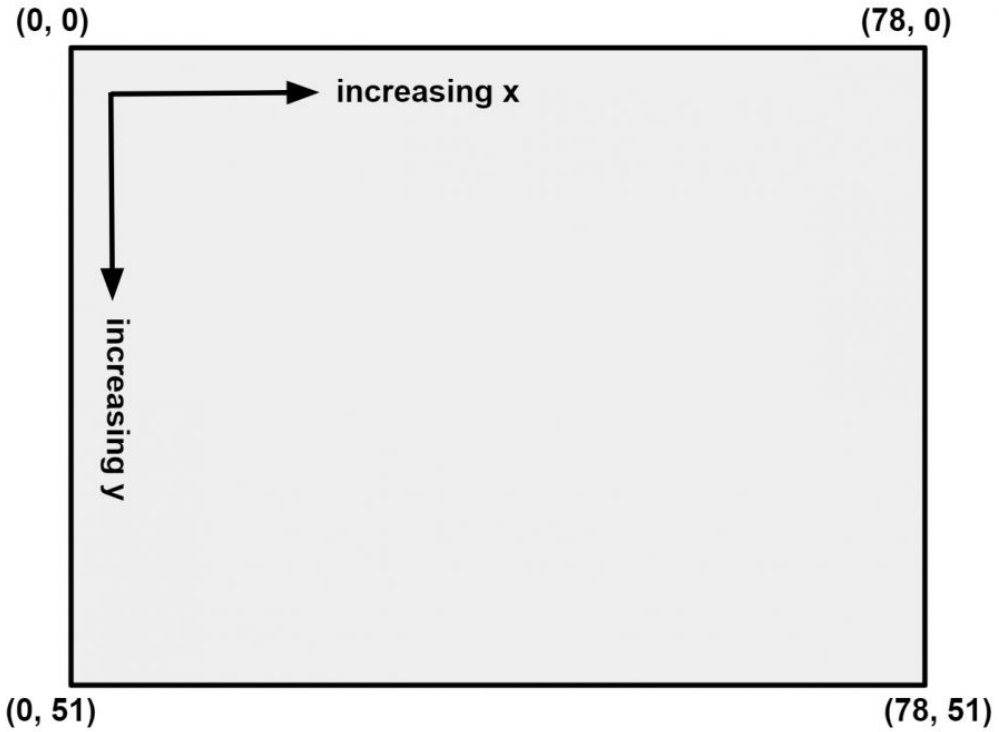


MAJOR ADVANTAGES OF USING PIXY2 CAMERA

Pixy2 tries to send only the most relevant information to your program:

