Universidad Tecnológica de Bolívar

Visual Servoing of a Five-bar Linkage Mechanism

by

Susana Navarro Mora

A work submitted in partial fulfillment for the degree of Ingeniera Mecatrónica

in the
Faculty of Engineering
Universidad Tecnológica de Bolívar

January 2017

Declaration

The work presented in this document was carried out at the Universidad Tecnológica de Bolívar, under the supervision of Juan Carlos Martínez Santos - Engineer and professor at UTB. Unless otherwise stated, it is the original work of the author.
While registered as a candidate for the degree of Ingeniera Mecatrónica, for which submission is now made, the author has not been registered as a candidate for any other award. This work has not been submitted in whole, or in part, for any other degree.

SUSANA NAVARRO MORA
Mechatronics Engineering Program
Universidad Tecnológica de Bolívar
Parque Industrial y Tecnológico Carlos Vélez Pombo
Km 1 Vía Turbaco
Colombia

January 2017

Abstract

This document is the written product of the graduation project developed: Visual Servoing of a Five-bar Linkage Mechanism. This project means to venture into the fields of a method of control, with visual feedback, known as Visual Servoing. The contents of this document show a summary of all the theory taken into account to realize the project. They also shows how other people have approached this method. These pages present the project establishing its aims, the importance of its realization, a detailed description of how it was carried out - including experiments and obstacles, - and the results obtained. This document also informs how is this work of use and what can be done from it. In the same way, here are consigned the books, articles, and works consulted in the way, which in their own pages provide a large quantity of references and information.

SUSANA NAVARRO MORA January 2017

Publications

In the course of completeing the work presented in this project, parts of the content have been submitted and accepted for publication in a refereed conference:

Navarro Mora, S. & Martínez Santos, J.C., 2015, 'Visual servoing of a five-bar linkage mechanism', *2015 IEEE 2nd Colombian Conference on Automatic Control (CCAC)*, 1-5.

Acknowledgements

To God;

To my family, for their unconditional support;

To my teacher, Juan Carlos Martínez Santos, for his guidance and dedication;

Thank you.

They always were and always will be essential for my intellectual growth.

SUSANA NAVARRO MORA January 2017

Declaration of Authorship

I, Susana Navarro Mora, declare that this document titled, 'Visual Servoing of a Five-bar Linkage Mechanism' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a professional degree at this University.
- Where any part of this work has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this document is entirely my own work.
- I have acknowledged all main sources of help.
- Where the document is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: __________________________________

Date: __________________________________

"She believed she could, so she did"

R. S. Grey, *Scoring Wilder*

Universidad Tecnológica de Bolívar

Summary

Faculty of Engineering
Universidad Tecnológica de Bolívar

Ingeniera Mecatrónica

by Susana Navarro Mora

Since visual feedback entered the control loop of robot systems many advantages have been introduced to control methods. Hutchinson et al. [1] present how visual information increased the flexibility and accuracy of the tasks performed by the robot systems, by providing higher position accuracy, robustness to sensor noise and calibration uncertainties, and faster reaction to changes in the environment. The result of introducing visual feedback into control loop is a technique known as Visual Servoing.

Visual Servoing refers to using computer vision data to control the motion of a robot [2]. It allows to obtain a vast quantity of information of the system and its environment from one sensor. Visual Servoing provides a solution with a large amount of advantages, like the possibility of path planning, grasping moving targets, and the ability to get a much more information from the same vision sensor.

The project presented in this document is an attempt to venture into Visual Servoing fields. To make it a successful attempt it intends to control the position of a five-bar linkage mechanism, using visual servoing. Using it in a hand-to-eye configuration with an image-based scheme. To meet the goal it is required a hardware component as well as a software one. The hardware component includes a web camera and a five-bar linkage mechanism attached to two DC motors disposed in a hand-to-eye configuration, as well as an interface for hardware-software communication. On the other hand, the software component includes a Processing project that executes a image-based visual servoing and an Arduino interface for software-hardware communication.

Contents

List of Figures

List of Tables

To those who also believe . . .

Introduction

Since visual feedback entered the control loop of robot systems a lot of advantages have been introduced to the control methods. Chen et al. [4] explain how visual feedback helps to estimate forward kinematics of hyper-redundant robots. This wasn't possible with encoder-based systems, that sense joint movements, along with tachometers and mathematical algorithms to get velocity information. Even though it works well for systems with a finite number of degrees of freedom. Hutchinson et al. [1] present how the visual information increased the flexibility and the accuracy of the tasks performed by the robot systems. These advantages are introduced by providing higher position accuracy, robustness to sensor noise and calibration uncertainties, and faster reaction to changes in the environment. One of the results of introducing visual feedback into control loop is a technique known as Visual Servoing (VS).

Visual Servoing refers to the use of computer vision data to control the motion of a robot [2]. It allows to obtain a vast quantity of information of the system and its environment from one same sensor. Its use has increased due to rapid advances in the technologies involved and the decrease in their costs. This has allowed developing more precise control systems taking as a feedback data information from a visual input [5]. Visual Servoing has been used for controlling robots doing many tasks, among those task can be easily found grasping moving targets, picking and placing, seam welding, etc.; tasks that are usually executed by planar robots.

A planar robot is an (idealized) type of robot embedded in the Cartesian plane [6]. This means that movements are made in one single plane. The five-bar, as well as the four-bar, linkage mechanism is a kinematic chain that can perform the functions of a planar robot. However, the five-bar linkage mechanism provides more control due to having one more degree of freedom over the four-bar linkage mechanism [7]. Five-bar mechanisms are widely employed as robotic systems in automated manufacturing operations, such as in the assembly of mechanical and electronic components (pick and place), welding, painting sealing, etc. [8].

The project presented in this document intends to venture into Visual Servoing. At the same time, it tries to provide a more efficient solution of control of the planar robots

of the industry, and a starting point for better systems also based on Visual Servoing. To enter into this topic, the project aims to control the position of a five-bar linkage mechanism using image-based VS with a hand-to-eye configuration. The five-bar linkage mechanism is selected knowing that the use of such mechanism allows a broad reach, according to its geometry; that it has fixated engines that generate its movement; and that it works in a single plane. The hand-to-eye configuration is selected for its facility to work with movements in a single plane. And, the image-based VS is selected to work with the angles read from the image and the known dimensions of the mechanism.
This project was started to help set a precedent of venturing into this area of engineering, at the same time it tries to create a solution for the industry. It intends to be a starting point for those who want to work with other areas of Visual Servoing, as 3D/position-based VS, given the new machines and laboratories now available for the students of UTB. In the contents of the following chapters, there are a literature review of the topics at hand (VS and five-bar linkage mechanism), a detailed description of how the project was carried out - including experiments and obstacles- the results obtained, the work that can be done from it, and the relation to all the studies consulted in the way.

Research aims

The project presented in this document reaches until the calculation of the time and direction engines need to move to reach the desired position, into the control process, including the image processing to find the mechanism joints and the calculation of the difference between both positions (actual and target). This project also leaves the bases so video processing, path planning and an improved system can be developed.

Chapter 1

Theoretical Framework

1.1 Visual Servoing

Visual Servoing is a control method that aims to control robots' motion using visual feedback gotten from a vision sensor [7]. Figure 1.1 shows a general block's diagram that aims to explain how Visual Servoing works. VS emulates the human process of acting on what is seen; which is the sense that humans use the most. It was first heard from Shirai and Inoue to the year 1973 [7] when they explained how to use a visual feedback loop to improve the accuracy of the task done by the robot.

Including a visual sensor as a camera in the system of a robot helps to get vast information about the environment in which it is navigating [9], making it easier to control the robot. To reach its goals, Visual Servoing includes areas like high-speed image processing, kinematics, dynamics, control theory, and real-time computing [1].

According to the configuration of the hardware Visual Servoing can be divided into Eye-in-Hand or Hand-to-Eye. The the kind of information included in the processing classifies Visual Servoing into Position-based, Image-based, and Hybrid. On the other hand, Visual Servoing can be also separated into local or remote, according to where the processing is done. Another classification consists in Direct Visual Servoing, when the control is applied to the joints, or Dynamic look and move Visual Servoing, where a position command is given to the robot controller [10]. More information about these types is explained below, and a detailed explanation of how this types work can be found in Seth Hutchinson's article *"A Tutorial on Visual Servo Control"*.

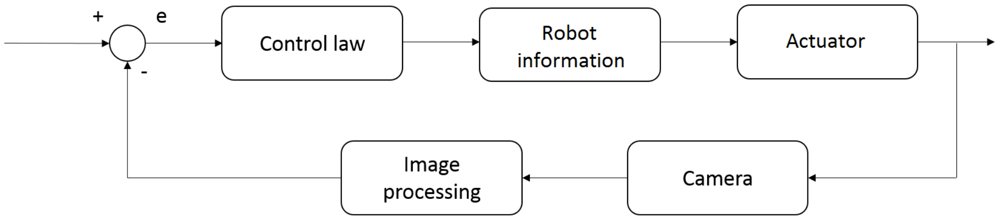

FIGURE 1.1: VS block's diagram

1.1.1 Types of Visual Servoing

Visual Servoing can be classified into different types according to how the error is calculated. Chaumette et al. define these types of VS as schemes and explains the mathematical analysis behind them. Despite doing it differently, the aim of all of the schemes is to minimize the error [2]. Below there is an explanation of the characteristics of the different schemes of VS. Additionally, a more detailed explanation can be found in "Visual Control of Robot Manipulators - A Review" by Peter I. Corke [11].

Position Based

Position-based Visual Servoing - also known as 3D Visual Servoing - refers to the scheme in which the main information is the pose of the camera respect a reference coordinate frame.

To compute that pose from the information obtained from an image it is needed to know the intrinsic parameters of the camera and the 3D model of the object observed [2]. Position-based Visual Servoing works with spacial coordinates and space dimensions to calculate the error. And, in order to execute the control process, there must be first an image processing and a robot's position estimation (Figure 1.2). Seth Hutchinson explains that the features of the image, after being extracted, are used to estimate the location of the target and this information is used to find the error between the current and the desired position [1].

This scheme, apart from needing more information, is very sensible to noises and errors. However, it allows knowing how spacial movement goes. This is why it is often used for 3D motion [12], [13], [14].

Image Based

Image-based Visual Servoing - also known as 2D Visual Servoing - refers to the schem in which image-plane coordinates are used to calculate the error.

Image-based Visual Servoing calculates based on the features of the image captured by

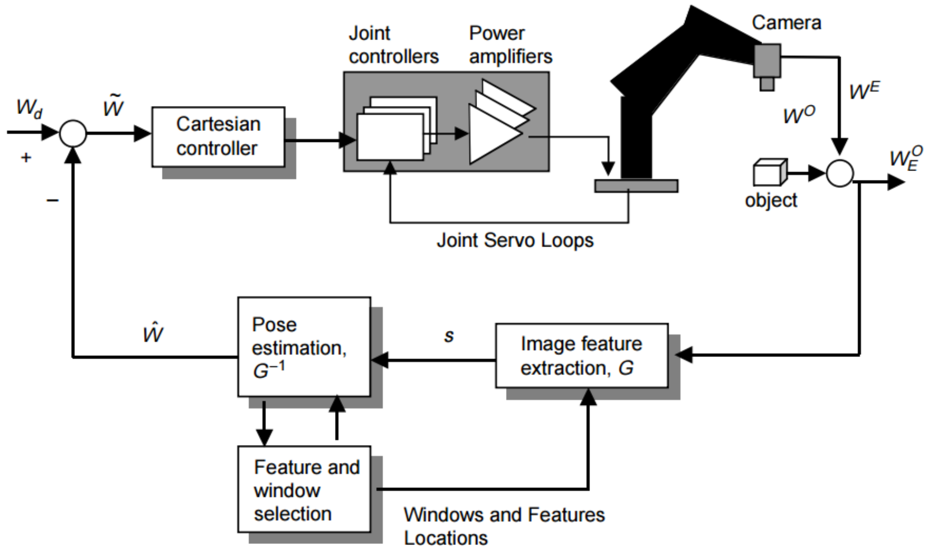

FIGURE 1.2: Structure of Position-based Visual Servoing [3]

the camera. Usually, the image measurements correspond to the pixel coordinates. This scheme of Visual Servoing executes control actions after processing the images, using its dimensions. In image-based VS the error is calculated in terms of image feature parameters [1].

"The image-based approach may reduce computational delay, eliminate the necessity for image interpretation (Figure 1.3) and eliminate errors in sensor modeling and camera calibration" [11] Image based VS effectively works without a 3D model of the robot and handles better noises and errors, not needing a perfect camera calibration [15]. However, spacial movement remains unknown with this technique.

Image-based Visual Servoing works efficiently with an eye-in-hand configuration [1]. And, it is often used for 2D motion, like planar robots. Examples of image-based Visual Servoing can be found in [16], [17], [18], [19].

