

A COMPREHENSIVE STUDY ON INDUSTRIAL VISION ROBOT WITH PIXY CAMERA

DUTTA SIVA NANDINI

Scholar

Electronics and Communication
Engineering

Krishna Chaitanya Institute of Technology
& Sciences, A.P

MR. K. SATHISH

Assist. Professor

Electronics and Communication
Engineering

Krishna Chaitanya Institute of Technology
& Sciences, A.P

ABSTRACT:

The capacity to perceive articles has been an objective of PC researchers and AI specialists for a considerable length of time. In the past this has required enormous PCs running refined programming which has kept the advances included bound to labs and research divisions with huge spending plans. Blaze forward to today, object acknowledgment has turned out to be standard. Item acknowledgment and PC vision innovation is presently accessible for experimenters too, with a few packs and cameras with different abilities accessible. The Pixy is a picture sensor intended for item acknowledgment, line following and basic scanner tag perusing. Mechanical robots are not human, they are machines. They are programmable controller gadgets which can move devices or parts by means of a set grouping of movements. Also, they can be reprogrammable, that is, the robot's activity can be adjusted by changing the control settings without supplanting the equipment. They include a few attributes of customary machines in like manner as qualities of machine administrators. For an administrator, it is anything but difficult to be instructed to complete another undertaking. However, for a machine, an errand can be rehashed for delayed occasions with extraordinary exactness. This task centered at building up a Robot Vision framework utilizing a blend of ease camera equipment and PC calculations to empower robots to process visual information from the world. The stereo vision calculation which comprises of two cameras, and the created application can figure a 3D position from s 2D identified article. Moreover, the discovery calculation dependent on shading contrasts was utilized by the cameras which empower 2D item following and yielded information directions of the

article being distinguished. At that point, the 3D item position is created through the determined 2D object information facilitates, which made prepared for robot educating.

Key words: Robot Vision, Pixy camera, computer algorithms

1.0 INTRODUCTION TO ROBOTICS

Robotics is a branch of engineering and science that includes electronics engineering, mechanical engineering and computer science and so on. This branch deals with the design, construction, use to control robots, sensory feedback and information processing. These are some technologies which will replace humans and human activities in coming years. These robots are designed to be used for any purpose but these are using in sensitive environments like bomb detection, deactivation of various bombs etc. Robots can take any form but many of them have given the human appearance. The robots which have taken the form of human appearance may likely to have the walk like humans, speech, cognition and most importantly all the things a human can do. Most of the robots of today are inspired by nature and are known as bio-inspired robots. Robotics is that branch of engineering that deals with conception, design, operation, and manufacturing of robots. There was an author named Issac Asimov, he said that he was the first

person to give robotics name in a short story composed in 1940's. In that story, Issac suggested three principles about how to guide these types of robotic machines. Later on, these three principles were given the name of Issac's three laws of Robotics. These three laws state that:

- Robots will never harm human beings.
- Robots will follow instructions given by humans with breaking law one.
- Robots will protect themselves without breaking other rules

Knowing the type of camera to used:

Three-dimensional, are the real-world scenes a camera faces today. In most cases, when exposed to different depths, objects would appear to each other in form of two-dimensional mapped scene of the sensor of the camera. In figure 1, is a presentation of a figure obtained from Middlebury picture dataset as illustrated. The picture succinctly provides details that the motorbike at a view from the background of the photo is about two meters closer to the camera when compared to the shelf at the background. Watch point 1 and 2 as marked in the figure. Point 1 as illustrated in the red box place in the background of the figure shows to be adjacent to the second fork of the motorcycle in the image, even with two-meter distance from the camera. Owing to the power of perspective

2.0 PIXY - INTRODUCTION

What is Pixy Exactly?

- Small, fast, easy-to-use, low-cost, readily-available vision system.
- Learns to detect objects that you teach it.
- Outputs what it detects 50 times per second.
- Connects to Arduino Uno with

included cable. Also works with Raspberry Pi, BeagleBone and similar controllers.

- All libraries for Arduino are provided.
- 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.
- All hardware documentation including schematics, bill of materials, PCB layout, etc. are provided.