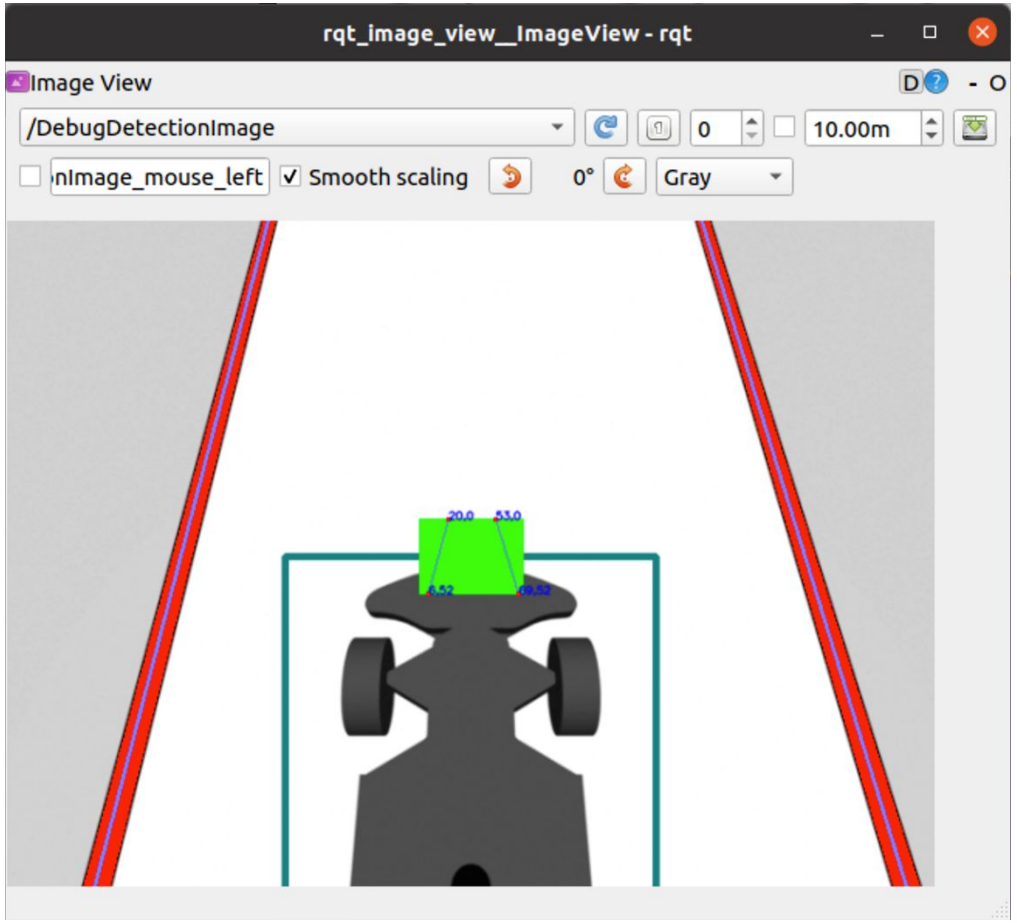
- It finds the best Vector candidate and begins tracking it from frame to frame. The Vector is often the only feature Pixy2 provides to your program, and it's the only feature your program typically needs when following a line.
- It detects intersections and reports them after they have met the filtering constraint.
- It “executes” branches by assigning the Vector to the branch line that your program chooses.

But one may want to use Pixy2's line tracking algorithms in a different way that requires that Pixy2 share more of the information that it is gathering. If so, Pixy2 can provide program with all lines, intersections and other necessary information that it detects, regardless of filtering constraints or the state of the Vector.
Pixy2 will still track each line, intersection and barcode from frame to frame and provide your program with this information, but your program can choose to use the information or ignore it. In other words, your program will be unconstrained regarding the information it receives from Pixy2.