Hybrid

Hybrid Visual Servoing merges the advantages of both position and image based VS when combined effectively and efficiently. In order to do that, hybrid VS uses the translation analysis of the image-based control and rotational analysis of the position-based control (Figure 1.4). Hybrid Visual Servoing solves the inability of the Position-based VS to directly control the position of the image features in the image plane, hence the risk to lose some image features from the field of view of the camera by using the capability of the Image-based VS to do so not being affected by camera calibration error. On the other hand, Hybrid Visual Servoing avoids undesirable/unpredictable movements with

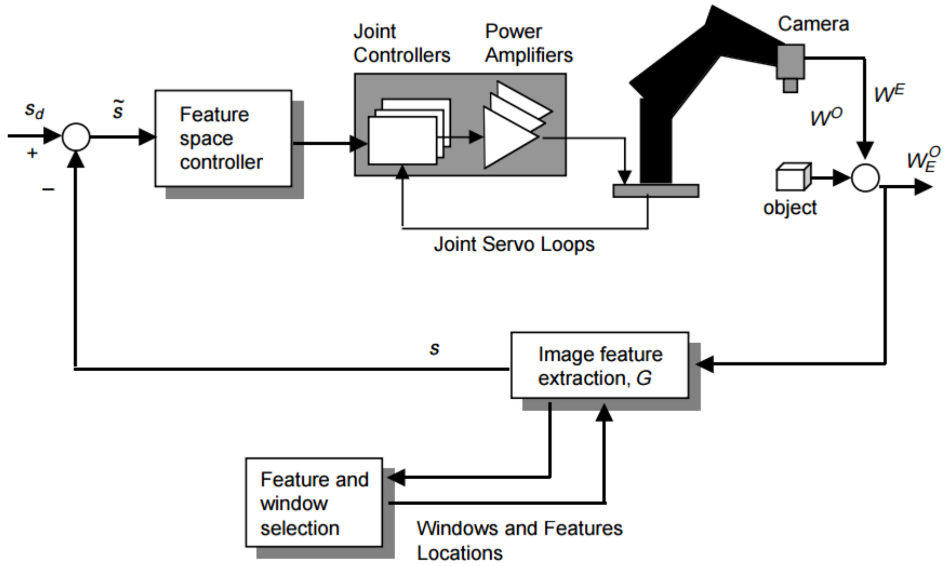

FIGURE 1.3: Structure of Image-based Visual Servoing [3]

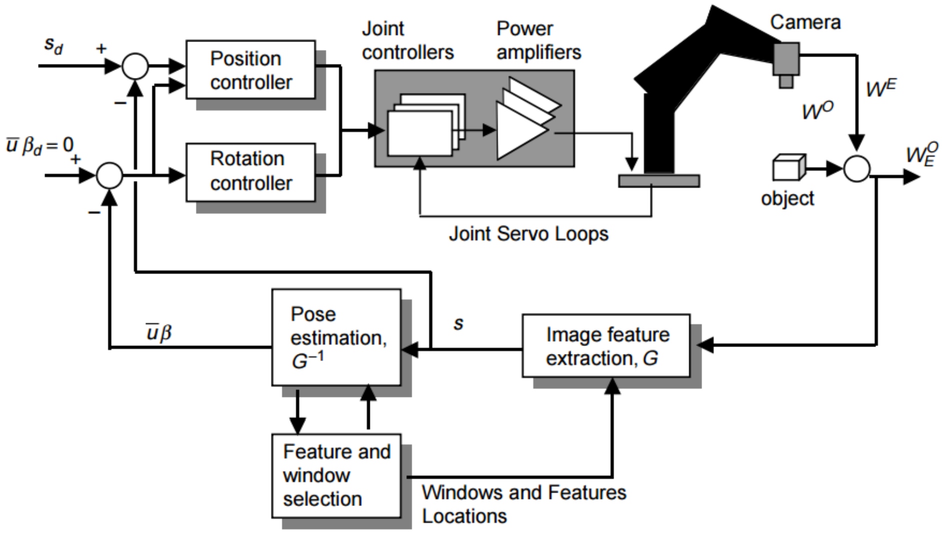

FIGURE 1.4: Structure of Hybrid Visual Servoing [3]

the direct control of the system movement in the Cartesian space of the Position-based VS. In case of not avoiding them, it could end in collisions [20].

1.1.2 System Settings

Eye in Hand

In the 'Eye in Hand' setting the camera goes joined to the final effector as shown in Figure 1.5. This implies that both perform the same movement.

For this configuration, the position difference, between the robot's final effector and the

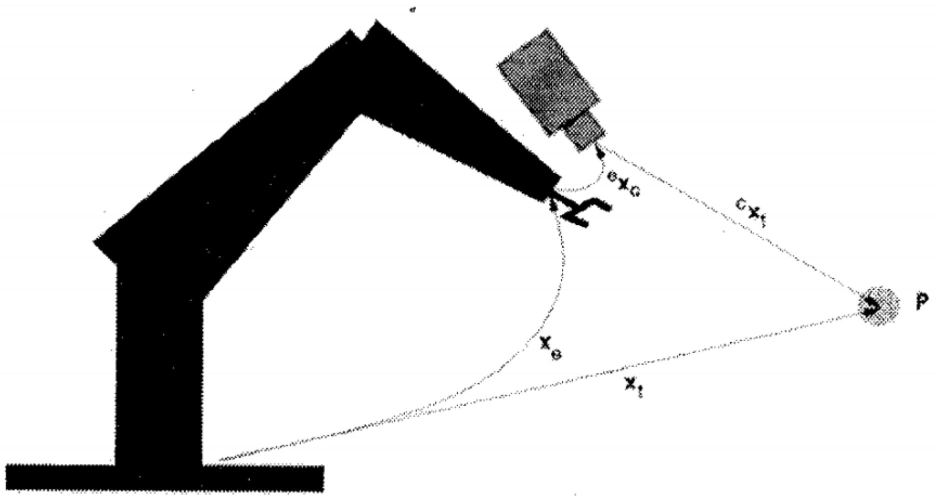

FIGURE 1.5: Eye in hand configuration [1]. Point P is the target, other labels are the ones used by Hutchinson et al. in their analysis and calculations.

camera, is constant. Then, the movement the robot has to do is known once the goal position appears in the image.
This setting allows better viewing possibilities with less probability of viewing obstruction from the moving arm and target [3]. This is often used along with Position-based Visual Servoing controlling 3D motion [21], [22].

Hand to Eye

In the 'Hand to Eye' setting the camera is fixated always to the same point in the workspace as shown in Figure 1.6. This implies that the robot is in the field of view of the camera.
For this configuration, the camera's position is related to the coordinate system of the robot and the goal position. This means that the goal position does not change with the robot's movements.
The Hand-to-Eye - also known as fixed-camera - is often used along with Image-based Visual Servoing to control 2D motion [23], [4].

1.1.3 Processing

According to the distance between where the processing is done and where the robot is there are two types of Visual Servoing. The first one is performed by a computer and is known as remote processing. The second one uses an embedded system and is known as local processing. The election between these two options must be made based on the mobility and the requirement of processing of the system. Local processing

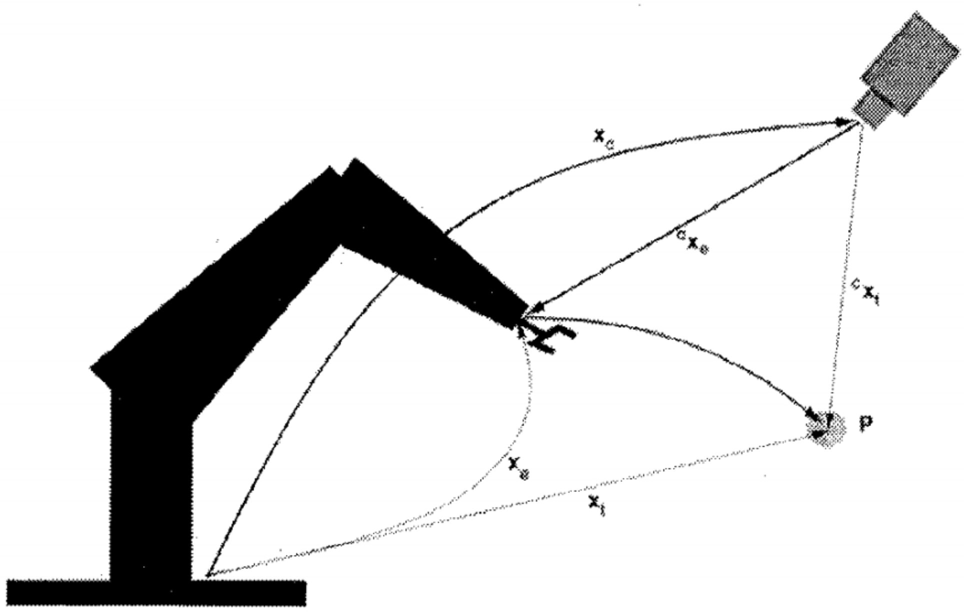

FIGURE 1.6: Hand to eye configuration [1]. Point P is the target, other labels are the ones used by Hutchinson et al. in their analysis and calculations.

is recommended when the system requires mobility. On the other hand, when it is required high capability of processing but the system remains in a fixed workspace, it is recommended a remote processing.

1.2 Computer Vision

Computer vision is a field that studies the images and the image processing techniques in order to understand how vision works and be able to create machines with similar capacities to humans [24]. Open CV is library created to help reach this goal [25].
Even though they are similar, computer vision and image processing are not the same thing. Computer vision extracts features from images in an attempt to describe and interpret it through the computer. On the other hand, the goal of image processing as a field is to improve the quality of the images to allow an easier interpretation of them. Computer vision allows to recognize and locate objects in the environment through image processing.
Computer vision is now worldwide used for applications like the ones listed below [26].

- Optical character recognition. Reading handwriting and automatic number plate recognition.
- Machine inspection. Inspection on aircraft wings or auto body parts.
- Retail. Automated checkout lanes through object recognition.

- 3D model building. Construction of 3D models from aerial photographs.
- Medical imaging. Brain morphology studies.
- Automotive safety. Detection of unexpected obstacles.
- Match move. Estimation of 3D camera motion and shape of the environment.
- Motion capture.
- Surveillance. Monitoring for intruders, analyzing highways traffic, and monitoring pools for drowning victims.
- Fingerprint recognition and biometrics.
- Robotic system control. The one applied to this project.

1.2.1 Types of Computer Vision

According to the use given to the image processing, computer vision can be separated into industrial and scientific uses. Each one requiring a different proceeding. For its scientific application it is required to

- capture the image,
- improve the quality of it,
- determine the elements to measure,
- measure,
- save measurements and realize graphic or statistical processes.

One the other side, for its industrial application - the one relevant for this project - the process consists on

- image capturing,
- defining the region in which measurements will be done,
- comparing to tolerances to determine if the piece is or is not correct,
- measuring, and
- generating a result.

These applications also differ in their goals. While the industrial application looks for speed of processing, the scientific application goes for the analysis of complex images [27].

1.2.2 Vision System

To apply computer vision it is needed a vision system. A vision system depends on its components to work properly. It must contain at least the steps listed below [27].

- Caption. Obtaining visual image.
- Pre-Processing. Noise reduction and details enhancing.
- Segmentation. Division of the image into objects of interest.
- Description. Obtainment of convenient features to differentiate the types of objects.
- Recognition. Assignation of a meaning to a recognized group of objects.

In order to complete these steps it is necessary an architecture composed by the following items [24].

- Capture device. A vision sensor. A device sensible to the electromagnetic spectrum that generates and electrical signal according to the energy detected.
- A/D Converter. Converts the electrical into a digital signal.
- Video memory. Where the digital signal is saved.
- Processor. To manipulate the image

1.2.3 Image

The base of image processing and computer vision is the image as the information obtained from a vision sensor. An image is what occurs when a sensor captures the light that has interacted with other objects [24]. Image obtained by the sensor can be taken as a bi-dimensional function. It is often associated to a coordinates system with origin on it upper left corner.
A digital image is the discretization of an image. This means that the properties of the image become numbers in a range. Additionally, the digital image can now be treated as a matrix filled with the values of its properties. These properties - like color or pixel location - are the ones evaluated to identify objects in the environment.

