Robotic Arm

Second partial project

Gerardo Daniel Naranjo Gallegos, A01209499

Elvis Enrique Jiménez Reyes, A01204803

05/03/2019

Professor: Rick Swenson

Course: Robotic Vision

Tabla de contenido

Introduction

The vision field is one of the most important of the world of the technology. Nowadays, the incredible progress of computers allows the processing of images, and therefore the field of vision, to be increasingly used in different disciplines.

Artificial vision allows, among many applications, to improve and optimize different processes, by replacing old systems.

Project Description

Problem statement

We have four geometric figures: square, triangle, pentagon and circle. The problem is to move a geometric piece with the help of a robotic arm, to a fixed position; in this case, we used a mechanical arm that had six motors. We have a fixed point of origin, where the geometric piece is placed, and from this point the program recognizes which is the piece, takes it, rotates it and leaves it in the predestined place.

This was achieved with the help of the Sherlock software, which has a camera connected to it that is used in industrial environments. With the help of the program, we applied identification algorithms, to be more specific, the "Search Edge" was used, which can learn a model (in this case a piece) which takes as a reference to recognize if there is another one. This is essential for the operation, since once the program recognizes the object, it can be assigned the predestined place for that object. The mechanical arm that has 6 motors and communicates with our program serially, then when each piece has its destination, but all start from the same source.

Material required

For this project, it is necessary to have the following material:

- A source of light that is consistent all the time.
- PC with Sherlock (full version).
- Robotic Arm kit, in this case is the AL5D.
- Camera, in this case provide by Infaimon.
- Pieces to move.
- References for every poses of the robotic arm, camera, light, etc.

Description of the robot

The robotic arm is an AL5D, provide by Lynxmotion. It has six motors for its movements. It has a controller which communicates serially and the commands that are sent go from 0 to 2500, having an average value of 1500. The arm is controlled by Sherlock software.

Develop

Before doing anything, the first step is to take note of the references of all positions of the source light, the camera and the robotic arm; in order to be sure that these will not change. We attached the following images to show it:

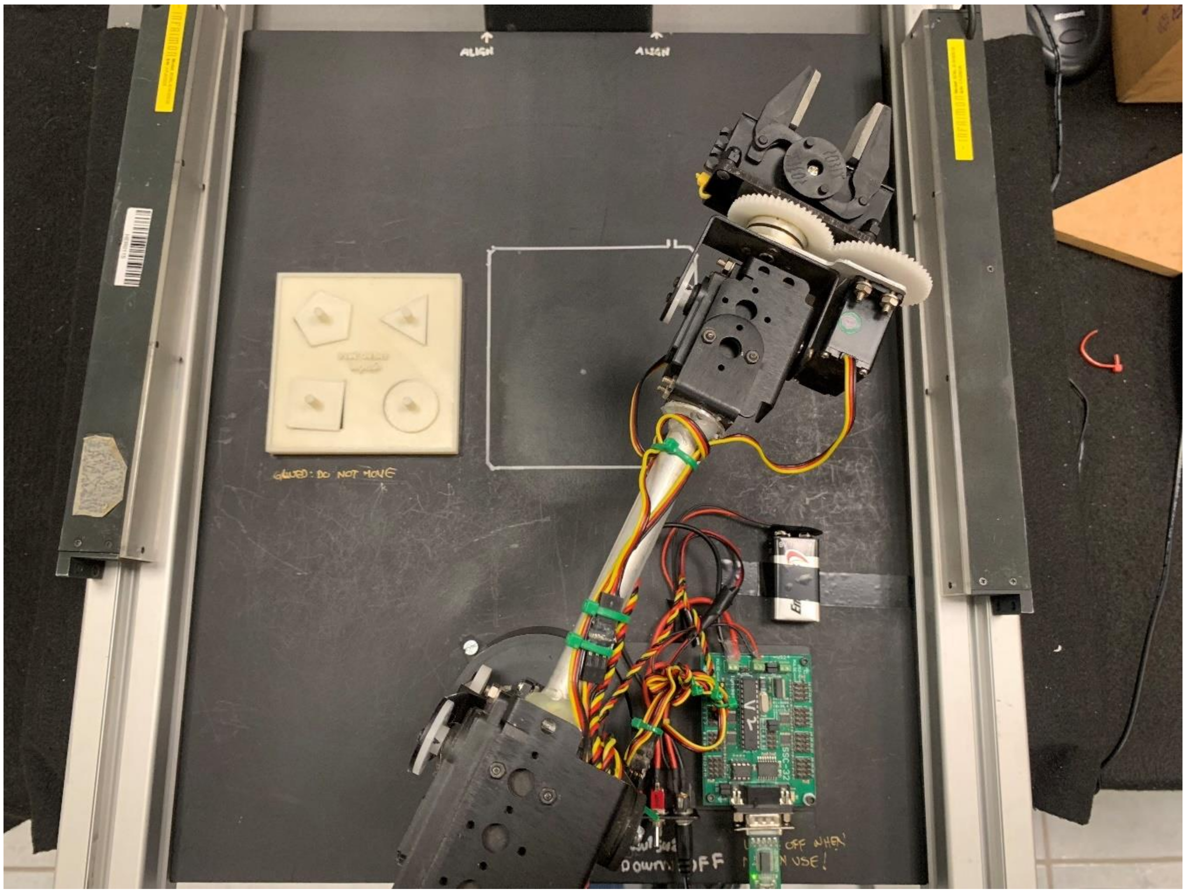

Image 1. Robotic arm (all servos set to home position) and the work table.

Image 2. Alignment of the working table.

Image 3. Focus configuration of the camera.

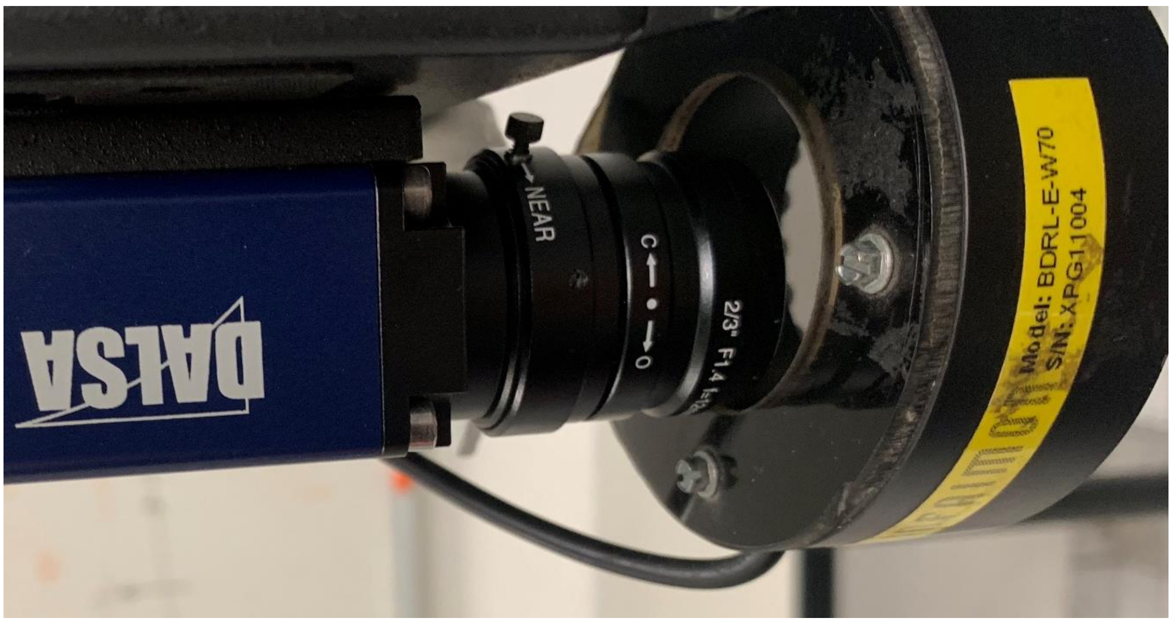

Image 4. Light sensitivity configuration of the camera.