Figure 1. Pixy

Pin Configuration of Pixy

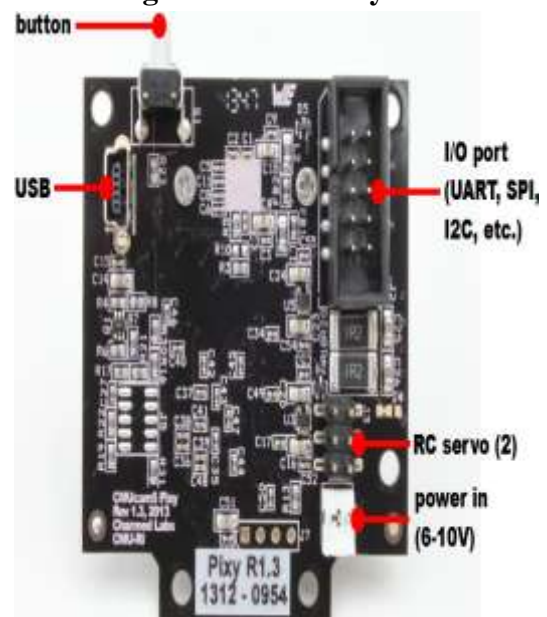


Figure 1. Pixy Pin Descriptions

- Button which is used to set the

signature (setting colour code) inside the pixy.

- There will be Mini USB to connect the pixy to pixymon software.
- I/O port is used for communication using various supported protocols like UART, SPI, I2C, etc.

And finally there will be a power in (6 – 10V) jack to power up the pixy

3.0 METHODOLOGY

Technical Specifications

Table 1 Technical Specifications of Arduino Uno

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm

Weight	25 g
--------	------

Teaching Pixy an Object

Showing Pixy an item is overly simple, yet first how about we talk about which articles will function admirably with Pixy. Imp utilizes a tint based shading separating calculation to identify objects. Since Pixy uses shade (shading), the item needs a particular tone. Here are a few items that are great since they have great, particular tints. Here are some awful items in light of the fact that either there is no shade (dark, white or dim) or the tone isn't unmistakable.

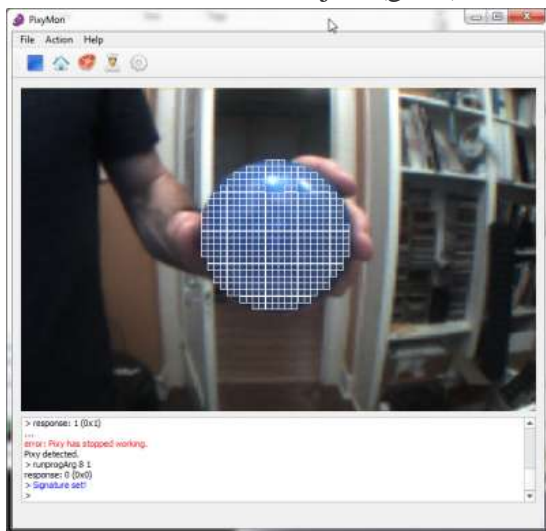
Remembering these rules, pick an item to instruct Pixy. (To begin with, apply capacity to Pixy by means of battery or USB link on the off chance that you haven't as of now. When you catalyst Pixy, it will experience a progression of LED flashes. Trust that the LED will mood killer before showing Pixy an article.) Now, hold down the catch over Pixy. After around 1 second, the LED will turn on - first white, at that point red, at that point different hues - yet when it turns red, discharge the catch.

When you discharge the catch, Pixy will enter what's classified "light pipe" mode, where the LED shading is the shade of the item that Pixy has "bolted" onto. Sprite will secure onto questions in the focal point of its video outline, so hold the article straightforwardly before Pixy, somewhere in the range of 6 and 20 creeps from the focal point.

Fairy utilizes a district developing calculation to attempt to figure out which pixels are a piece of your item and which

pixels are a piece of the foundation. Utilizing these pixels, Pixy will attempt to make a measurable model of your article so it can identify it dependably under various lighting conditions. Utilize the LED shading as criticism to decide whether Pixy has a decent lock on the item, and utilize the accompanying rules to pass judgment:

1. When Pixy has locked onto your object, the LED color should match the color of your object.
2. The brighter the LED, the better the lock. Move the object a little and see if you can maximize the LED brightness.
3. Look at the region-growing grid in the PixyMon window and see how the size of the grid corresponds to the LED color and brightness. A grid that covers more of your object is better than a grid that covers only some of your object and results in a brighter LED. The picture below shows a grid that covers most of the object (good).