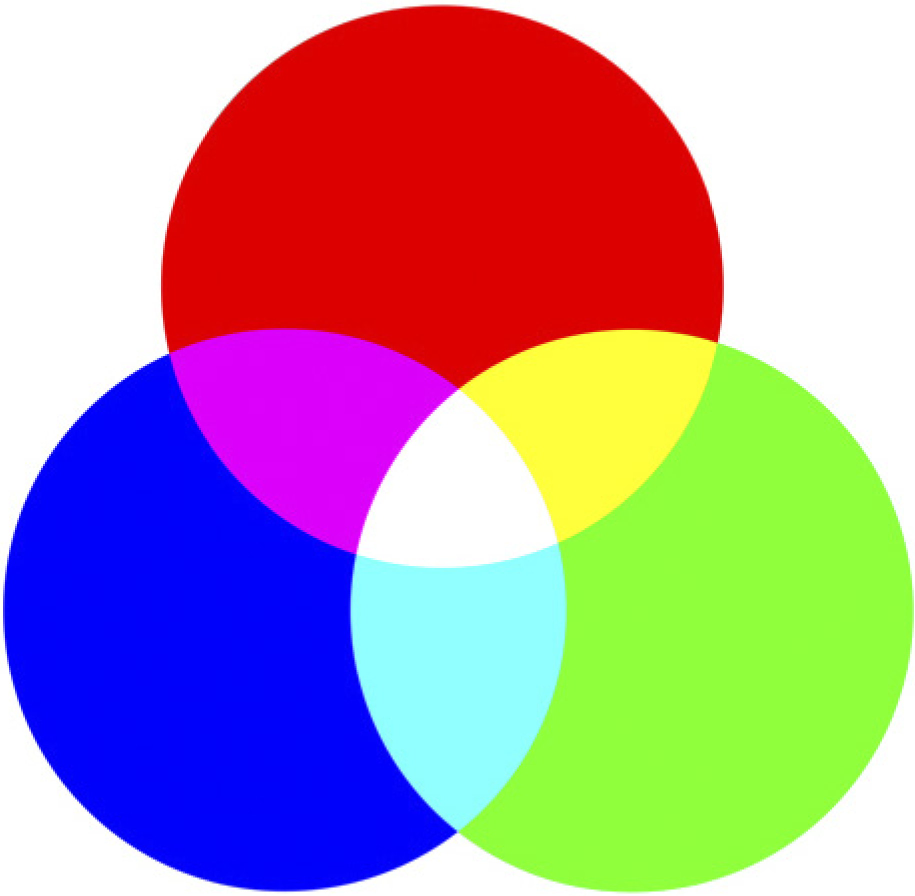

FIGURE 1.7: RGB color model

1.2.4 Color

Color is a phenomenon associated to the human response to different length of waves. This response is, on its own, composed by three different responses of three different sensors in the eye. Colors can be organized/coded according to their basic components [24]. Said organizations are called models/space. RGB and HSI are examples of these models. Other spaces are: CIE XYZ, CMYK.

RGB

RGB space is based on the three human sensors. It establishes that every color is a combination of three basic colors: Red, Green, and Blue [24]. Every color can be obtained by mixing different proportions of these colors, while white is a mixing of the three components in equal parts. RGB space is a light additive color model. This means that colors have to be added in order to produce a lighter color ending in white [28] (Figure 1.7). This model is the used to produce images on the screen of computers.

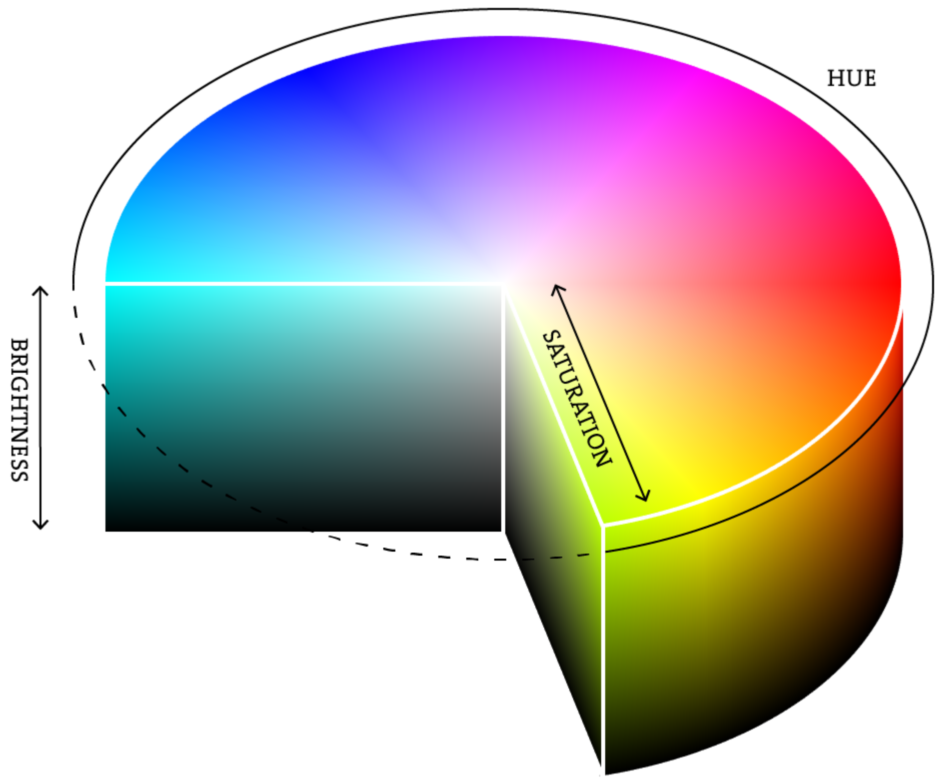

Figure 1.8: HSI color model

HSI

HSI (Hue, Saturation, Intensity) is a color model commonly showed as a cylindrical projection of the chromatic circle defined for RGB model. In this cylinder Intensity is related to height, Saturation is related to the distance from the central axis, and the Hue is related to the angle [24] (Figure 1.8).

CMYK

CMYK is a subtractive color model. This model is used for most printers. That means that to obtain the lighter color, ink must be removed [28]. The basic color for this model are: Cyan, Magenta, Yellow, Key (Black) (Figure 1.9).

1.3 Five-bar Linkage Mechanism

The five-bar linkage mechanism, a two-degree of freedom kinematic chain, can perform the functions of a planar robot as soon as any change in its two input angles can only generate movement at maximum, two directions. Hence, it is useful when the task ahead

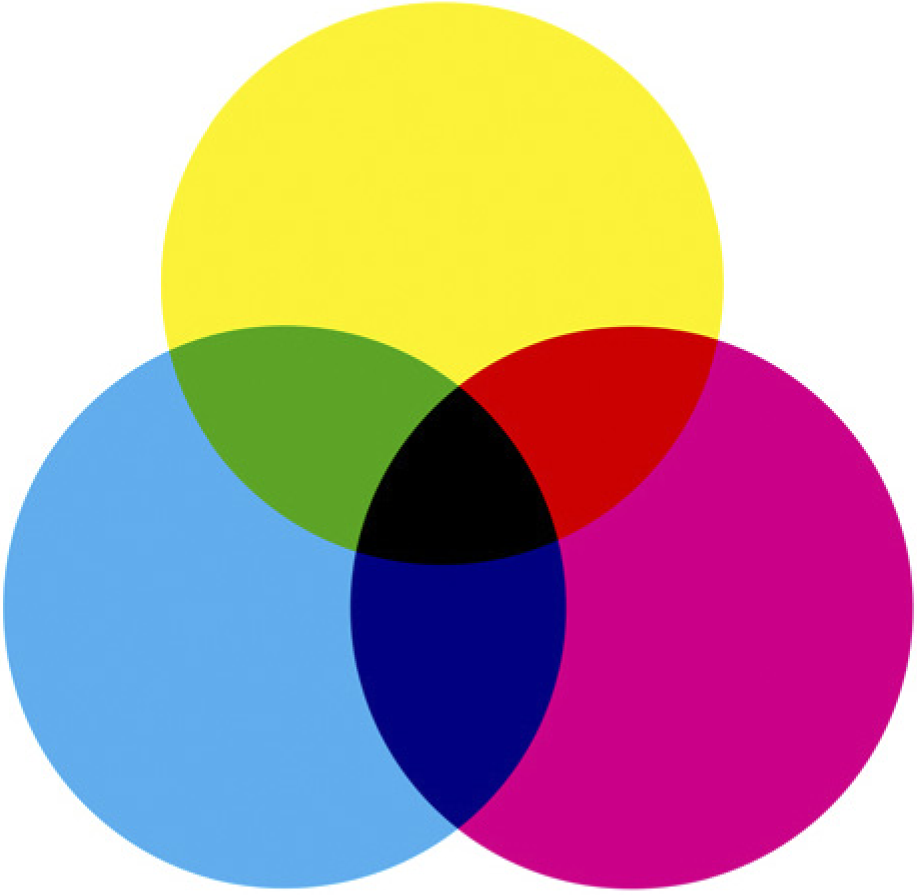

FIGURE 1.9: CMYK color model

involves movement on a single plane or movement in a third direction does not change the position of the end effector in the plane above mentioned. Five-bar mechanisms are widely employed in automated manufacturing operations, such as in the assembly of mechanical and electronic components (pick and place), welding, painting sealing, etc. [11]. Four-bar is another mechanism widely used for these tasks. However, the five-bar linkage mechanism offers one more degree of freedom. This brings greater mobility as well as a greater control because of the second external force that has to be supplied in order to generate the desired movement [8].

1.3.1 Position Analysis

In order to calculate the angles required to reach the desired position, it is realized the position analysis [29]. A quick access to the results of the analysis of position of a five-bar linkage mechanism is shown in the mathematical development below, which leads to the functions presented in Equation 1.4 and Equation 1.7. This analysis enables to determine the relationship between the end-effector (C) and input angles (1 and 4), shown in Figure 1.10. From this analysis, we obtain the expressions for the input angles

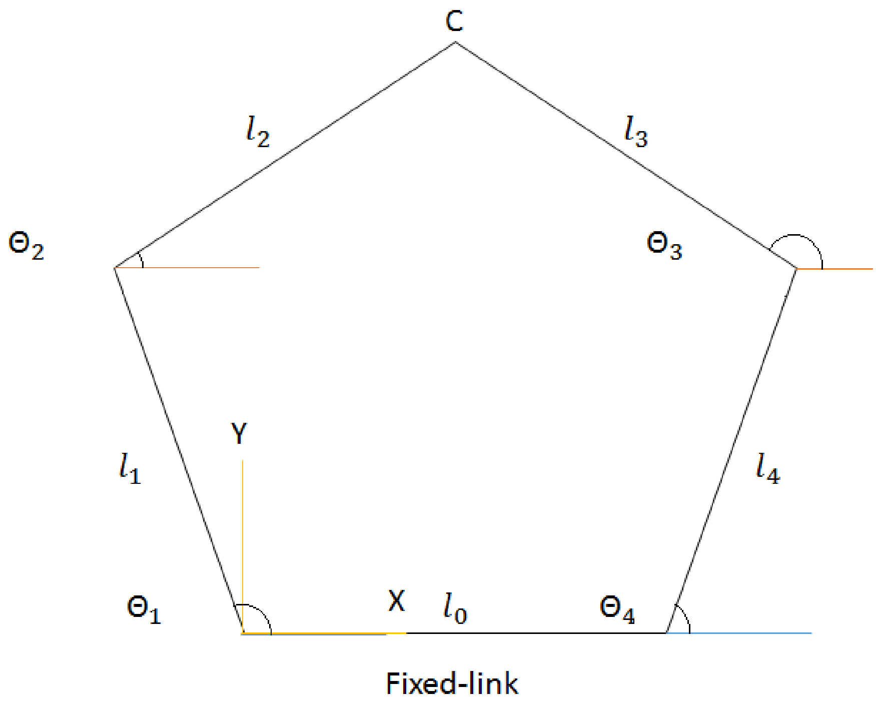

FIGURE 1.10: Scheme of five-bar linkage mechanism

based on the current position in the plane of the end-effector. This will allow to know the value of the entry angles for a specific output point.

