

UNIVERSIDAD CATÓLICA DE SANTA MARÍA

FACULTAD DE CIENCIAS E INGENIERÍAS FÍSICAS Y FORMALES

PROGRAMA PROFESIONAL DE INGENIERÍA MECÁNICA, MECÁNICA

ELÉCTRICA Y MECATRÓNICA



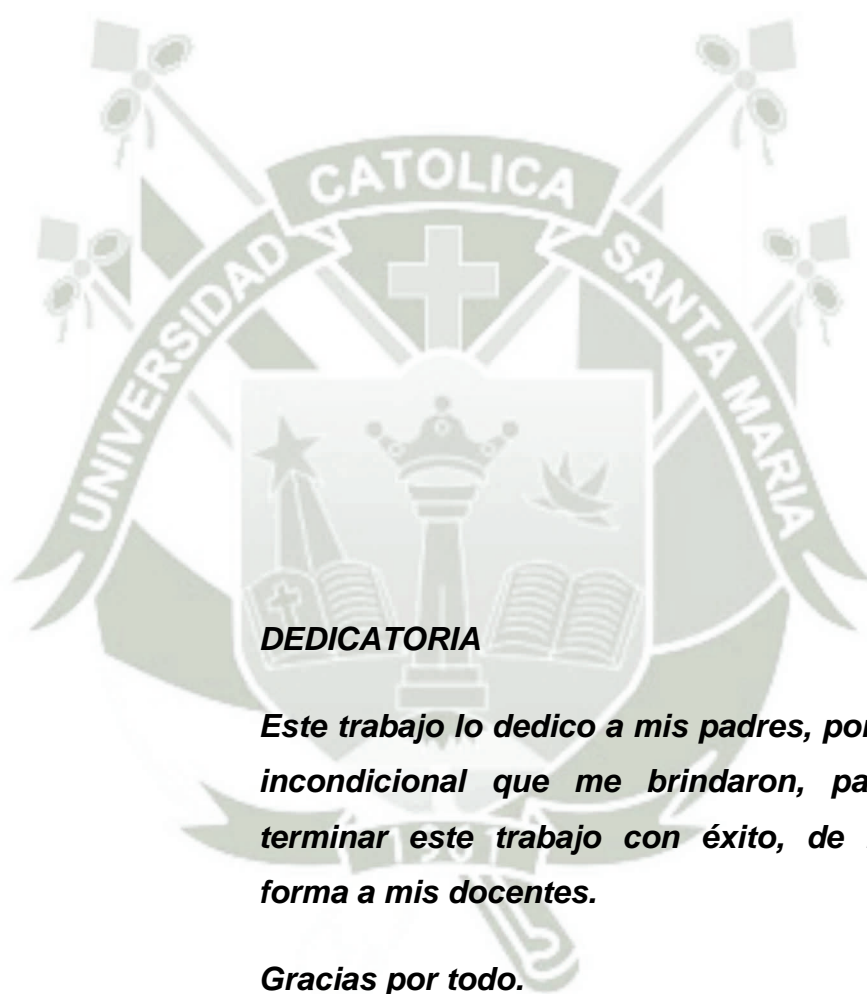
“DISEÑO DE UN SISTEMA DE IDENTIFICACIÓN Y CLASIFICACIÓN POR VISIÓN ARTIFICIAL”.

Tesis presentada por el Bachiller:
VICTOR MANUEL SANCHEZ CACERES

Para optar el Título Profesional de
INGENIERO MECATRONICO

AREQUIPA-PERÚ

2013



DEDICATORIA

Este trabajo lo dedico a mis padres, por el apoyo incondicional que me brindaron, para lograr terminar este trabajo con éxito, de la misma forma a mis docentes.

Gracias por todo.