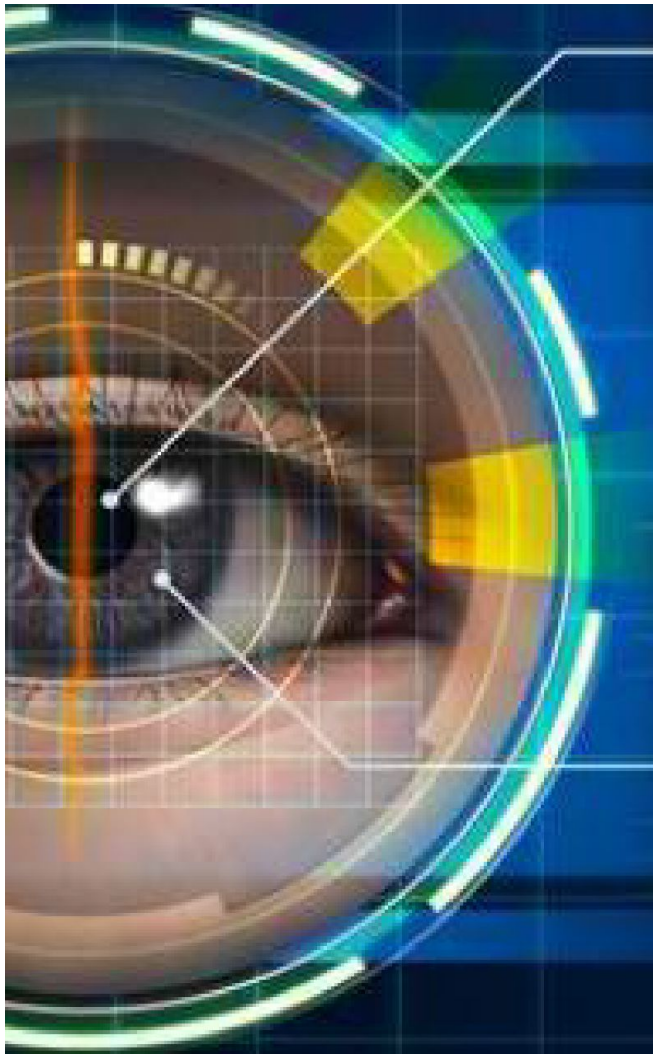
PIXY CAMERA INTEGRATED IN SIMULATION

- Pixy Camera has been simulated in Gazebo Environment
- Detects lines and outputs vector data just like the real Pixy camera.
- The source code for the simulated Pixy camera uses OpenCV to fit vectors to detected lines in simulation



WHAT IS COMPUTER VISION

Acquiring, processing, analyzing and **understanding** digital images, and extraction of **high-dimensional** data ***from the real world*** in order to produce numerical or symbolic information, in the forms of decisions