Θ_1 as a function of end-effector position

Mathematically, point C can be described as

$$C = \begin{bmatrix} C_x \\ C_y \end{bmatrix} = \begin{bmatrix} l_1 cos(\theta_1) + l_2 cos(\theta_2) \\ l_1 sin(\theta_1) + l_2 sin(\theta_2) \end{bmatrix}$$

from where is obtained that

$$l_2 cos(\theta_2) = C_x - l_1 cos(\theta_1)$$

and

$$l_2 sin(\theta_2) = C_y - l_1 sin(\theta_1)$$

squaring these two expressions and adding one to the other is gotten that

$$C_x cos(\theta_1) + C_y sen(\theta_1) = K_1$$

where

$$K_1 = \frac{C_x^2 + C_y^2 + l_1^2 - l_2^2}{2l_1} \tag{1.1}$$

writing it in terms of tangent of half angles gives the expression

$$C_x - C_y tan^2(\frac{\theta_1}{2}) + 2C_y tan(\frac{\theta_1}{2}) = K_1 + Ktan^2(\frac{\theta_1}{2}) \tag{1.2}$$

which rewritten and solved as a cuadratic equation results in

$$tan(\frac{\theta_1}{2}) = \frac{C_y \pm \sqrt[2]{C_x^2 + C_y^2 - K_1^2}}{C_x + K_1} \tag{1.3}$$

where is gotten that the angle is

$$\theta_1 = 2tan^{-1}[\frac{C_y \pm \sqrt[2]{C_x^2 + C_y^2 - K_1^2}}{C_x + K_1}] \tag{1.4}$$

Θ_4 as a function of end-effector position

Point C can also be described as

$$C = \begin{bmatrix} C_x \\ C_y \end{bmatrix} = \begin{bmatrix} l_0 + l_4cos(\theta_4) + l_3cos(\theta_3) \\ l_4sin(\theta_4) + l_3sin(\theta_3) \end{bmatrix}$$

from where is obtained that

$$l_3cos(\theta_3) = m - l_4cos(\theta_4); m = C_x - l_0 \tag{1.5}$$

and

$$l_3sin(\theta_2) = C_y - l_4sin(\theta_4)$$

squaring these two expressions and adding one to the other, is gotten that

$$mcos(\theta_4) + C_y sen(\theta_4) = K_2$$

where

$$K_2 = \frac{m^2 + C_y^2 + l_4^2 - l_3^2}{2l_4} \tag{1.6}$$

and, proceeding as before, is found that

$$\theta_4 = 2tan^{-1}[\frac{C_y \pm \sqrt[2]{m^2 + C_y^2 - K_2^2}}{m + K_2}] \tag{1.7}$$

Chapter 2

State of Art

The encoder-based are the control systems by excellence when is being talked about robot control. However, they present difficulty determining the exact robot position due to backlash and flex [30]. Other method used consist on controlling forces [31], but it still leaves behind unexpected elements and path planning. In addition to all of this, since the birth of systems using visual feedback, there has been a growing interest in using vision-based systems in the industry [32].

Since the appearance of Visual Servoing, as a vision-based system, its popularity has grown due to the rapid advances in the technologies involved and the decrease in their costs. It has also represent a disadvantage to the encoder-based systems due to its ability to respond to path planning [33]. VS presents advantages like being able to respond to unexpected situation due to its ability to obtain a large quantity of information from just one sensor, this makes it a desirable option to program answers to those situations or simply having more information of the environment.

Below there are some of the advances accomplished by using VS. Shirai and Inoue [7] successfully controlled the realization of assembly tasks by integrating a computer vision and a computer controlled manipulator. The vision system analyzed scenes with a block a a box to make the line drawing, through line fitting and edge extractions. Then it would compute their relative error of the position and of the posture. This resulted in the use of a feedback loop until the box was precisely assembled.

On the other hand, SRI International [34] used Visual Servoing for bolt insertion and picking moving parts from a conveyor. This was posted in a paper that discussed advanced automation. The paper also covered five principal topics: (1) two dimensional curve segmentation; (2) projector-camera range sensing of three dimensional data; (3) a system for recognition and location of three dimensional parts; (4) modular programmable assembly; and (5) enhancements to robot programming systems (RPS). It presents an algorithm for segmenting a two dimensional curve into straight line segments and circular arcs, said algorithm is applied to silhouette boundaries of industrial parts and to images of their interactions with the plane of light of a projector-camera range sensor.

Hill and Park [35] demonstrated planar and 3D visually-guided motor, as well as tracking and grasping moving parts. They describe and analyze a control system for a robot that derives its control information from a small camera attached to its end-effector. They explore how visual input may be in either of two modes: using conventional lighting or using projected patterns of light to give distance by triangulation. The number of degrees of freedom that may be controlled and their attainable accuracy depend on the mode of illumination, the geometric form of the image, the resolution of the camera, and the relationships between the camera, target, and projector. They explain their dynamic analysis of the system accounts for discrete delays in the control loop as well as the transfer function of the robot itself. They achieved to demonstrate the system in several modes simulating manufacturing operations in static and moving coordinate systems.

Following some of the achievements, Prajoux [36] worked on the VS of a 2 DOF mechanism for following a swinging hook. Later on, Venkatesan and Archibald [37] used two hand-held laser scanners for real-time 5DOF robot control. They mounted two laser range finders orthogonally on the wrist of a six-degree-of-freedom robot. The objective was to have the robot end-effector maintain a constant position and orientation with respect to a flat object moving in 3D space in the field of view of the range finders. They used sensory feedback from the range finders to servo control the robot, in a way that its trajectory is similar to that of the moving object in five degrees of freedom. The robot was capable of maintaining the required pose with respect to the object with a small lag due to data-collection time. It was presented a performance study of the tracking experiment.

Weiss [38] proposed the use of adaptive control for the non-linear time varying relationship between robot pose and image features. Their research on Vision-based robot control was focused on vision processing issues. They formalized an analytical approach to dynamic robot visual servo control systems by first casting position-based and image-based strategies into classical feedback control structures. The image-based structure represents a new approach to visual servo control and presents formidable engineering problems for controller design, including coupled and nonlinear dynamics, kinematics, and feedback gains, unknown parameters, and measurement noise and delays. For this, they designed a model reference adaptive controller.

On their side, Coulon and Nougaret [39] explain how to use a digital video processing system for determining the location of one target, this within a processing window, and they use this information for a closed-loop position control of an XY mechanism. Weber and Hollis [40] develop a high-bandwidth planar-position controlled micro-manipulator.

They presented a method for controlling the endpoint position of a coarse-fine manipulator using a fast vision sensor. They estimated a relative displacement in x and y by visually sampling the workpiece surface within the operating range of the fine manipulator. They also developed a dedicated correlation processor based on three digital signal processors because of the high computational complexity and the speed requirements.
Dickmanns [41] worked with road vehicle guidance. They discussed the precise position control for planar docking between 3-D vehicles. And, demonstrated the application to high speed road vehicle guidance. They accomplished to control with speeds up to 96 km/h in a five ton vehicle. The test was run more than 20 km length, and was driven autonomously several times under various weather conditions.
Additionally, work on VS in a tele-robotic environment was done by Yuan et al., Papanikolopoulos et al., and Tendick et al.. Yuan et al. [42] developed a patent for a telerobotic system adapted for tracking and handling a moving object that includes a robot manipulator, a video monitor, an image processor, hand controls and a computer. The robot manipulator includes a movable robotic arm having an effector for handling an object, a drive system for moving the arm in response to arm input signals, sensors for sensing the position of the arm and for generating arm output signals which characterize the dynamic motion behavior of the arm, and a video camera carried by the arm. The camera responds to motion of the moving object. The video monitor receives an input video signal from the video camera, for displaying an image of the object to a human operator. The image processor is responsive to the output signal of the camera, and is capable of acquiring and pre-processing an image of the object on a frame by frame basis. The hand control is capable of generating a hand control output signal in response to input from a human operator. The computer generates arm input signals and is disposed between the hand control means, the robot manipulator, and image processor. The computer receives output signals from the image processor and the arm output signals and the hand control output signal, and generates arm input signals in response to the received signals whereby the arm tracks the motion of the object.
Papanikolopulos et al. [43] address the problem of integrating the human operator with autonomous robotic visual tracking and servoing modules. They mounted a charge coupled device camera on the end-effector of a robot to servo around a static or moving rigid target. The design allows that in manual control mode, the human operator, with the help of joystick and a monitor, commands robot motions in order to compensate for tracking errors. For the shared control mode, the human operator and the autonomous visual tracking modules command motion along orthogonal sets of degrees of freedom. In autonomous control mode, the autonomous visual tracking modules are in full control of the servoing functions. Finally, in traded control mode, the control can be transferred from the autonomous visual modules to the human operator and vice versa. They present experimental results of all modes of operation, as well as the related issues are

discussed. In certain degrees of freedom the autonomous modules perform better than the human operator. On the other hand, the human operator can compensate fast for failures in tracking while the autonomous modules fail.

Tendick et al. [44] describes a hierarchical telerobotic control system for experiments with model-based vision feedback. The human operator issues commands to a high-level controller which plans actions to be carried out by the low-level controller. The vision system uses a feedforward model of the robot's motion to look for visual enhancements on the robot in their anticipated positions. A middle-level module is planned which will alter low-level control gains to maintain performance within error bounds.

A more detailed explanation of these works, as well as a chronological description of how VS has been growing can be found in [11], a paper that "attempts to present a comprehensive summary of research results in the use of visual information to control robot manipulators and related mechanisms".

The works above mentioned support the idea that when it comes to planar robots is preferred a configuration of hand-to-eye, like the used in this project, to an eye-in-hand configuration, which is more used when there is movement in three dimensions. This preference corresponds to the possibility of obtaining in a single image the error between the desired signal and the current one [45]. This is not possible when the movement is in the space, since more than one image is required in this case, which is resolved by the settings eye-in-hand.

Other studies - presented in this document, section 1.1.1 - have proved that image-based control presents disadvantages respect to a position-based or hybrid control. In certain cases, image-based control presents problems of convergence and stability [46]. Reasons why the position-based and hybrid control have been object of study. However, image-based VS is easier and useful when working on two dimensions, it provides a reduced computational delay and eliminates the necessity for image interpretation [11]. Hassanzadeh, et al. used the hybrid control to generate paths for a robot of three-dimensional movement using a fixated-camera configuration with two computers, one for imaging processing and other for path generation [23]. This approach differs from theirs [23] in the implementation. Instead of using two computer running Simulink (The MathWork Inc., Natick, MA), this approach uses a computer running Processing 2.0 [47] and an Arduino board. The results are similar, but this approach is running at double of processing speed, 30 fps.

Chapter 3

Visual Servoing of a Five-bar Linkage Mechanism

3.1 Overview

The system studied for this project is intended to control the position of a five-bar linkage mechanism using image-based VS with a hand-to-eye configuration. The five-bar linkage mechanism is selected knowing that the use of such mechanism allows a broad reach, according to its geometry; that it has fixated engines that generate its movement; and that it works in a single plane. The hand-to-eye configuration is selected for its facility to work with movements in a single plane. And, the image-based VS is selected to work with the angles read from the image and the known dimensions of the mechanism.

3.2 Proposed System

A Visual Servoing system can be divided in four parts, as explained below, working as shown in Figure 3.1.

- The robot. This includes the robot that is being controlled, as well as the environment it needs to work properly.
- The sensor. This includes the camera, as well as the settings required for a proper caption of data.
- The control. This includes all the elements involved in the processing of the data, for the generation of a control answer.

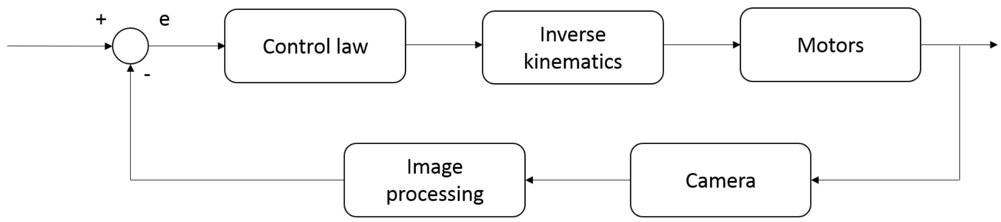

FIGURE 3.1: Block's diagram for the proposed system

- The interconnection. This includes all the elements related to the communication of the parts mentioned above.

3.2.1 Materials

- The robot.
  A five-bar linkage mechanism and two motors attached to it.
  It was used a five-bar linkage mechanism built with pieces of balsa wood joined by bolts of 1/8 in. The links used for the project had a square cross section of 1.2 cm and the following length:
  - L1 = 29 cm
  - L2 = 48 cm
  - L3 = 48 cm
  - L4 = 29 cm

  There were also used two Pololu DC motors (Pololu Corporation, Las Vega, NV).

- The sensor.
  A camera and its support above the robot.
  It was used a QHM495LM Web Camera (Quantum HiTech, New Delhi, India) [48] - with a capture speed of 30fps - held 1.2 m above the mechanism by a PVC structure.

- The control.
  Software for image processing running in Processing 2.0 [47] and calculations explained in this document, section 3.3, running in a personal computer with Windows 10.1 64-bits with 8GB of RAM and a 2.30GHz Intel Core i7 (Optiplex 990, Dell, Round Rock, TX).

- The interconnection.
  An Arduino UNO board (Arduino SRL, Scarmagno, Italy) and a circuit to control the direction of the movements of the motors.

It was used a circuit with a L293D ship that allows to variate the direction of the movement [49] and an Arduino board programmed to turn on the motors for an specified time and direction.

3.2.2 Assembled System

The used VS system consists of the Web Camera connected to a personal computer from which information is received and processed. An Arduino UNO used to transmit the necessary information to generate the desired movement to two Pololu DC motors. The Arduino board works as an interface to control the two motors. The vision system (hardware and software) allows obtaining the position of the mechanism by conducting a position analysis. Once the end-effector is located, the system calculates the difference of its location respect to the desired position. The system sends the commands movement to the Arduino board that immediately translates them into direction of rotation and time, making the shortest possible movement.