Víctor Sánchez

INDICE

RESUMEN.....	xii
ABSTRACT	xiv
CAPITULO I	1
MARCO METODOLOGICO.....	1
1.1. INTRODUCCIÓN.....	1
1.2. JUSTIFICACIÓN.....	3
1.3. ANTECEDENTES	3
1.4. OBJETIVOS	4
1.5. HIPÓTESIS	4
1.6. IDENTIFICACIÓN DEL PROBLEMA.....	5
1.7. ESQUEMA.....	5
1.8. ALCANCES Y LIMITACIONES	6
CAPITULO II	8
MARCO TEÓRICO.....	8
2.1. VISIÓN ARTIFICIAL.....	8
2.1.1. Introducción.....	8
2.1.2. Origen de la visión.....	8
2.1.3. Visión humana	9
2.1.4. Anatomía de la visión humana	9
2.1.5. Que es la visión artificial.....	13
2.1.6. Generación de posibles soluciones.....	16
2.1.7. Procesamiento de Imágenes	17
2.2. INSTRUMENTACIÓN (CÁMARA WEB).....	50
2.2.1. Definición de cámara web.....	51
2.3. MICROPROCESADOR (PLACA ARDUINO).....	62

2.3.1. Definición Arduino Uno	62
2.4. CONTROL POR PC (LABVIEW)	67
2.4.1. Labview	67
2.4.2. Servomotor Tower Pro sg 5010	86
CAPITULO III	94
MARCO EXPERIMENTAL.....	94
3.1. ADQUISICIÓN DE IMÁGENES.....	94
3.2. DISEÑO DEL PROGRAMA DE CLASIFICACION.....	103
3.2.1. Clasificación por Figuras Geométricas	103
3.2.1.1. Detección de la figura geométrica (cuadrado).....	105
3.2.1.2. Detección de la figura geométrica (circulo).....	108
3.2.1.3. Detección de la figura geométrica (Triangulo).....	110
3.2.1.4. Control de PWM.....	120
3.2.2. Clasificación por Colores.....	125
3.2.2.1. Localización del color rojo.....	132
3.2.2.2. Localización del color verde.....	134
3.2.2.3. Localización del color azul	136
3.2.2.4. Control de PWM.....	141
3.3. ILUMINACIÓN.....	147
3.4. MECANISMOS.....	150
CAPITULO IV	156
PUESTA EN MARCHA.....	156
4.1. INICIACIÓN DEL PROGRAMA DE VISIÓN ARTIFICIAL.....	156
CONCLUSIONES.....	161
RECOMENDACIONES	162
BIBLIOGRAFIA.....	163

ÍNDICE DE FIGURAS

Figura 1. 1: Esquema del proyecto de visión artificial	5
Figura 1. 2: Diagrama de bloques del sistema a Implementar	6
Figura 2. 1: Espectro Electromagnético	9
Figura 2. 2: Partes del Ojo	9
Figura 2. 3: Percepción de Objetos	11
Figura 2. 4: Conos y bastones.....	12
Figura 2. 5: Sensibilidad Espectral humana al color.....	13
Figura 2. 6: Diseño de visión artificial.....	19
Figura 2. 7: Algoritmos de análisis de imagen.....	24
Figura 2. 8: Mascara de convolución.....	31
Figura 2. 9: Ejemplo mascara de convolucion.....	31
Figura 2. 10: Dos mascaras de 3x3 para realizar un filtrado pasa bajos (suavizado, promediado).....	32
Figura 2. 11: Resultados de suavizar con máscaras cuadradas	34
Figura 2. 12: Filtro de mediana.....	36
Figura 2. 13: Rectángulo blanco.....	42
Figura 2. 14: Filtros Notch	45
Figura 2. 15: Filtro Butterworth	48
Figura 2. 16: Funcionamiento interno.....	52
Figura 2. 17: Conectores	53
Figura 2. 18: Cámara web Gigaware.....	54
Figura 2. 19: Representación de un Pixel	55
Figura 2. 20: Pixeles extraídos de una imagen	55
Figura 2. 21: Modelo de color RYB	58
Figura 2. 22: Modelo de color RGB	59
Figura 2. 23: Modelo de color CMY	60
Figura 2. 24: Triángulo de color HSI.....	61
Figura 2. 25: Modelo HSV	62
Figura 2. 26: Placa Arduino Uno	64
Figura 2. 27: Placa Arduino y Cable USB	65