OPEN CV

OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

OpenCV Algorithm Modules Overview

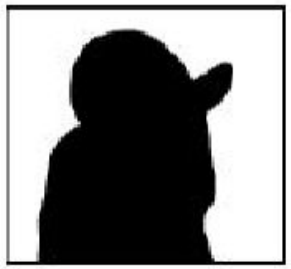
HighGUI:
I/O, Interface

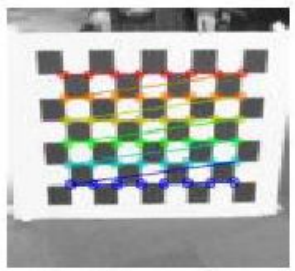

Image Processing

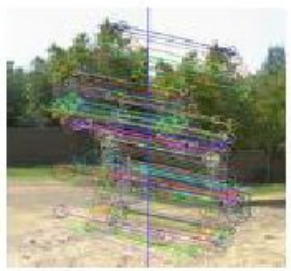

Transforms

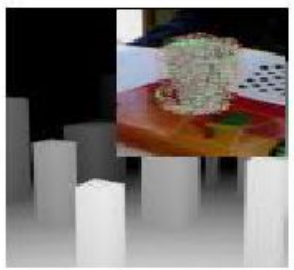

Fitting

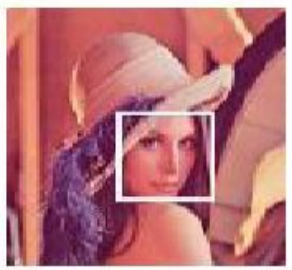

Optical Flow
Tracking


Segmentation


Calibration


Features
VSLAM


Depth, Pose
Normals, Planes,
3D Features