3.3 Algorithms

In order to control the position of the end-effector of the five-bar linkage mechanism, it is required to calculate three main pieces of information: the angles forming the initial position of the mechanism, the angles forming the final/desired position, and the movement to get to the final position.The way this information was calculated is explained below.

3.3.1 Initial Position

The process of calculating the initial position of the mechanism is divided in three steps: first, getting the picture of the mechanism in its actual position; second, identifying the joints; and third, calculating the entry angles of the mechanism. The whole process is shown as a flowchart in Figure 3.2. The first step (Figure 3.2 part a) was executed using Appendix A. The second (Figure 3.2 part b) as well as the third step (Figure 3.2 part c) are detailed below.

- Identify joints

  To identify the joints of the mechanism, using VS, there were used markers. Said markers were created by painting a different color each joint. Once the picture of the mechanism is uploaded, a processing code reads, pixel by pixel, the RGB

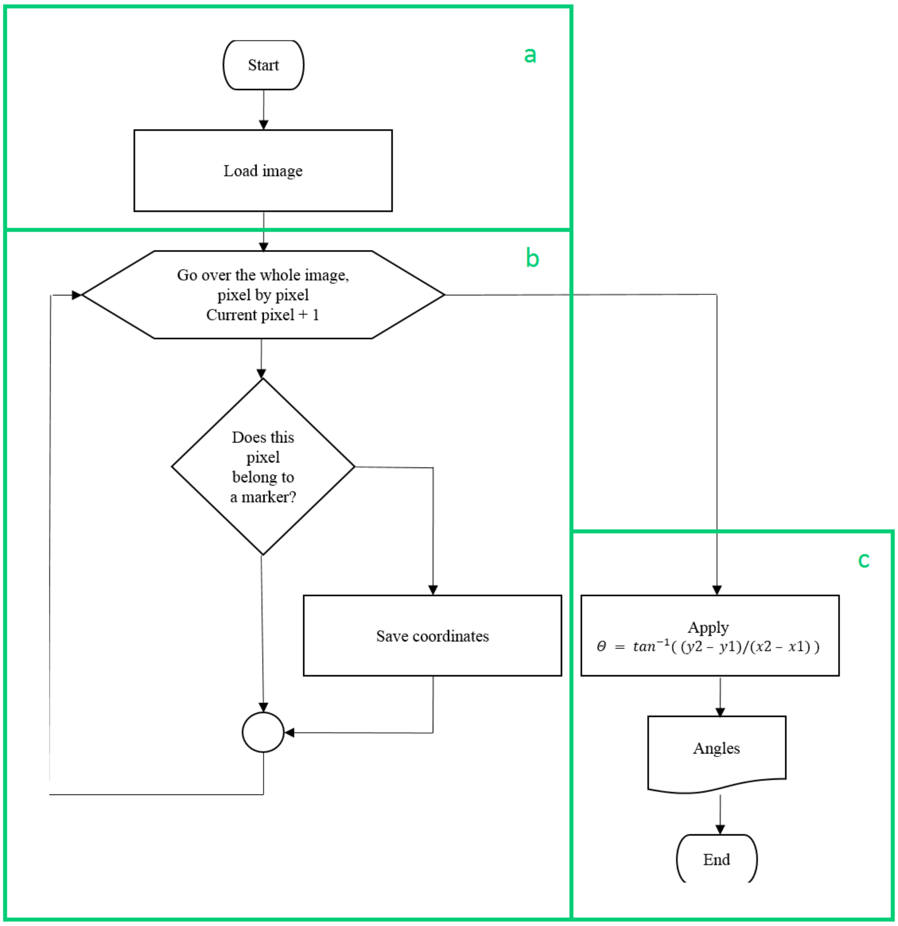

FIGURE 3.2: Flowchart for measuring angles

components and compare them to the ones from the used markers. In case of a match in the RGB components, the location of the pixel is saved in predefined variables (Appendix B).

Due to the variability of lighting, it was established that the RGB components for the same color can suffer little variations. To cover these variations, the RGB components of the read pixel are compared to a range instead of to a unique value.

– Identify RGB parameters for markers

To identify the range of values of each RGB component, of each marker, there were extracted the RGB components (Appendix C)- of each marker - from pictures taken under different conditions of lighting. The data obtained for each marker was averaged and, through trial and error, a range was established. Said range would had the average as the middle value and the combinations of the ranges of the three components would not interfere with any other marker.

- Calculate angles
  Once the markers are identified, the entry angles are calculated using the vertical (Y) and horizontal (X) components in Equation 3.1 (Appendix D). For this calculation X_1 and Y_1 belong to the unmovable/motors joints.

$$\theta = tan^{-1}(\frac{Y_2 - Y_1}{X_2 - X_1}) \tag{3.1}$$

  - Generate models to prove the calculations
    In order to prove the method used to calculate the angles, there were created digital simple models of the mechanism (Appendix E) which angles were calculated and compared to the ones established to the creation of the model (Appendix F). The models were created to cover all the combinations of angles with a variation of ten degrees on each entry.

3.3.2 Final Position

To calculate the final/desired position (Appendix G) it is necessary to convert the coordinates of the desired position of the end-effector into entry angles. This is done using Equation 1.4 and Equation 1.7 for θ_1 and θ_4 respectively.

3.3.3 Movement

Having both the initial and the desired position in terms of entry angles, it is needed to

- Calculate the difference between the angles - initial and final.
- Convert the difference into direction of turn (Appendix H) and time the motor have to be on to reach the desired position.

Chapter 4

Results

4.1 Joints Identification

For joints identification, it was tested the way the system works using RGB color model. Figures 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7 show the results of the identification of the markers. As we can see, the visual system is able to find the markers in all images, even when they have different light conditions. Figures 4.1, 4.2, 4.3, 4.5, 4.6, 4.7 show four scenarios with markers detected out of the working area. On Figure 4.1 is shown the worst case of noise detecting markers, while Figures 4.2, 4.3, 4.5, 4.6, 4.7 - with a more uniform background - present less noise. This noise problem is overcome by having a uniform background as shows Figure 4.4.

4.1.1 RGB Parameters for Markers Identification

In Table 4.1 are shown the averages calculated and the rages established as explained in the last chapter, section 3.3.1

Table 4.1: Averages and ranges for markers identification

	R			G			B		
Markers	lower	**avg**	upper	lower	**avg**	upper	lower	**avg**	upper
Red	59	**69**	79	36.5	**46.5**	56.5	27.25	**37.25**	47.25
Blue	36.67	**46.67**	56.67	49.33	**59.33**	69.33	93.67	**103.67**	113.67
Green	34.5	**44.5**	54.5	59.5	**69.5**	79.5	48	**58**	68
Yellow	91.25	**111.25**	131.25	95.75	**115.75**	135.75	22.75	**42.75**	62.75

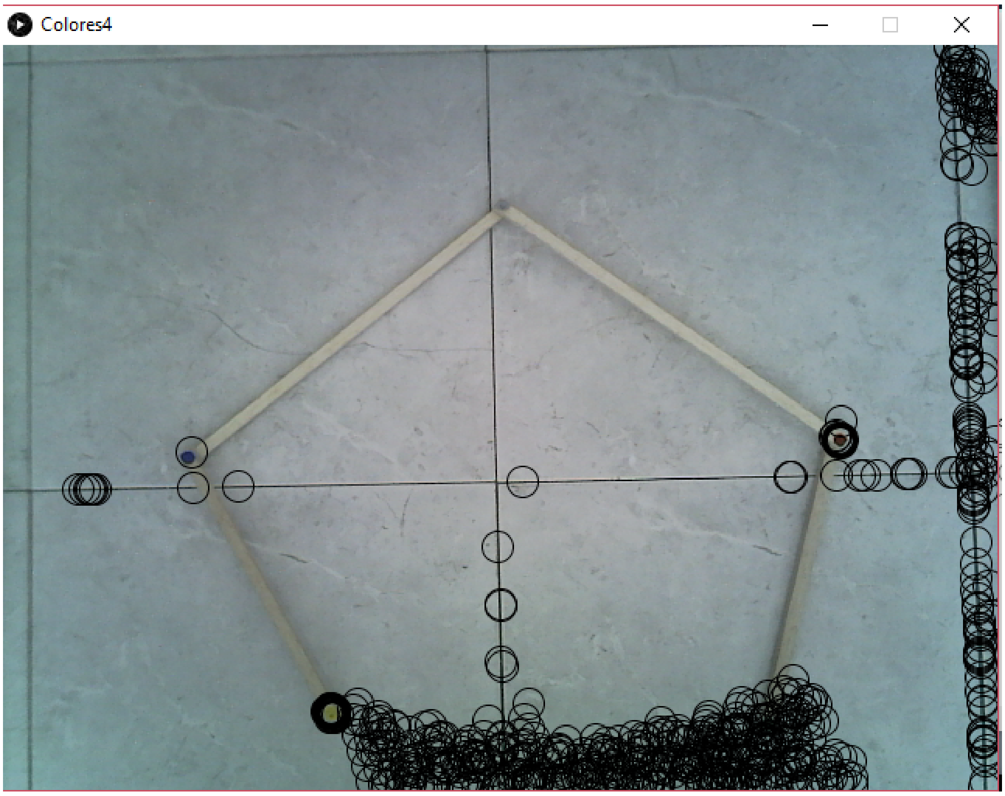

FIGURE 4.1: Example of marker identification. High quantity of noise

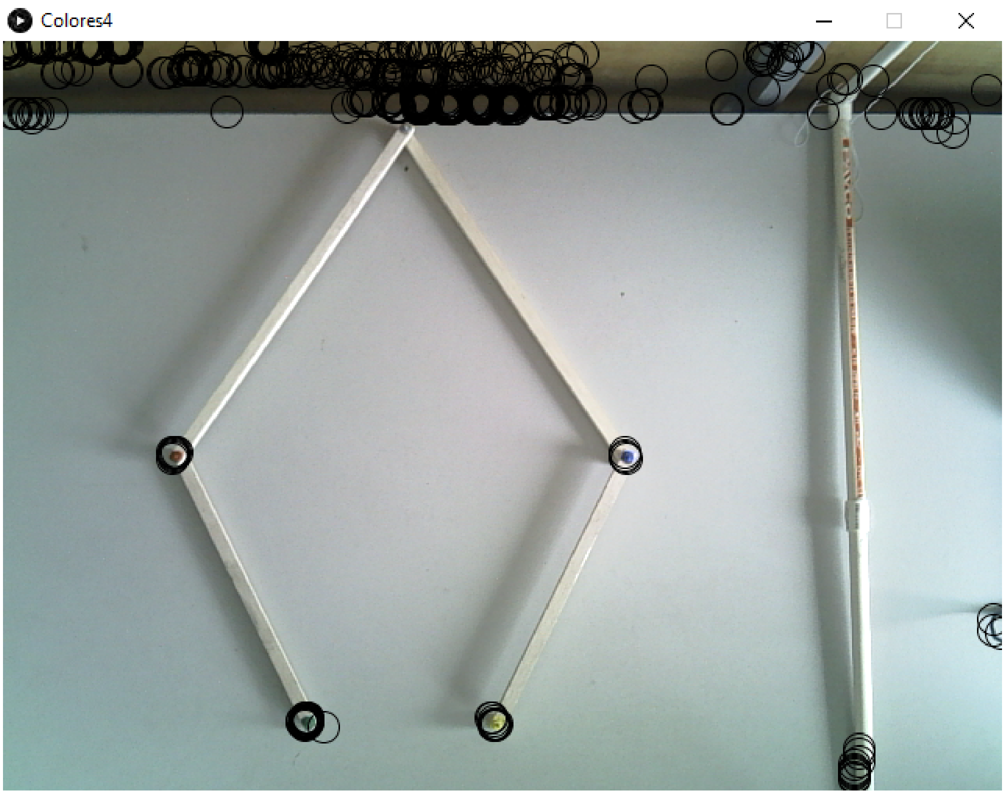

FIGURE 4.2: Example of marker identification. Blue and yellow markers with higher lighting