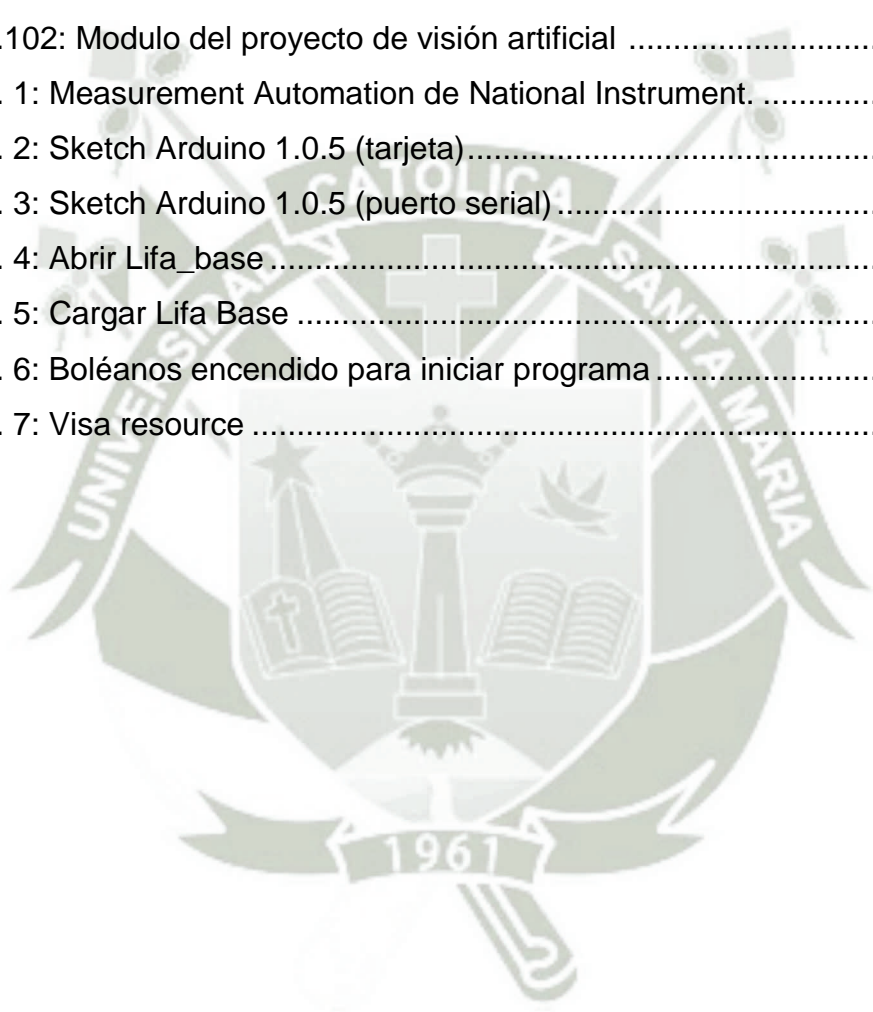
Figura 2. 28: Placa Arduino UNO alimentación	65
Figura 2. 29: Placa Arduino UNO (señales digitales)	66
Figura 2. 30: Logo de LabVIEW	70
Figura 2. 31: Panel frontal	71
Figura 2. 32: Diagrama de bloques	72
Figura 2. 33: Paletas	73
Figura 2. 34: Menú de funciones Vision and Motion.....	74
Figura 2. 35: Menú NI-IMAQ	75
Figura 2. 36: Menú Vision Utilities	75
Figura 2. 37: Menú Image Processing.....	76
Figura 2. 38: Menú Machine Vision	76
Figura 2. 39: Menú NI-IMAQdx.....	77
Figura 2. 40: Menú Visión Express.....	78
Figura 2. 41: Menú Image Management	78
Figura 2. 42: Imaq Create	79
Figura 2. 43: Bordes de imagen	79
Figura 2. 44: Image type	80
Figura 2. 45: Build Array bloques	83
Figura 2. 46: Build array	83
Figura 2. 47: Ejemplo de array	84
Figura 2. 48: Ejemplo Build array	85
Figura 2. 49: Ejemplo Build array	86
Figura 2. 50: Servomotor.....	87
Figura 2. 51: Parte interna servomotores	88
Figura 2. 52: Señales para el servomotor	89
Figura 2. 53: Señales para el servomotor	89
Figura 2. 54: Servomotor en modelismo	90
Figura 2. 55: Partes del servomotor	91
Figura 2. 56: Señal de pulsos.....	92
Figura 2. 57: Especificación servo motor Tower pro	93
Figura 3. 1: IMAQdx Open Camera.vi	95
Figura 3. 2: IMAQdx Configure Grab .vi	95