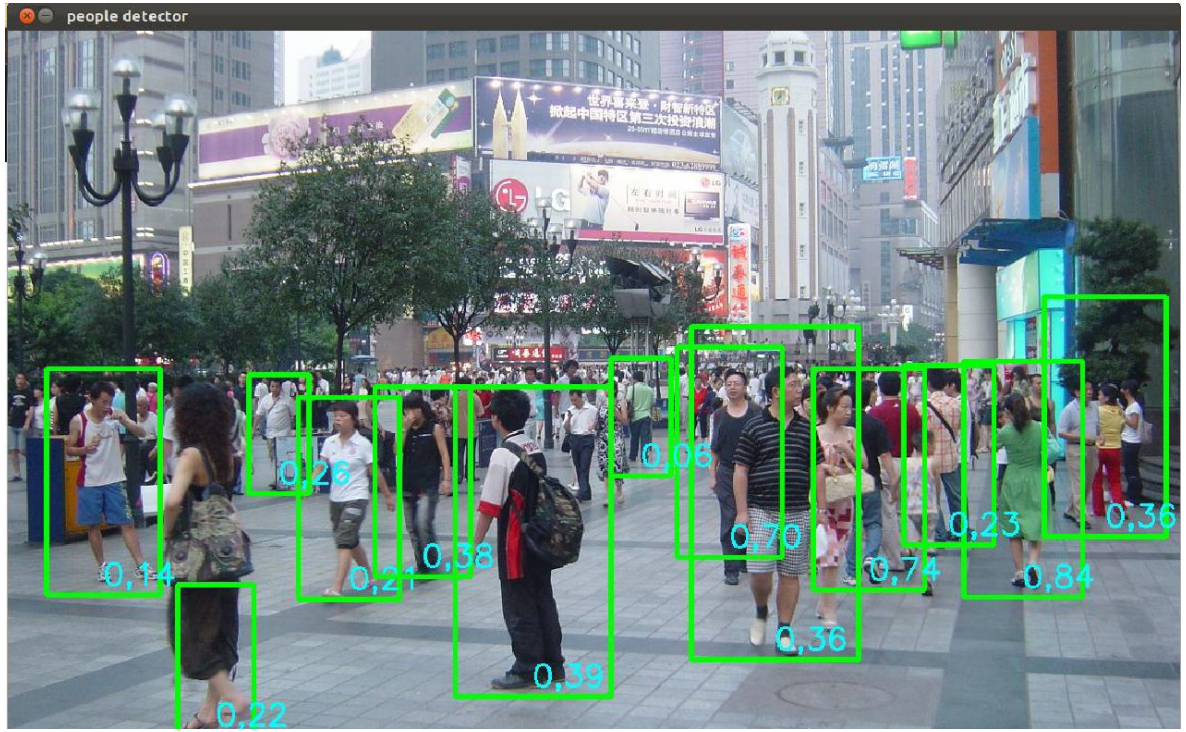

Object recognition
Machine learning

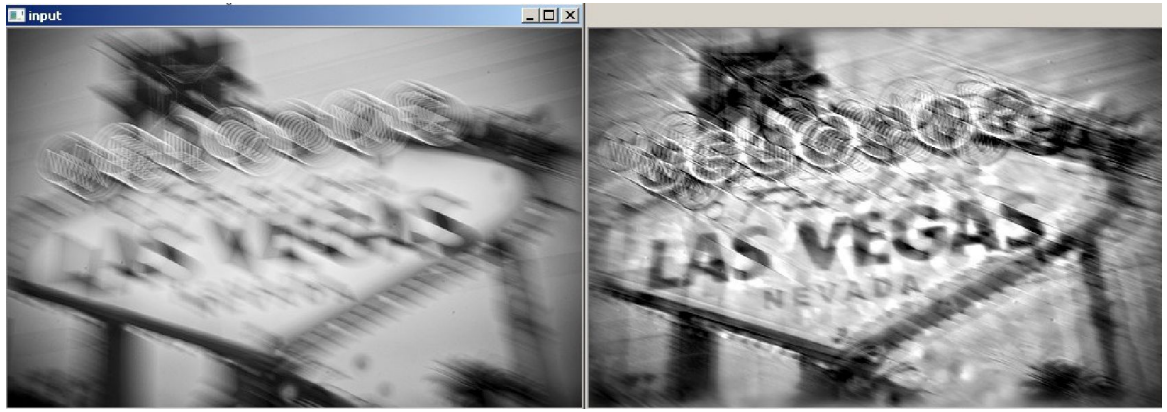

Computational
Photography

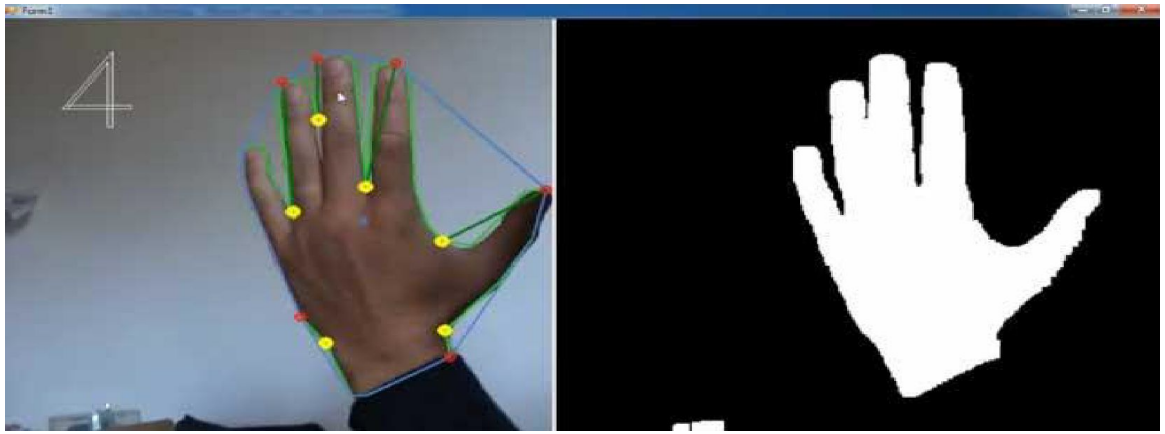
CORE:
Data structures, Matrix math, Exceptions etc

BASIC REQUIREMENTS FOR IMAGE PROCESSING USING OPENCV

- Have basic programming skills.
- Basic knowledge using C++, python or another programming language
- Image Processing and Linear Algebra



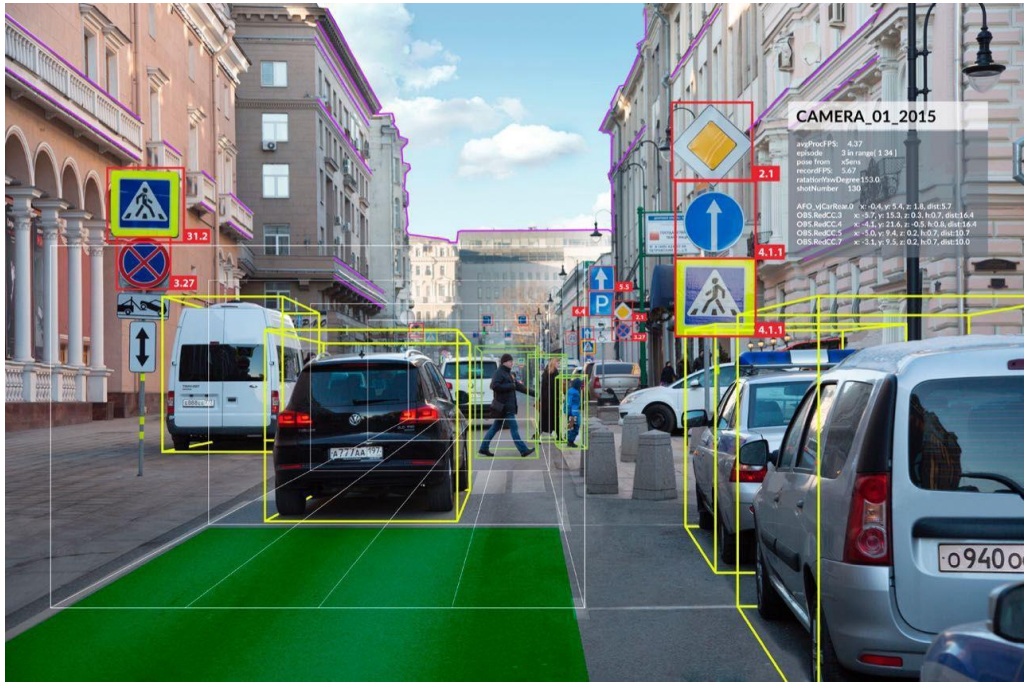




WHY OPEN-CV?