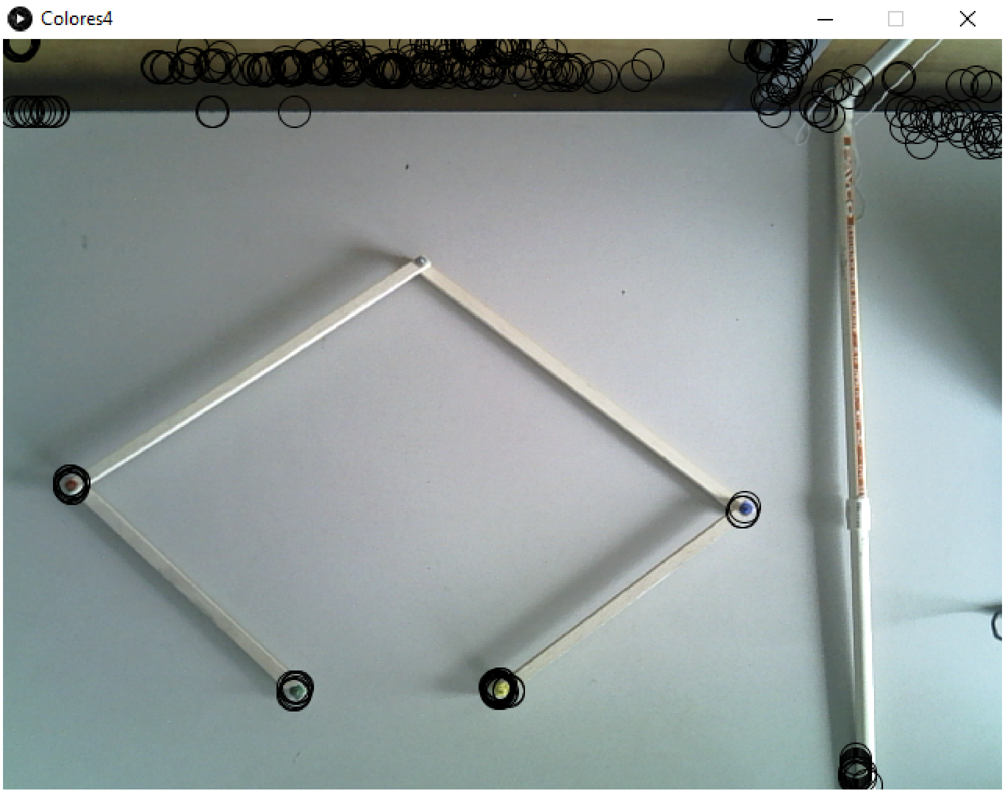

FIGURE 4.3: Example of marker identification.Blue marker with higher lighting, and red marker with lower lighting

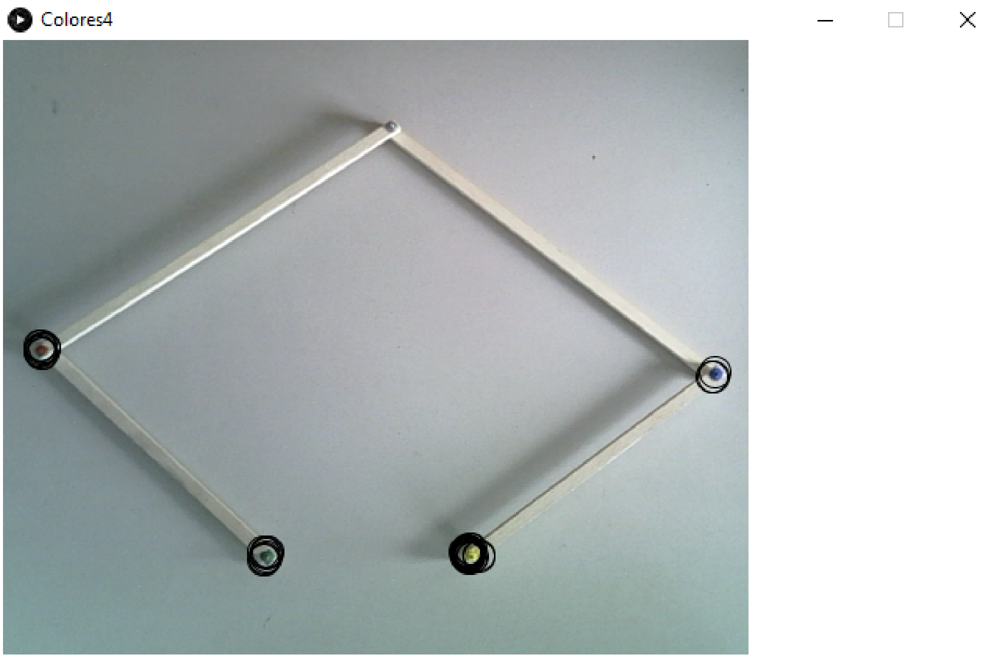

FIGURE 4.4: Example of marker identification. No noise

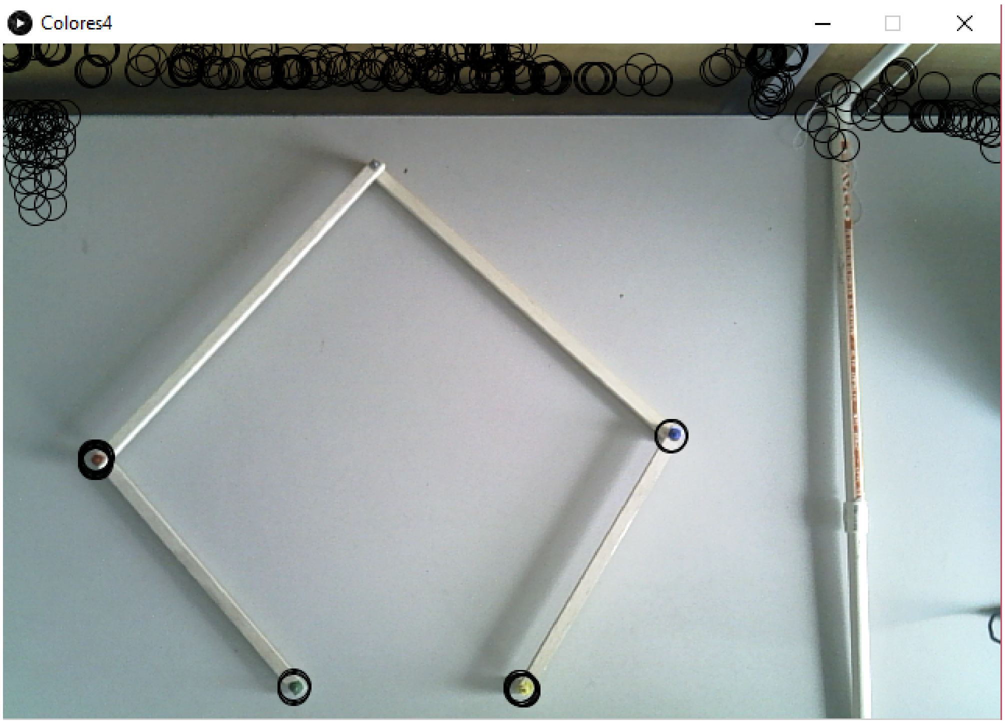

FIGURE 4.5: Example of marker identification. Red and green marker with lower identification

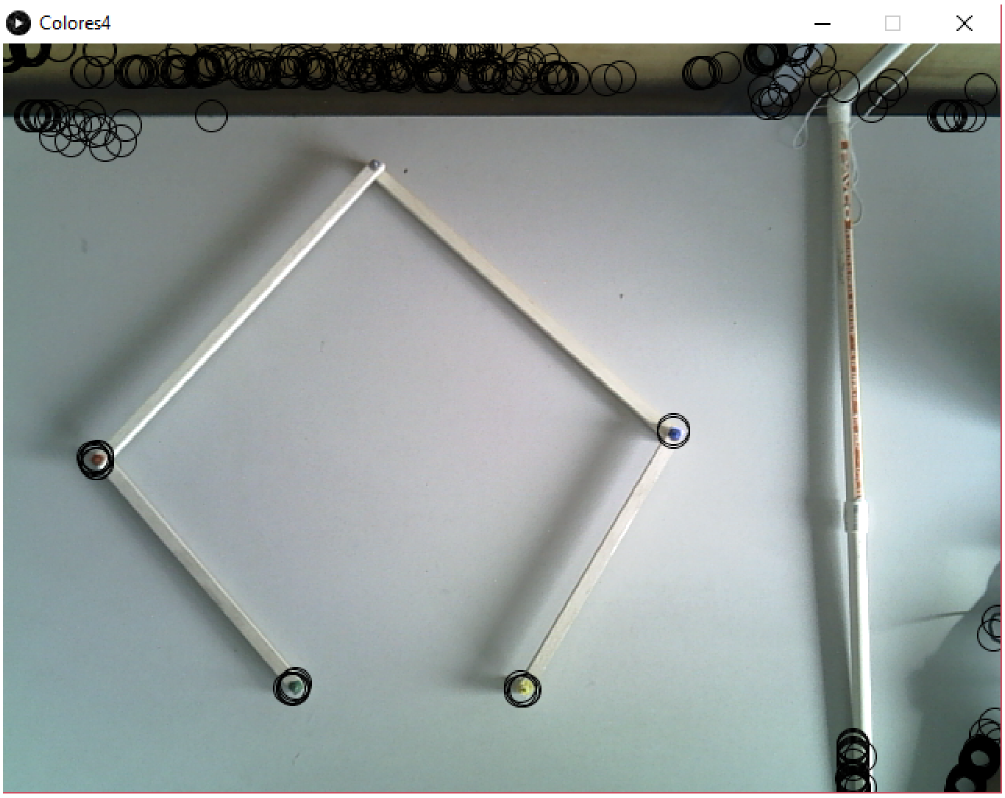

FIGURE 4.6: Example of marker identification.

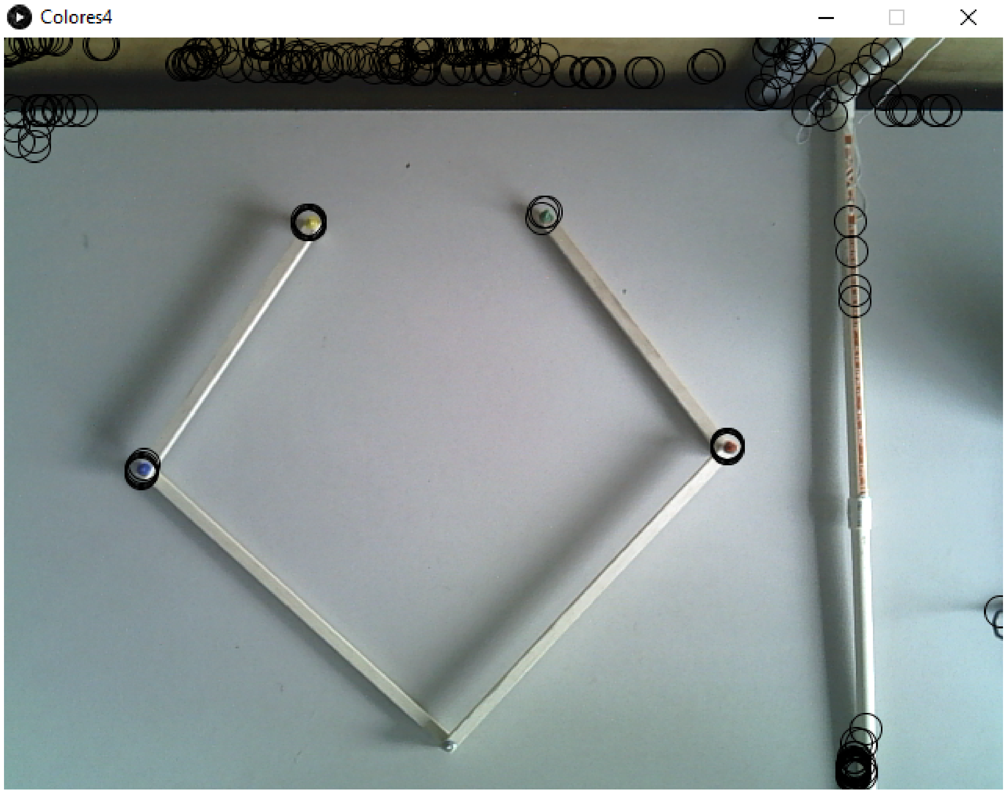

FIGURE 4.7: Example of marker identification. Red and green markers with higher lighting

4.2 Input Angles Calculation

To determine the percentage of error in the angle measure, three hundred twenty-four images were processed using the function to measure the angles between linkages (Appendix F). Each image shows the five-bar mechanism in a different stage with input angles changing between 0° and 180° with an interval of 10°. The angles were calculated based on Equation 3.1. Figure 4.8 shows a given image with input angles $\theta_1 = 150°$ and $\theta_4 = 30°$ while Figure 4.9 shows the measured valued found by the VS system. The idea is to determine how well performs the visual input capture extracting the input angles. The results are shown in Table 4.2 for θ_1 and Table 4.3 for θ_4. As we can see, the error is small, less than 4% in the worst case and less than 1% in average. Using the code in Appendix F it is also calculated the time used to determine the entry angles, obtaining a maximum time of 69 milliseconds, making it a fast process.