Learning when Pixy has a decent lock by looking at the LED might take a little practice, but once you get the feel of it, you can teach Pixy new objects reliably **without needing to use PixyMon**. When you are satisfied, press and release the button on Pixy, like you'd click your mouse. The LED will flash a couple times

indicating that Pixy has now “learned” your object. It will now start tracking your object

Software Implementation

The Code which is dumped in Arduino is given below:

```
#include <SPI.h>
#include <Pixy.h>
#include <Wire.h>
int oldSignature;
int data[5];
Pixy pixy;
int k;
void sendDataToRPi(){
  for(k=0;k<5;k++){
    //Serial.println(data[k]);
    //Serial.println(data[k]);
    Wire.write(data[k]);
    //Serial.println("block k");
    //Serial.println(data[k]);
    //Serial.println("Data sent");
    // delay(1000);
  }
  Wire.write(oldSignature);
}
int TrackBlock(int blockCount){
  int trackedBlock = 0;
  long maxSize = 0;
  Serial.print("blocks =");
  Serial.println(blockCount);
  for (int i = 0; i < blockCount; i++){
    if ((oldSignature == 0) ||
    (pixy.blocks[i].signature ==
    oldSignature)){
      long newSize = pixy.blocks[i].height *
      pixy.blocks[i].width;
      if (newSize > maxSize){
        trackedBlock = i;
        maxSize = newSize;
      }
    }
  }
  data[0] = pixy.blocks[trackedBlock].x;
  data[1] = pixy.blocks[trackedBlock].y;
```



```

data[2] =
pixy.blocks[trackedBlock].width;
data[3] =
pixy.blocks[trackedBlock].height;
data[4] =
pixy.blocks[trackedBlock].angle;
int k;
Wire.onRequest(sendDatatoRPi);
for(k=0;k<5;k++){
  Serial.println(data[k]);
}
oldSignature =
pixy.blocks[trackedBlock].signature;
Serial.println(oldSignature);
return trackedBlock;
}
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial.print("Starting...\n");
  pixy.init();
  Wire.begin(8);
}
void loop() {
  // put your main code here, to run
  repeatedly:
  uint16_t blocks;
  blocks = pixy.getBlocks();
  if (blocks){
  int trackedBlock = TrackBlock(blocks);
  }
}

```

Output

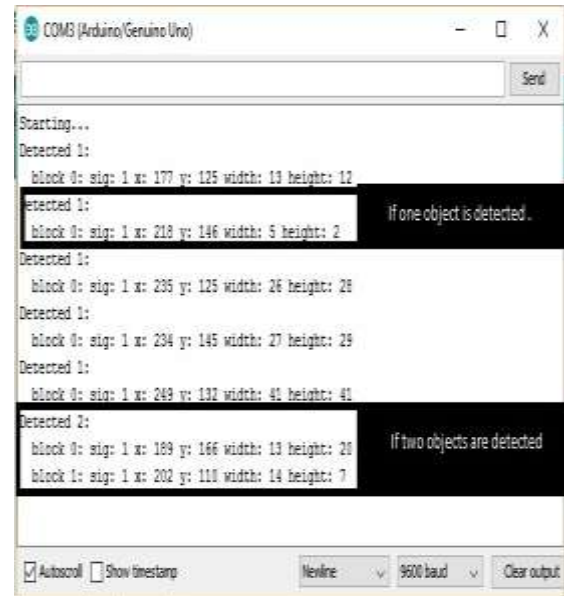


Figure 3 Output of Arduino Uno

Triangle similarity is the idea behind the situation. (Probably more than you ever wanted to know - <http://www.mathopenref.com/similartriangles.html>). The idea is a triangle are similar if they have the same shape but not necessarily the same size.

As we can see in the output, we have only two parameters i.e., height and width. but it is important to know distance between pixy bot and object. So we use a method similar triangles.

Let's say we have a object with a known width (W). We place the object a known distance (D), and take a picture of it and measure the apparent width in pixels (P). We come up with this formula for perceived focal length (F) of our camera.

$$F = (P \times D) / W$$

We can now use this formula and will have two known values. (F and W). We can now figure out a unknown distance.

$D = (W \times F) / P$ The camera will read a different apparent width in pixels if the camera or the object moves. And a rough distance can be found.

We took this idea one more step and applied it to the height as well. The formula is pretty much the same,

substituting width with the height of the object, and substituting the apparent height in pixels for P.

We then take the two unknown distances add them together, and divide them by two.

There are things that can negatively effect these. Background noise, not being able to see the whole object.

Either too much light or not enough light seems to play a big role in how well this works as well. Setting up the color code

4.0 RESULTS AND DISCUSSIONS

All the hardware components discussed earlier have been put together into a robot like structure. This structure is built in a way such that the Raspberry pi and Brick pi are put beside the Arduino IDE board. On the top shelf we have the Pixy camera mounted and the 9.6v supply battery box is place behind it. The rotating action inbuilt into the pixy cam is disabled as we are using lego motors to move the robot with attached wheels.