- More than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms.

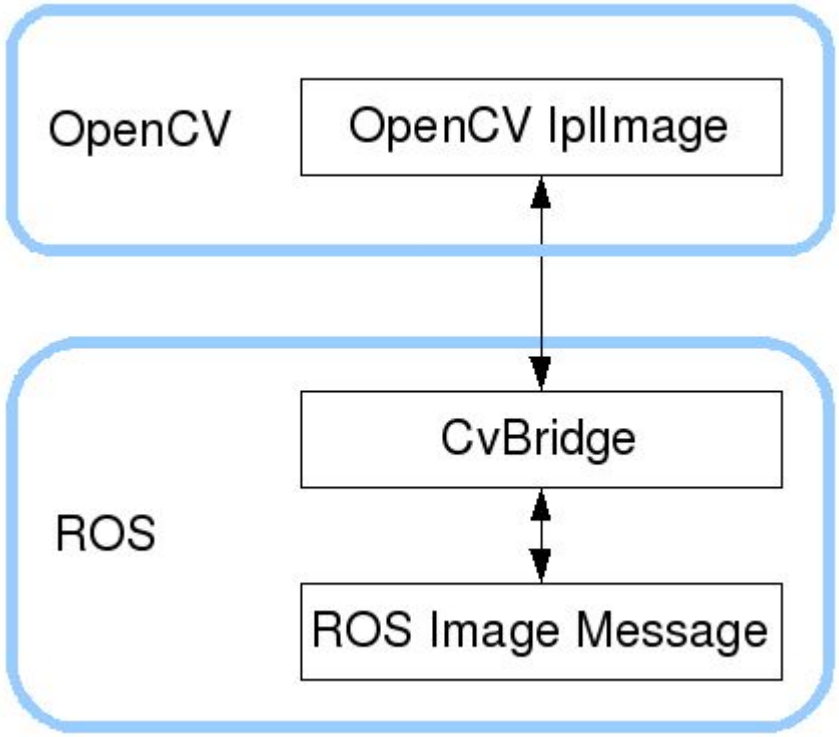
- These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.



VISION_OPENCV LIBRARY

- Packages for interfacing ROS with OpenCV.
- The vision_opencv stack provides packaging of the popular OpenCV library for ROS.
- For OpenCV vision_opencv provides several packages:

cv_bridge: Bridge between ROS messages and OpenCV.
image_geometry: Collection of methods for dealing with image and pixel geometry



QT VIEWER

Qt is a widget toolkit for creating graphical user interfaces as well as cross-platform applications that run on various software and hardware platforms such as Linux, Windows, macOS, Android or embedded systems with little or no change in the underlying codebase while still being a native application with native capabilities and speed.

Qt Framework

Rely on the tools you know and combine them with a powerful framework.

Reusable Components

Save time and effort by reusing powerful QML components in the same or different projects.

Code Auto Generation

Easily generate code from your design and save time.

Multidimensional Design Capabilities

Take your UI designs to a whole new dimension with native support for 2D and 3D graphics that you can mix and animate seamlessly.

Custom Styles

Easy options to customize your controls for brand consistency.

