

**Tecnológico de Monterrey
Campus Querétaro**

Second Partial Evaluation: Robotic Arm

Developed by:

**José Ramón Romero Chávez (A01700318)
Jorge Adán Campos Amador (A01272415)**

Robotic Vision

Dr. Rick Leigh Swenson Durie

05/04/2019

Index

Index	2
Introduction	3
Results	6
Evidence	6
Observations	6
Conclusions	7
References	8
Sherlock and Scripting Appendix	9

Introduction

Machine vision is one of the key technologies in manufacturing because of increasing demands on the documentation of quality and the traceability of products. It is concerned with engineering systems, such as machines or production lines that can perform quality inspections in order to remove defects or controlling other aspects.[1]

About Sherlock

Sherlock is an advanced machine vision software interface that can be applied to a wide variety of automated inspection applications. It offers maximum design flexibility and provides a rich suite of proven tools and capabilities that have been deployed in thousands of installations worldwide.[2]

About the Robot Arm

The AL5A robotic arm delivers fast, accurate, and repeatable movement. The robot features: base rotation, single plane shoulder, elbow, wrist motion, a functional gripper, and optional wrist rotate all this controlled by SSC-32 Servo Controller via serial communication to the computer and the camera.[3][4]

Development:

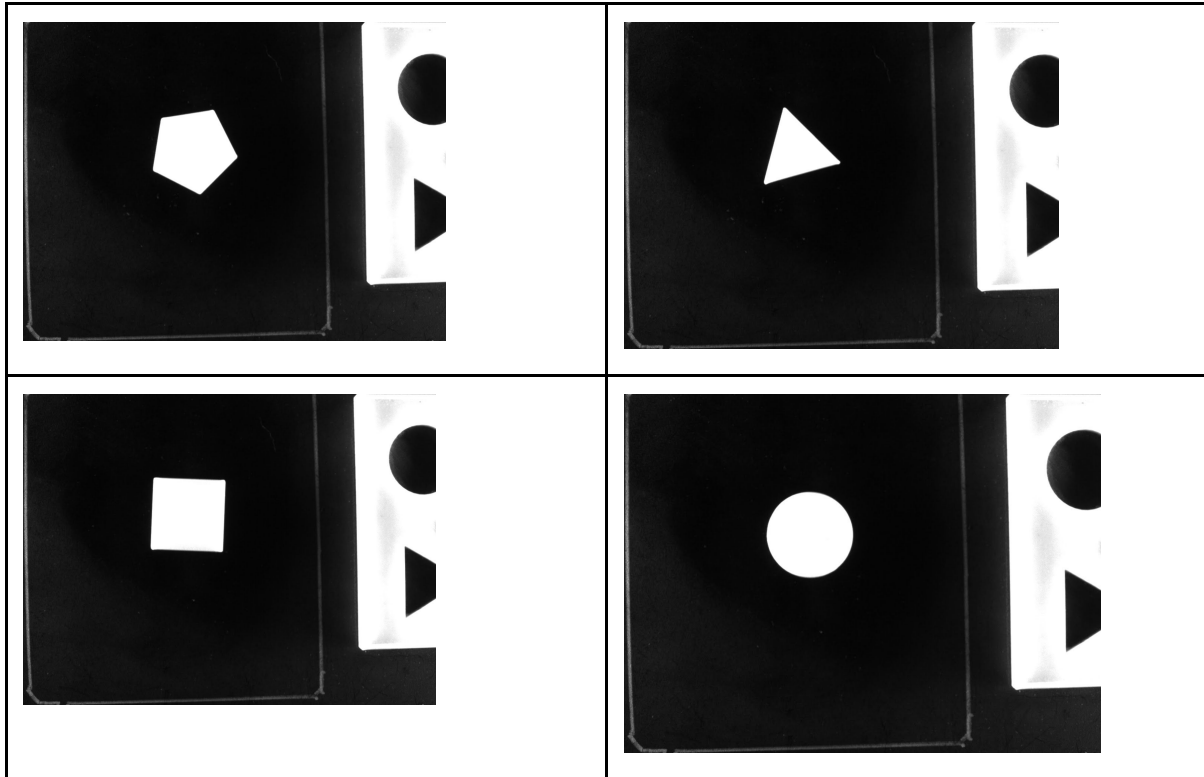
The objective of this practice is to see a practical approach of the usage and advantages of machine vision.

The implementation consists of these main steps:

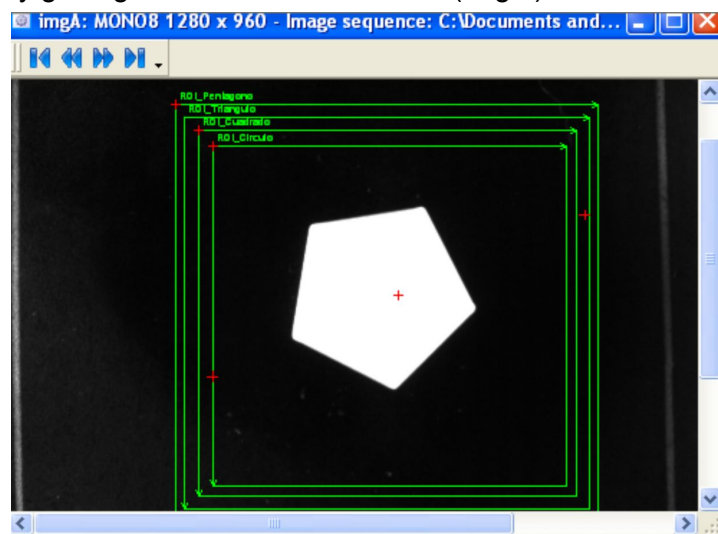
- A camera which detects the pattern of a piece.
- The arm which takes the piece from the start point.
- The manipulation of the piece
- The setting of the piece on its corresponding place

Images Data Set:

In order to perform the proper recognition the system was provided with the following images (Training Set):



By using the Edge-Search Detection Algorithm over the Region of interest, it was possible to identify each of the elements and characteristics in order to try and compare by using the camera, specially by getting the difference in rotation(angle).