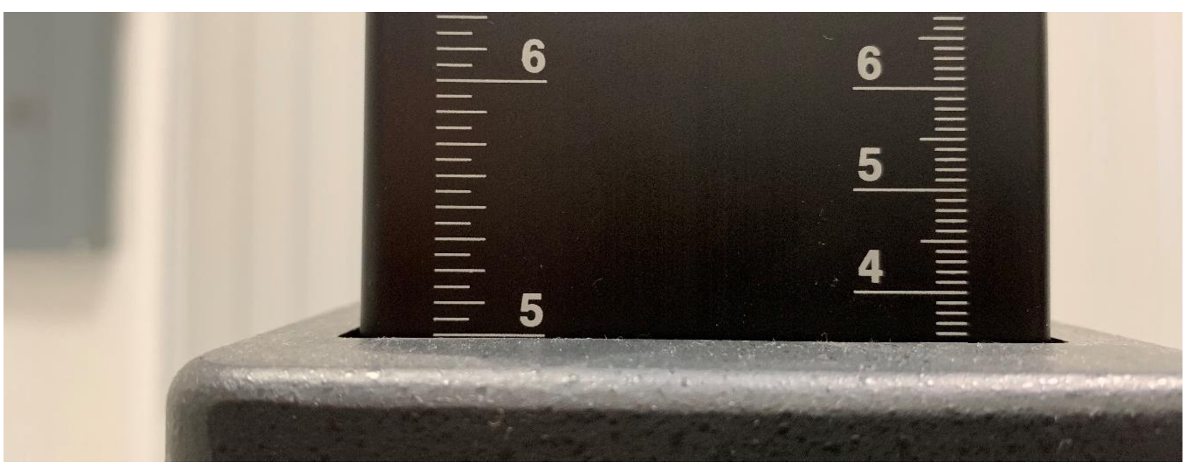

Image 5. Altitude of the camera.

Software SSC-32U servo controller board

Second, we use the software SSC-32 Servo Sequencer Utility, which is a software tool that helps us to move the servos of the motors. We use this software in order to obtain the coordinates of the position for the four geometric figures. The way to use the software is selecting the servo you want to move and increasing or decreasing its value (all the values of the servos will be initially at 1500, which is the home position); it is important to say that, as we have six servos, we use the servos from 0 to 5. Attached is an image of the interface. Also, attached is a table of the description of each servo.

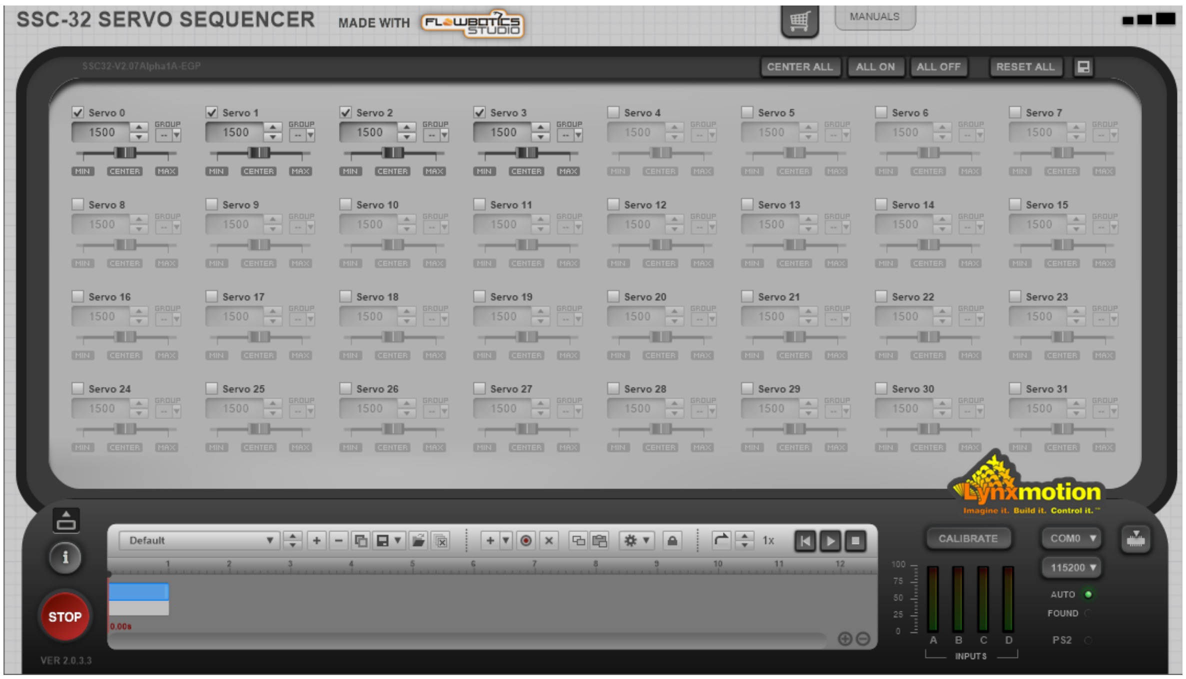

Image 6. SSC-32 Servo Sequencer Utility.