Ready-Made Controls

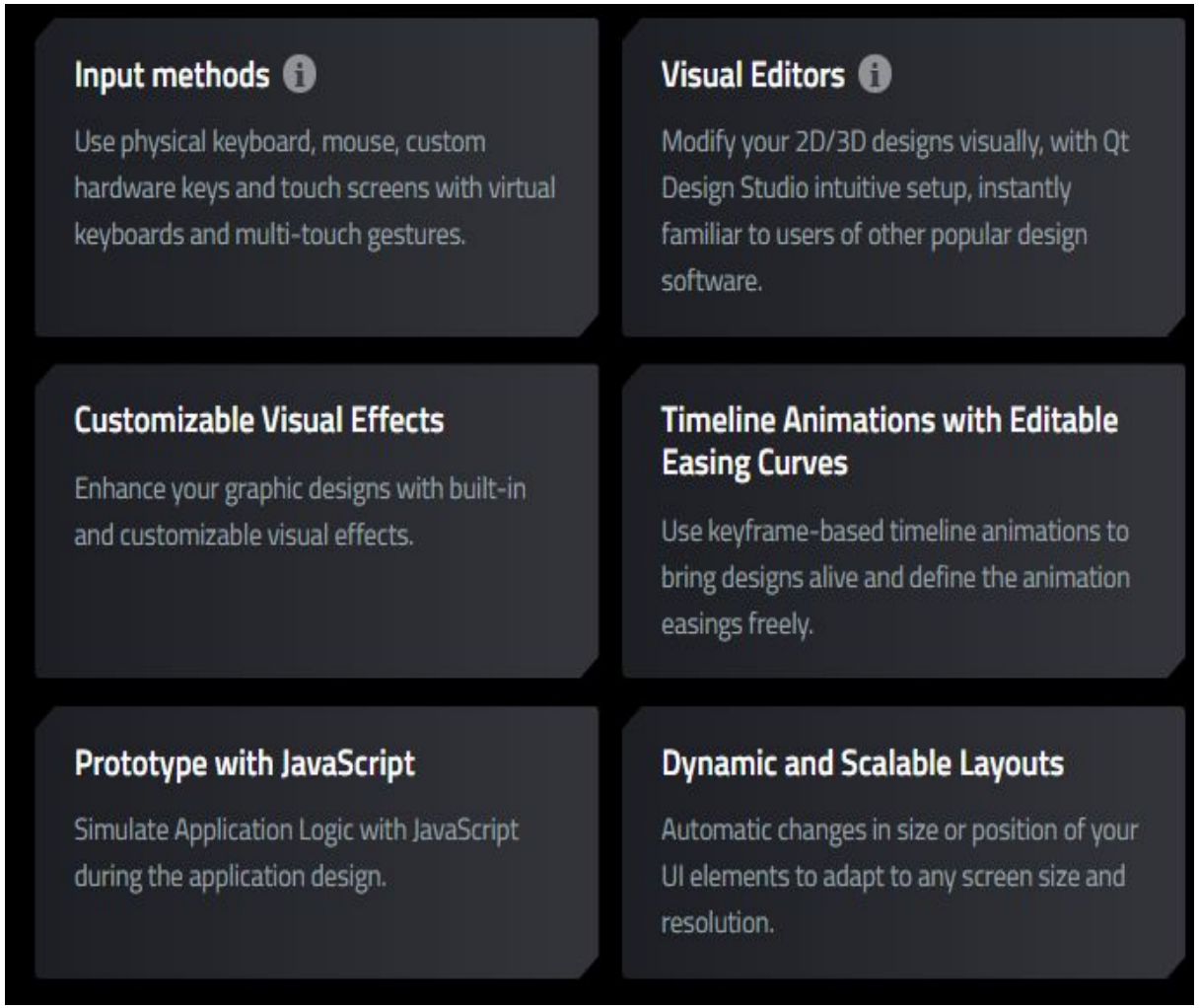
Use our ready-made, native-looking UI elements instead of creating them from scratch