The angle was generated by using the following regression calculated by the PWM from the robotic Arm:

45° -> 2000
0 -> 1500
-45 -> 1000

```
function Rotar(Angulo){  
  if (Angulo<0){  
    Angulo=Angulo*(-1);  
  }  
  var Y=Math.floor( 11.11*(Angulo)+1000);  
  var comando= "#4 P"+Y.toString() +" T3000\\x0D";  
  return comando;  
}
```

Results

The robotic arm was capable to pick, rotate and put on the correct place each of the pieces, the movements were slow in order to give all the time that the action needed to take to the place each of the pieces. You can find a clear evidence of the result in the evidence section above.

Evidence

https://drive.google.com/drive/folders/1SqmNE_Vr436YISygd_9JgS2HZvTkvUBG?usp=sharing

Observations

- It was useful to see this approach, but dealing with a single arm for multiple teams can lead on some changes an unexpected situations as the modifications or damage to the arm robot.
- This team tried an approach using Inverse Kinematics (check the “others” folder), because of time it was impossible to try it.
- Also serial communication using Python was tested with interesting results, it is important to consider the difference between UNIX based systems and MS, using OpenCV can generate similar or even better results than Sherlock, because of its community and size of library and implementations
- Sherlock and Teledyne products, on the other hand, suffers of lack of documentation and community but its simplicity for common tasks (compared to OpenCV) results on realizing this tool as standar on the industrie

Conclusions

Ramon Romero: It was a hard challenge, because electronics is not one of my areas of expertise, but at the end hardware and software have to merge in order to do something great. It is important to know this kind of technologies as they are predominant in the industry but I would prefer to use an open source and bigger alternative such as OpenCV.

Adán Campos: The complexity of this evaluation, managed to give me a broader perspective of the applications for which Sherlock works. The great variety of solutions that can be developed from this software is impressive and it is even more its putting into action. Move the robotic arm of the way we did it in if it was not very complicated, although we did not achieve the greatest challenge we were able to obtain good results recognizing the pattern of the figures to be collected and the alignment of the same to put them in place. A very practical application that has a lot of learning value, waited soon to re-apply what was learned in projects of this type.

References

- [1]Steger, Carsten; Markus Ulrich; Christian Wiedemann (2018). Machine Vision Algorithms and Applications (2nd ed.). Weinheim: Wiley-VCH. p. 1. ISBN 978-3-527-41365-2.
- [2]"Sherlock | Teledyne DALSA", *Teledynedalsa.com*, 2019. [Online]. Available: <https://www.teledynedalsa.com/en/products/imaging/vision-software/sherlock/>. [Accessed: 05- Apr- 2019].
- [2]"Lynxmotion - SSC-32 Servo Controller", *Lynxmotion.com*, 2019. [Online]. Available: <http://www.lynxmotion.com/p-395-ssc-32-servo-controller.aspx>. [Accessed: 05- Apr- 2019].
- [3]"Lynxmotion - AL5A", *Lynxmotion.com*, 2019. [Online]. Available: <http://www.lynxmotion.com/c-124-al5a.aspx>. [Accessed: 05- Apr- 2019].

Sherlock and Scripting Appendix



```
//Declarar variables locales
var AngPen;
var AngTri;
var AngCua;

//Cambiando de Radianes a Grados
AngPen=Vars.Angulo_Pentagono*(Math.PI/180);
AngTri=Vars.Angulo_Triangulo*(Math.PI/180);
AngCua=Vars.Angulo_Cuadrado*(Math.PI/180);

function Rotar(Angulo){
if (Angulo<0){
Angulo=Angulo*(-1);
}
var Y=Math.floor( 11.11*(Angulo)+10);
var comando= "#4 P"+Y.toString() +" T3000\\x0D";
return comando;
}

//Identificar la figura

if (Vars.Pentagonos==1) {
```



```

var ang=Vars.Angulo_Pentagono;
Vars.Rotacion=Rotar(ang);
Vars.PosicionF="#0 P950 T3000\\x0D";
Vars.CasiSoltar="#1 P1190 T3000 #2 P1420 T3000 #3 P910 T3000\\x0D";
}

if(Vars.Cuadrados==1) {
var ang=Vars.Angulo_Cuadrado;
Vars.Rotacion=Rotar(ang);
Vars.PosicionF="#0 P910 T3000\\x0D";
Vars.CasiSoltar="#1 P1300 T3000 #2 P1580 T3000 #3 P880 T3000\\x0D";
}

if(Vars.Triangulos==1) {
var ang=Vars.Angulo_Triangulo;
Vars.Rotacion=Rotar(ang);
Vars.PosicionF="#0 P1045 T3000\\x0D";
Vars.CasiSoltar="#1 P1250 T3000 #2 P1490 T3000 #3 P880 T3000\\x0D";
}

if(Vars.Circulos==1) {
Vars.PosicionF="#0 P1020 T3000\\x0D";
Vars.CasiSoltar="#1 P1280 T3000 #2 P1590 T3000 #3 P760 T3000\\x0D";
}
Vars.Soltar="#5 P1500 T3000\\x0D";

```