Ch	Function	Center	<1500	>1500	Point of view
0	Base Rotation	1500	Left	Right	Top
1	Shoulder up/down	1500	Lower arm	Rise arm	Side
2	Elbow up/down	1500	Raise arm	Lower arm	Side
3	Wrist up/down	1500	Lower wrist	Raise wrist	Side
4	Wrist rotation	1500	Clockwise	Counter Clock	Front
5	Gripper	1500	Open	Close	Any

Table 1. Description of the servos.

In addition, it can be mentioned for future references that the servos work with the following command line: **#<ch> P<pw> S<spd> ... #<ch> P<pw> T<time> <cr>**; where: PW = posse width in microseconds, SPD = speed and CR = Carry Return character (\x0D).

Sherlock

To configure sherlock in the correct way, we need to:

1) Click on the bottom Options in the image window (in the Image 7), next click on the first option named camera. This helps to have the actual image of the camera.
2) Click on the tab Options in the main window, next click on IO menu, next click on Serial and configure as Table 2.

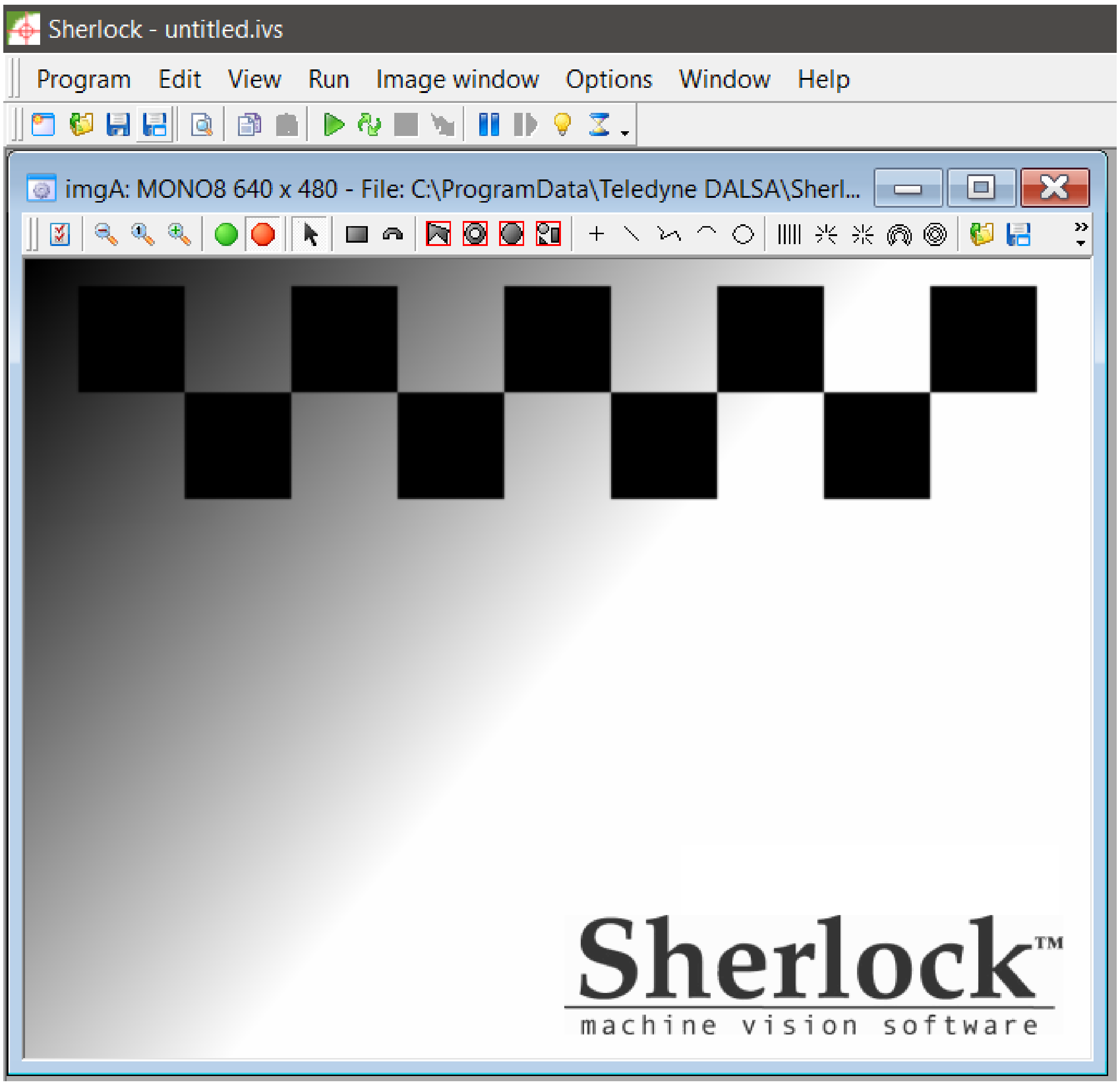

Image 7. Sherlock interface.

Serial Port	(you need to check which COM are you using)
Baud Rate	115200
Data:	8
Parity	None
Stop bits	1
Flow control	None

Table 2. Configuration for Sherlock.

After considering what is our initial position of the piece, so that our program learns it in the correct way, and mark what the initial position is, the movement of the destination position is performed. Our program, recognizing the piece at the starting point, will know what its origin is for each geometric figure. This is achieved thanks to the Search Edge algorithm, which also provides us with the angle of the reference, which we later use to rotate the piece in case it is not in the correct position given the reference it learned.

Several tests were carried out so that the destination movements of each piece were the correct ones, to be able to test the trajectory, the software was used by default of the arm which allows you to move each one of the motors at will. When the program already learns the parts that serve as reference, calculate the angle when rotating with respect to the reference piece, it is done with a simple average that a degree is equivalent to sending the value 5 to the motors, since each motor of the arm you can send values between a range of 0 to 2500, knowing that this is equivalent to go from 0 to 180 degrees, we can take an approximate value to a degree, which serves to adjust the piece according to your reference, performing the following operation:

$$1500 + (Reference\ Angle * (180 / Math.PI) * 5)$$

The result is sent to motor # 4 which controls the rotation of the arm. After the piece is identified by Search Edge, the starting and ending points of each piece have been registered, and the movement of rotation was calculated, it was only a matter of implementing a series of conditionals for each case and executing the actions.

As well already described, in Sherlock we use an ROI to have an area which we apply the Search Edge algorithm, we add models, in this case each piece and in this way, they are recognized. We send the movements in a serial way, by means of a String. For the conditionals of each case, the If tool that provides the software is used, which is very easy to use and programmable, and the scripts are used using the JavaScript language for the computation of data, as in this case the calculation of rotation and to give a small pause between movements

Sherlock was used in the way that the camera captures what it sees now, so the process is done in a physical and direct way.

Scripts

As mentioned earlier in the Sherlock section, the program requires a script, in which the commands that determine the movement of the robotic arm and the different operations and processes necessary to achieve the objective are written.

The used script was to calculate the rotation and to obtain a delay.

Script to Delay:

```
System.Sleep(6000);
```

Script to Rotation:

```
var Time = " T3000 \x0D";

if ( Vars.Angle != 0 )

{

   Vars.NewAngle =   1500 + ( Vars.Angle * ( 180 / Math.PI ) * 5 );

   Vars.Rotate    = "#4 P" + Vars.NewAngle + Time;

}
```

Video

Next, we attach a link to YouTube with a demonstration video for the four figures, where you can see how the robot performs all the processes correctly.

https://youtu.be/O7BZnrWv72s

Referencias

RobotShop, INC. (2019). *About the Robot Arm*. Obtenido de Lynxmotion: http://www.lynxmotion.com/c-130-al5d.aspx

RobotShop, INC. (2019). *FREE Download - SSC-32 Servo Sequencer Utility (Created Using FlowBotics Studio)*. Obtenido de Lynxmotion: http://www.lynxmotion.com/p-895-free-download-ssc-32-servo-sequencer-utility-created-using-flowbotics-studio.aspx