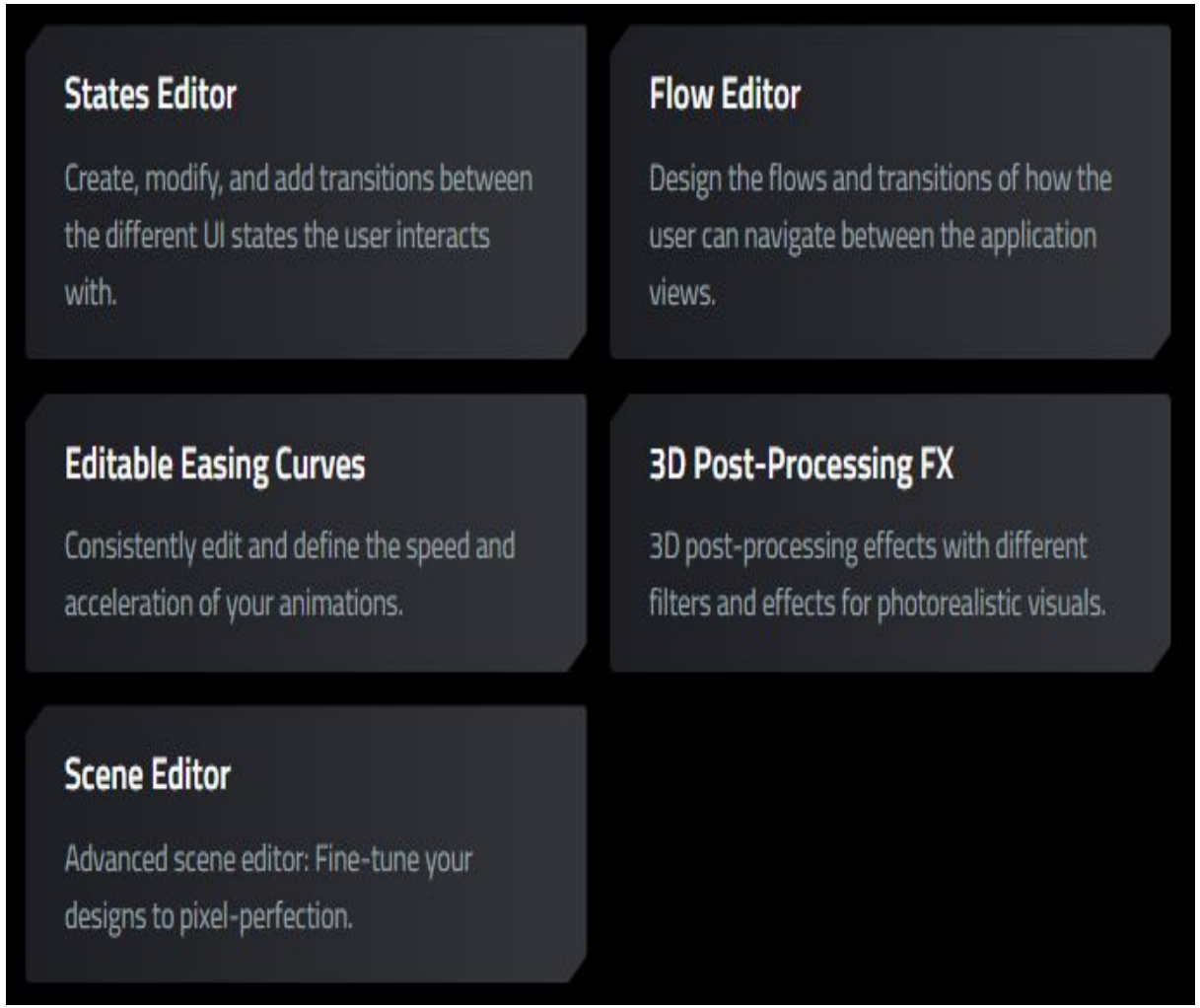
Fast Iterations on Real Hardware

Turn your designs quickly into interactive prototypes to validate and iterate the UI on target hardware. The developer can use the prototype as a starting point for implementation.

Optimized Workflow

Less need to write and update heavy specifications as the designs are immediately usable by developers.





SECURE CONNECTIONS
FOR A SMARTER WORLD

Thank you