This is how the final bot looks likes:

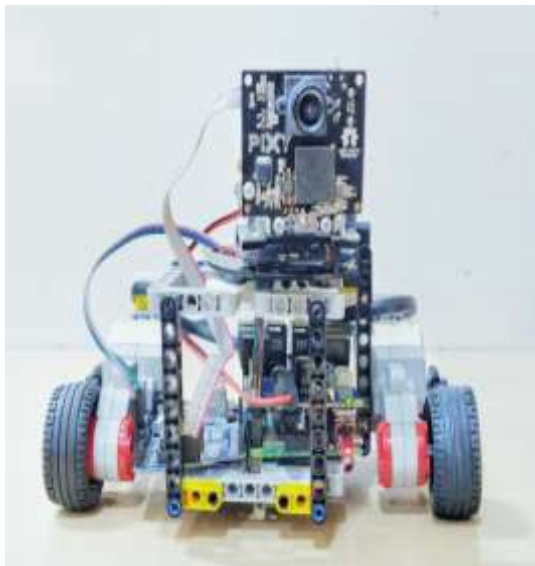


Figure 5 Final Pixy Bot (Front View)

On completing the connections a signature is being set using display via the cable which provides connection between a pixy cam and a computer using the pixymon software as the interface. The

Arduino code is uploaded after compiling, which sends the information regarding the position of the object to the raspberry pi microcontroller thru I2C connection. Using the putty software the raspberry pi is controlled wirelessly hence cutting down the use of wired connections. The final structure is not connected to any display unit or any other external hardware.



Figure 6 Final Pixy Bot (Side View)

We use the Brick pi driver in order to control the motors installed in the robot. When put in action, this robot is capable of following an object which is pre set as a signature in the pixymon software by its differentiable colour code. As the object moves further away the robot follows it hence, helping us in keeping a track of it

5.0 CONCLUSION

Pixy following an article is a propelled venture that utilizes two microcontrollers and a versatile bot developed utilizing EV3 pack. It has a wide scope of utilizations. It utilizes a picture sensor to distinguish the item which has been prepared as of now and tails it to a most extreme degree as conceivable continuously. This mechanized mechanical structure may

diminishes the manual blunders in item acknowledgment field and may likewise convey powerful outcome. Article acknowledgment/identification may have key job in the modern computerization and some different spaces of society. Be that as it may, there is negligible possibility of not distinguishing an article in certain circumstances like dull lighting condition and quicker developments of items. A few pieces of small following frameworks strategy, there are a few calculations and procedure ways to deal with this, it relies upon the kind of framework one is actualizing and the writing that can be found on the web. Additionally, Track It Yourself, the proposed method utilized for 3D following framework was clarified. portrays the general review of PC vision, machine vision and their applications additionally the money related avocation machine vision and camera adjustment will be examined.

REFERENCES:

1. Andermann ML, Gilfoxy NB, Goldey GJ, Sachdev RN, Wölfel M, McCormick DA, Reid RC, Levene MJ (2013) Chronic cellular imaging of entire cortical columns in awake mice using microprisms. *Neuron* 80:900–913.
2. CrossRef Medline Arkley K, Grant RA, Mitchinson B, Prescott TJ (2014) Strategy change in vibrissal active sensing during rat locomotion. *Curr Biol* 24:1507–1512.
3. CrossRef Medline Berg RW, Kleinfeld D (2003) Rhythmic whisking by rat: retraction as well as protraction of the vibrissae is under active muscular control. *J Neurophysiol* 89:104–117.
4. CrossRef Medline Bermejo R, Harvey M, Gao P, Zeigler HP (1996) Conditioned whisking in the rat. *Somatosens Mot Res* 13:225–233.
5. Medline Bobrov E, Wolfe J, Rao RP, Brecht M (2014) The representation of social facial touch in rat barrel cortex. *Curr Biol* 24:109–115.
6. CrossRef Medline Carvell GE, Simons DJ (1990) Biometric analyses of vibrissal tactile discrimination in the rat. *J Neurosci* 10:2638–2648.
7. Medline Carvell GE, Simons DJ, Lichtenstein SH, Bryant P (1991) Electromyographic activity of mystacial pad musculature during whisking behavior in the rat. *Somatosens Mot Res* 8:159–164.
8. Medline Cheung E, Chatterjee D, Gerlai R (2014) Subcutaneous dye injection for marking and identification of individual adult zebrafish (*Danio rerio*) in behavioral studies. *Behav Res Methods* 46:619–624.
9. CrossRef Medline Clack NG, O'Connor DH, Huber D, Petreanu L, Hires A, Peron S, Svoboda K, Myers EW (2012) Automated tracking of whiskers in videos of head fixed rodents. *PLoS Comput Biol* 8:e1002591.
10. Vuong C, Popeney B, Arisaka K, Mehta MR (2013) Multisensory control of multimodal behavior: do the legs know what the tongue is doing? *PLoS One* 8:e80465. CrossRef Medline