4.3 Movement

Even though the difference between the actual and desired position can be calculated as well as the direction of the turn, the movement cannot be generated due to problems

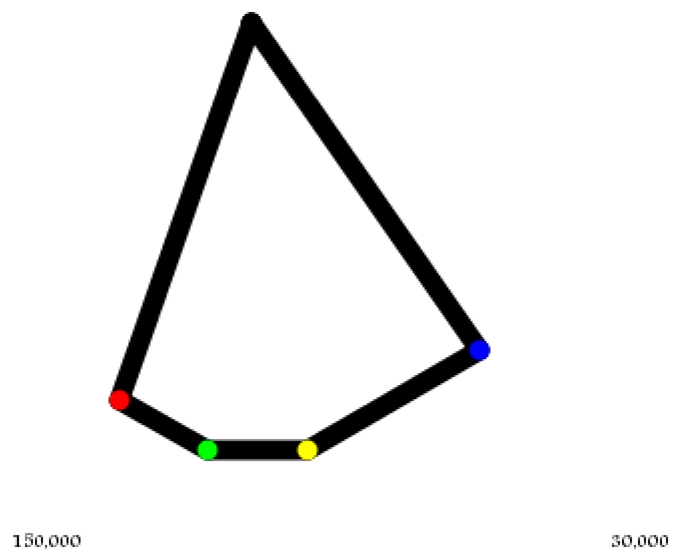

FIGURE 4.8: Example of created models

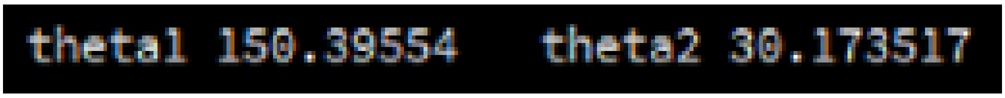

FIGURE 4.9: Results of reading angles for model

controlling the speed of the motors.

Table 4.2: Percentage error Theta 1

Real Theta 1	Measured Theta 1	e%
10,00	10,31	3,08
20,00	20,37	1,85
30,00	30,17	0,58
40,00	40,97	2,43
50,00	50,63	1,26
60,00	60,40	0,66
70,00	70,11	0,16
80,00	80,91	1,14
90,00	90,00	0,00
100,00	100,20	0,20
110,00	110,96	0,87
120,00	119,60	0,33
130,00	130,24	0,18
140,00	139,76	0,17
150,00	150,40	0,26
160,00	159,04	0,60
170,00	169,80	0,12
180,00	180,00	0,00
	Average percentage error	0,77

Table 4.3: Percentage error Theta 4

Real Theta 4	Measured Theta 4	e%
0,00	0,00	
10,00	10,31	3,08
20,00	20,62	3,12
30,00	30,17	0,58
40,00	40,54	1,35
50,00	50,27	0,54
60,00	60,11	0,19
70,00	70,11	0,16
80,00	80,26	0,32
90,00	90,00	0,00
100,00	100,30	0,30
110,00	110,42	0,38
120,00	119,89	0,09
130,00	130,17	0,13
140,00	139,83	0,12
150,00	150,11	0,08
160,00	159,58	0,26
170,00	169,70	0,18
	Average percentage error	0,60

Chapter 5

Discussion and Future Work

In this document was presented the implementation of a Visual Servoing System to a five-bar linkage mechanism. The work done using Visual Servoing allowed to determine the position of each linkage as well as the entry angles of the five-bar mechanism evaluated. Despite the position of the end-effector of the mechanism not being controlled due to problems controlling the speed of the motors, it was proved how VS represents a easy yet very useful method of control and how, through visual feedback, it gives better information about the actual situation of the mechanism.

Getting to know a working speed function for the motors is the next step in the continuation of this work. It will allow to implement the option of path planning, which can be controlled looking for unexpected elements, and will also allow to give a fast answer to the situation. It will also be a starting point to anyone looking for a way to control a planar robot.

Looking to improve the work done, it is wanted to use a high speed camera that will allow to have a higher precision of the measurements. On the other hand, it is wanted to explore different color models that have shown to be more convenient than RGB [50] by being less affected by lighting. Models like CIELAB [51] and CIECAM02 [52] are computational intensive with better uniform color spacing, as well as better perceptual uniformity [53].

Despite this work's purpose being controlling a planar robot, it can be used to start working on non-planar robots, for which [10], [54], and [30] can be helpful. Given the case of wanting to venture into controlling a robot with a non-rigid structure Santosh Kumar Divvala propose a VS method in [9].

Appendix A

Getting Web Camera Image

```
import processing.video.*;

Capture cam;

void setup() {
  size(640, 480);

  String[] cameras = Capture.list();

  if (cameras.length == 0) {
    println("There are no cameras available for capture.");
    exit();
  } else {
    println("Available cameras:");
    for (int i = 0; i < cameras.length; i++) {
      println(cameras[i]);
    }

    // The camera can be initialized directly using an
    // element from the array returned by list():
    cam = new Capture(this, cameras[7]);
    cam.start();
  }
//}

  if (cam.available() == true) {
    cam.read();
  }
  image(cam, 0, 0);

  save("cam.jpg");
  }
```

Appendix B

Identifying Markers

```
PrintWriter output;

void setup(){
size(640,480);
PImage img;
img = loadImage("pic3.png");
image(img, 0, 0);


loadPixels();
float a = millis();
for(int i=0; i<640; i++){
  for(int j=0; j<480; j++){
    color azul=get(i,j);

// Promedios
    if ((red(azul)>59 && red(azul)<79 && green(azul)>36.5 && green(azul)<56.5
    && blue(azul)>27.25 && blue(azul)<47.25) || (red(azul)>36.67 && red(azul)<56.67
    && green(azul)>49.33 && green(azul)<69.33 && blue(azul)>93.67 && blue(azul)<113.67)
    || (red(azul)>34.5 && red(azul)<54.5 && green(azul)>59.5 && green(azul)<79.5
    && blue(azul)>48 && blue(azul)<68) || (red(azul)>91.25 && red(azul)<131.25
    && green(azul)>95.75 && green(azul)<135.75 && blue(azul)>22.75 && blue(azul)<62.75) ) {
//    if ((red(azul)>36.67 && red(azul)<56.67 && green(azul)>49.33
&& green(azul)<69.33 && blue(azul)>93.67 && blue(azul)<113.67) ) {
//    if ((red(azul)>34.5 && red(azul)<54.5 && green(azul)>59.5
&& green(azul)<79.5 && blue(azul)>48 && blue(azul)<68) ) {
//    if ((red(azul)>91.25 && red(azul)<131.25 && green(azul)>95.75
&& green(azul)<135.75 && blue(azul)>22.75 && blue(azul)<62.75) ) {

      noFill();
      ellipse(i,j,20,20);
    }else{
      pixels[i + j*640] = color(255,255,255);
    }

  }
}
float b = millis();
```

```
println(b-a);
}
```

Appendix C

Reading RGB Components of Pixel Under Mouse Pointer

```
void draw(){
loadPixels();
    color punto=get(mouseX,mouseY);
    float puntor = red(punto);
    float puntog = green(punto);
    float puntob = blue(punto);
    println(puntor+"   "+puntog+"   "+puntob);
}
```

Appendix D

Calculating Angles

```
float theta1 = (180/PI)*atan2(AY-BY,BX-AX);
float theta2 = (180/PI)*atan2(EY-DY,DX-EX);

println("theta1 " + theta1);
println("theta2 " + theta2);
```

Appendix E

Creating Models of the Mechanism

```
PFont f;
f = loadFont("Andalus-20.vlw");

size(500, 500);
background(255);

int L0 = 50;
int L1 = 50;
int L4 = 100;
int L2 = 200;
int L3 = 200;
float T0 = 0*PI/180;

for ( int i = 0; i <= 360; i = i + 10) {
  for ( int j = 0; j <= 360; j = j + 10) {
    background(255, 255, 255);

    float T1 = (180 - i)*PI/180;
    float T2 = j*PI/180;
    int AX = 200;
    int AY = 300;

    float EX = AX + L0*cos(T0);
    float EY = AY - L0*sin(T0);
    float BX = AX + L1*cos(T1);
    float BY = AY - L1*sin(T1);
    float DX = EX + L4*cos(T2);
    float DY = EY - L4*sin(T2);

    float A1 = L0*cos(T0) - L1*cos(T1) + L4*cos(T2);
    float B1 = L0*sin(T0) - L1*sin(T1) + L4*sin(T2);
    float C1 = (L2*L2 - A1*A1 - B1*B1 - L3*L3)/(2*L3);
    float T4 = 2*atan2(B1 + sqrt(B1*B1 - C1*C1 + A1*A1), C1 + A1);
```

```
    float A2 = - L0*cos(T0) + L1*cos(T1) - L4*cos(T2);
    float B2 = - L0*sin(T0) + L1*sin(T1) - L4*sin(T2);
    float C2 = (L3*L3 - A2*A2 - B2*B2 - L2*L2)/(2*L2);
    float T3 = 2*atan2(B2 - sqrt(B2*B2 - C2*C2 + A2*A2), C2 + A2);

    if (T1 > PI && T2 > PI) {
      //      T4 = 2*atan2(B1 - sqrt(B1*B1 - C1*C1 + A1*A1), C1 + A1);
      T3 = 2*atan2(B2 + sqrt(B2*B2 - C2*C2 + A2*A2), C2 + A2);
    }
    float CX1 = BX + L2*cos(T3);
    float CY1 = BY - L2*sin(T3);
    float CX2 = DX + L3*cos(T4);
    float CY2 = DY - L3*sin(T4);

    stroke(0, 0, 0);
    strokeWeight(10);
    line(AX, AY, int(EX), int(EY));
    line(AX, AY, int(BX), int(BY));
    line(int(EX), int(EY), int(DX), int(DY));
    line(int(BX), int(BY), int(CX1), int(CY1));
    line(int(DX), int(DY), int(CX2), int(CY2));

    stroke(0, 255, 0);
    strokeWeight(10);
    point(AX, AY);
    stroke(255, 0, 0);
    strokeWeight(10);
    point(int (BX), int (BY));
    stroke(0, 0, 0);
    strokeWeight(10);
    point(int (CX1), int (CY1));
    point(int (CX2), int (CY2));
    stroke(0, 0, 255);
    strokeWeight(10);
    point(int (DX), int (DY));
    stroke(255, 255, 0);
    strokeWeight(10);
    point(int (EX), int (EY));

    textFont(f, 10);
    fill(0);
    text(T1*180/PI, 100, 350);
    text(T2*180/PI, 400, 350);

    save("m5b" + i + "." + j +".png");
  }
}
// Succesful
```

Appendix F

Identifying Markers and Calculating Angles for Models

```
PrintWriter output;
PrintWriter r1;
PrintWriter m1;
PrintWriter r4;
PrintWriter m4;

// Inicio
  int AX = 0;
  int AY = 0;
  int BX = 0;
  int BY = 0;
  int DX = 0;
  int DY = 0;
  int EX = 0;
  int EY = 0;

  size(500, 500); // Tama o de la ventana
  background(255); // Fondo blanco
  stroke(0, 0, 0); // Color negro para las l neas
  strokeWeight(10); // Grosor de las l neas

output = createWriter("EPA.txt");
r1 = createWriter("EPA1r.txt");
m1 = createWriter("EPA1m.txt");
r4 = createWriter("EPA4r.txt");
m4 = createWriter("EPA4m.txt");

//  PImage img;
//  img = loadImage("m5b.png");
//  image(img, 0, 0);

// Comparaci n de color de pixel con referencia
  for (int l = 0; l < 360; l = l+10){
    for (int k = 0; k <360; k = k+10){
```

```
      PImage img;
      img = loadImage("m5b"+l+"."+k+".png");
      image(img, 0, 0);

  loadPixels();
  int a = millis();
  for (int i = 0; i < 500; i++) {
    for (int j = 0; j < 500; j++) {
        if ((pixels[500*j + i] == color(255, 0, 0)) ) {
          BX = i;
          BY = j;
//          updatePixels();
        }else{
        if ((pixels[500*j + i] == color(0, 255, 0)) ) {
          AX = i;
          AY = j;
//          updatePixels();
        }else{
        if ((pixels[500*j + i] == color(0, 0, 255)) ) {
          DX = i;
          DY = j;
//          updatePixels();
        }else{
        if ((pixels[500*j + i] == color(255, 255, 0)) ) {
          EX = i;
          EY = j;
//          updatePixels();
        }
        }
        }
        }
    }
  }
  int b = millis();



float theta1 = (180/PI)*atan2(AY-BY,BX-AX);
float theta2 = (180/PI)*atan2(EY-DY,DX-EX);

//println("AX " + AX);
//println("AY " + AY);
//println("BX " + BX);
//println("BY " + BY);
//println("DX " + DX);
//println("DY " + DY);
//println("EX " + EX);
//println("EY " + EY);
print("m5b"+l+"."+k+"   ");
print("theta1 " + theta1);
println("   theta2 " + theta2);
println(b-a);
output.println(l+"     "+theta1+"          "+k+"          "+theta2);
r1.println(l);
m1.println(theta1);
```

```
r4.println(k);
m4.println(theta2);
    }
  }
  output.flush(); // Writes the remaining data to the file
  output.close(); // Finishes the file
  r1.flush(); // Writes the remaining data to the file
  r1.close(); // Finishes the file
  m1.flush(); // Writes the remaining data to the file
  m1.close(); // Finishes the file
  r4.flush(); // Writes the remaining data to the file
  r4.close(); // Finishes the file
  m4.flush(); // Writes the remaining data to the file
  m4.close(); // Finishes the file
  exit(); // Stops the program

//  save("SSNM.jpg"); // Guardar imagen
```

Appendix G

Converting End-effector's Position Into Entry Angles

```
int CX = 250; // Depends on target
int CY = 200; // Depends on target
float L0 = 150;
float L1 = 200;
float L2 = 200;
float L3 = 200;
float L4 = 200;

float K1 = (CX*CX + CY*CY + L1*L1 - L2*L2)/(2*L1);
float Theta1 = (180/(2*PI))*2*atan2((CY + sqrt(CX*CX + CY*CY - K1*K1)),(CX + K1));

float m = CX - L0;
float K2 = (m*m + CY*CY + L4*L4 - L3*L3)/(2*L4);
float Theta4 = (180/(2*PI))*2*atan2((CY + sqrt(m*m + CY*CY - K2*K2)),(m + K2));

println (K1);
println (Theta1);
println (K2);
println (Theta4);
```

Appendix H

Converting Difference of Angles Into Direction of Turn

```
float diff1 = 10;
float diff4 = -10;

float turn1;
float turn4;

int giro1;
int giro4;

if diff1 < 0 {
  turn1 = 180 - diff1;
  giro1 = 1; //sentido horario
}else{
  turn1 = diff1;
  giro1 = 0; //sentido antihorario
}

if diff4 < 0 {
  turn4 = 180 - diff1;
  giro4 = 0; //sentido horario
}else{
  turn4 = diff1;
  giro4 = 1; //sentido antihorario
}
```

Bibliography

[1] Seth Hutchinson, Gregory D Hager, and Peter I Corke. A tutorial on visual servo control. *IEEE transactions on robotics and automation*, 12(5):651–670, 1996.

[2] François Chaumette and Seth Hutchinson. Visual servo control. i. basic approaches. *IEEE Robotics & Automation Magazine*, 13(4):82–90, 2006.

[3] Farrokh Janabi-Sharifi. Visual servoing: theory and applications. *Opto-Mechatronic Systems Handbook*, pages 15–1, 2002.

[4] J Chen, A Behal, D Dawson, and Y Fang. 2.5 d visual servoing with a fixed camera. In *American Control Conference, 2003. Proceedings of the 2003*, volume 4, pages 3442–3447. IEEE, 2003.

[5] Carlos Soria, Flavio Roberti, Ricardo Carelli, and José M Sebastián. Control servo-visual de un robot manipulador planar basado en pasividad. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 5(4):54–61, 2008.

[6] JOSHUA MOULTON, JAMES PELLEGRIN, and MATTHEW STEPHENSON. Planar robots and the inverse kinematic problem: An application of groebner bases.

[7] Yoshiaki Shirai and Hirochika Inoue. Guiding a robot by visual feedback in assembling tasks. *Pattern recognition*, 5(2):99IN3107–106108, 1973.

[8] Sridhar Kota. Adjustable robotic mechanism, April 28 1992. US Patent 5,107,719.

[9] Santosh Kumar Divvala. Visual servoing in unconventional environments. 2006.

[10] LEEE Weiss, ARTHURC Sanderson, and CHARLESP Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal on Robotics and Automation*, 3(5):404–417, 1987.

[11] Peter I Corke. Visual control of robot manipulators-a review. *Visual servoing*, 7:1–31, 1993.

[12] Benoit Thuilot, Philippe Martinet, Lionel Cordesses, and Jean Gallice. Position based visual servoing: keeping the object in the field of vision. In *Robotics and*

Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on, volume 2, pages 1624–1629. IEEE, 2002.

[13] Mohammad M Aref, Reza Ghabcheloo, Antti Kolu, Mika Hyvönen, Kalevi Huhtala, and Jouni Mattila. Position-based visual servoing for pallet picking by an articulated-frame-steering hydraulic mobile machine. In *2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pages 218–224. IEEE, 2013.

[14] Yacine Benbelkacem and Rosmiwati Mohd-Mokhtar. Position-based visual servoing through cartesian path-planning for a grasping task. In *Control System, Computing and Engineering (ICCSCE), 2012 IEEE International Conference on*, pages 410–415. IEEE, 2012.

[15] Alessandro De Luca, Giuseppe Oriolo, and Paolo Robuffo Giordano. Feature depth observation for image-based visual servoing: Theory and experiments. *The International Journal of Robotics Research*, 27(10):1093–1116, 2008.

[16] Aaron Mcfadyen, Marwen Jabeur, and Peter Corke. Image-based visual servoing with unknown point feature correspondence. *IEEE Robotics and Automation Letters*, 2016.

[17] Dongliang Zheng, Hesheng Wang, Jingchuan Wang, Siheng Chen, Weidong Chen, and Xinwu Liang. Image-based visual servoing of a quadrotor using virtual camera approach. *IEEE/ASME Transactions on Mechatronics*, 2016.

[18] Fujie Wang, Lulu Song, and Zhi Liu. Image-based visual servoing control for robot manipulator with actuator backlash. In *Informative and Cybernetics for Computational Social Systems (ICCSS), 2016 3rd International Conference on*, pages 272–276. IEEE, 2016.

[19] Mohammad Keshmiri and Wen-Fang Xie. Image-based visual servoing using a trajectory optimization technique. *IEEE/ASME Transactions on Mechatronics*, 2016.

[20] Luca Rosario Buonocore, Jonathan Cacace, and Vincenzo Lippiello. Hybrid visual servoing for aerial grasping with hierarchical task-priority control. In *Control and Automation (MED), 2015 23th Mediterranean Conference on*, pages 617–623. IEEE, 2015.

[21] Jenelle Armstrong Piepmeier and Harvey Lipkin. Uncalibrated eye-in-hand visual servoing. *The International Journal of Robotics Research*, 22(10-11):805–819, 2003.

[22] Hesheng Wang, Weidong Chen, and Yun-hui Liu. Dynamic eye-in-hand visual servoing with unknown target positions. In *Advances in Neural Network Research and Applications*, pages 545–550. Springer, 2010.

[23] Iraj Hassanzadeh and Hamed Jabbari. Fixed-camera visual servoing of moving object using an event-based path generation. In *Proceedings of the 10th WSEAS international conference on Systems*, pages 173–177. World Scientific and Engineering Academy and Society (WSEAS), 2006.

[24] L Enrique Sucar and Giovani Gómez. Visión computacional. *Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, México*, pages 96–98, 2011.

[25] OpenCV.org. Open cv (open source computer vision). http://www.opencv.org, 2015.

[26] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[27] Alejandro Nicolas Florencia Ysiquio. Modelado de sistemas de control de un robot manipulador basado en procesamiento digital de imagenes. 2004.

[28] Larry Chase. Tidbits & facts about the pantone, cmyk & rgb color models). https://www.lcipaper.com/kb/what-are-the-differences-between-pantone-cmyk-rgb.html, 2012.

[29] Robert L Robert L Norton, Robert L Norton, Ana Elizabeth, et al. *Diseño de maquinarias: síntesis y análisis de máquinas y mecanismos*. Number 621.01. McGraw-Hill,, 2009.

[30] Alessandro Bettini, Panadda Marayong, Samuel Lang, Allison M Okamura, and Gregory D Hager. Vision-assisted control for manipulation using virtual fixtures. *IEEE Transactions on Robotics*, 20(6):953–966, 2004.

[31] AF Brumovsky, VM Liste, and M Anigstein. Implementación de control de fuerzas en robots industriales: un caso. *IV Jornadas Argentinas de Robótica, JAR08. Córdoba*, 2006.

[32] Felipe A DeLaPeña-Contreras, Antonio Cárdenas-Galindo, Emilio J González-Galván, Guillermo DelCastillo, and Francisco E Martínez. Vision based control for industrial robots with interface on internet2.

[33] Bradley J Nelson and Pradeep K Khosla. An extendable framework for expectation-based visual servoing using environment models. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 1, pages 184–189. IEEE, 1995.

[34] John W Hill, C Rosen, et al. Machine intelligence research applied to industrial automation. *US Department of Commerce, National Technical Information Service, SRI International Tenth Report*, 1980.

[35] Gerald J Agin. Real time control of a robot with a mobile camera. In *9th Int. Symp. on Industrial Robots*, pages 233–246, 1979.

[36] RE Prajoux. Visual tracking. *Machine intelligence research applied to industrial automation*, pages 17–37, 1979.

[37] S Venkatesan and C Archibald. Realtime tracking in five degrees of freedom using two writs-mounted laser range finders. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pages 2004–2010. IEEE, 1990.

[38] Lee Elliot Weiss, Arthur C Sanderson, and CP Neuman. Dynamic visual servo control of robots: an adaptive image-based approach. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, pages 662–668. IEEE, 1985.

[39] PY Coulon and M Nougaret. Use of a tv camera system in closed-loop position control of mechanisms. In *Robot Vision*, pages 117–127. Springer, 1983.

[40] TE Weber and RL Hollis. A vision based correlator to actively damp vibrations of a coarse-fine manipulator. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pages 818–825. IEEE, 1989.

[41] Ernst Dieter Dickmanns and Volker Graefe. Applications of dynamic monocular machine vision. *Machine vision and Applications*, 1(4):241–261, 1988.

[42] Joseph S Yuan, Richard A MacDonald, and Felix HN Keung. Telerobotic tracker, July 17 1990. US Patent 4,942,538.

[43] Nikolaos P Papanikolopoulos and Pradeep K Khosla. Shared and traded telerobotic visual control. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pages 878–885. IEEE, 1992.

[44] Frank Tendick, Jonathan Voichick, Gregory Tharp, and Lawrence Stark. A supervisory telerobotic control system using model-based vision feedback. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 2280–2285. IEEE, 1991.

[45] François Chaumette and Seth Hutchinson. Visual servo control, part ii: Advanced approaches. *IEEE Robotics and Automation Magazine*, 14(1):109–118, 2007.

[46] Francois Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In *The confluence of vision and control*, pages 66–78. Springer, 1998.

[47] Processing.org. Processing. http://www.processing.org, 2001.

[48] Q. HiTech. Qhm495lm web camera. http://www.qhmpl.com/QHM495LM-WebCamera.html, 2014.

[49] guibot. Control your motors with l293d and arduino. http://www.instructables.com/id/Control-your-motors-with-L293D-and-Arduino/, 2014.

[50] Luz Alejandra Magre and Juan Carlos Martínez Santos. Using close range photogrammetric method to estimate kinetic variables in olympic-style weightlifting. In *Engineering Mechatronics and Automation (CIIMA), 2014 III International Congress of*, pages 1–4. IEEE, 2014.

[51] Ian L Weatherall and Bernard D Coombs. Skin color measurements in terms of cielab color space values. *Journal of investigative dermatology*, 99(4):468–473, 1992.

[52] Nathan Moroney, Mark D Fairchild, Robert WG Hunt, Changjun Li, M Ronnier Luo, and Todd Newman. The ciecam02 color appearance model. In *Color and Imaging Conference*, volume 2002, pages 23–27. Society for Imaging Science and Technology, 2002.

[53] S. Navarro-Mora and J. C. Martnez-Santos. Visual servoing of a five-bar linkage mechanism. In *2015 IEEE 2nd Colombian Conference on Automatic Control (CCAC)*, pages 1–5, Oct 2015.

[54] Koichi Hashimoto, Tsutomu Kimoto, Takumi Ebine, and Hidenori Kimura. Manipulator control with image-based visual servo. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 2267–2271. IEEE, 1991.

[55] Ricardo Corey Blackett. *Optimal Synthesis of Planar Five-link Mechanisms for the Production of Nonlinear Mechanical Advantage*. PhD thesis, Citeseer, 2001.

[56] Patrick Sebastian, Yap Vooi Voon, and Richard Comley. Colour space effect on tracking in video surveillance. *International Journal on Electrical Engineering and Informatics*, 2(4):298, 2010.

[57] Benjamin D Zarit, Boaz J Super, and Francis KH Quek. Comparison of five color models in skin pixel classification. In *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 1999. Proceedings. International Workshop on*, pages 58–63. IEEE, 1999.

[58] Rita Cucchiara, Costantino Grana, Massimo Piccardi, Andrea Prati, and Stefano Sirotti. Improving shadow suppression in moving object detection with hsv color information. In *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pages 334–339. IEEE, 2001.