Figura 3. 3: IMAQdx Grab.vi.....	96
Figura 3. 4: IMAQdx Close Camera.vi.....	96
Figura 3. 5: Diagrama de bloques	96
Figura 3. 6: Panel frontal	97
Figura 3. 7: Vision Express.....	97
Figura 3. 8: Vision Acquisition	98
Figura 3. 9: Select Acquisition Type	99
Figura 3. 10: Configure Acquisition Settings	99
Figura 3. 11: Select Controls/Indicators	100
Figura 3. 12: Diagrama de bloques	100
Figura 3. 13: Panel frontal	101
Figura 3. 14: Imaq Create	102
Figura 3. 15: Imaq Extract Single Color Plane.....	102
Figura 3. 16: Conexión de imaq créate, visión adquisition y Extract Single Color Plane.....	103
Figura 3. 17: Vision Assistant.....	104
Figura 3. 18: Entorno de vision assistant	105
Figura 3. 19: Template	105
Figura 3. 20: Define Curves.....	106
Figura 3. 21: Customize Scoring	107
Figura 3. 22: Specify Match Options	107
Figura 3. 23: Template Circulo	108
Figura 3. 24: Define Curves.....	109
Figura 3. 25: Customize Scoring	109
Figura 3. 26: Specify Match options	110
Figura 3. 27: Template tirangulo.....	111
Figura 3. 28: Define Curves.....	111
Figura 3. 29: Customize Scoring	112
Figura 3. 30: Specify Match options	112
Figura 3. 31: Asistente de visión	113
Figura 3. 32: Select Controls	114
Figura 3. 33: Array to cluster	114

Figura 3. 34: Unbundle by name	115
Figura 3. 35: Conexión	115
Figura 3. 36: Unbundle by name	116
Figura 3. 37: Round to nearest.....	116
Figura 3. 38: Greater or Equal.....	116
Figura 3. 39: Build Array.....	117
Figura 3. 40: Imaq overlay rectangle:	117
Figura 3. 41: Imaq overlay text:	117
Figura 3. 42: Conexión	118
Figura 3. 43: Flip Flop	119
Figura 3. 44: Diagrama de los flip flop en el panel frontal.	120
Figura 3. 45: Visa resource	120
Figura 3. 46: Init	121
Figura 3. 47: Set Number of Servos	121
Figura 3. 48: Configure Servo	121
Figura 3. 49: Servo Write Angle	122
Figura 3. 50: Servo read Angle.....	122
Figura 3. 51: Close	122
Figura 3. 52: Wait	123
Figura 3. 53: Increment	123
Figura 3. 54: Equal.....	123
Figura 3. 55: Select	123
Figura 3. 56: Diagrama de bloques para escribir los ángulos para el servomotor	124
Figura 3. 57: Diagrama de bloques completo.....	125
Figura 3. 58: Vision express	125
Figura 3. 59: Select acquisition source.....	126
Figura 3. 60: Select acquisition type.....	127
Figura 3. 61: Configure acquisition settings	127
Figura 3. 62: Select controls/indicator	128
Figura 3. 63: Diagrama de bloques de adquisicion de imagenes	129
Figura 3. 64: Panel frontal adquisicion de imagenes.....	129

Figura 3. 65: Imaq create	130
Figura 3. 66: Imaq set color Pixe line 2	130
Figura 3. 67: Visión assistant de localización de colores	131
Figura 3. 68: Color location	131
Figura 3. 69: Ni vision assistant para colores	132
Figura 3. 70: Create template (color rojo).....	132
Figura 3. 71: Select template in the image (color rojo).....	133
Figura 3. 72: Color detectado (color rojo).....	134
Figura 3. 73: Créate template (color verde).....	134
Figura 3. 74: Select template in the image (color verde)	135
Figura 3. 75: Color detectado (verde).....	135
Figura 3. 76: Create template (color azul)	136
Figura 3. 77: Select template in the image (color azul)	137
Figura 3. 78: Color detectado (azul).....	137
Figura 3. 79: Imaq overlay rectangle	138
Figura 3. 80: Conexión del asistente de visión con el imaq overlay rectangle y el imaq overlay text.....	139
Figura 3. 81: Flip flip rs.....	140
Figura 3. 82: Booleanos del flip flop	141
Figura 3. 83: Visa resource	141
Figura 3. 84: Init	142
Figura 3. 85: Set Number of Servos	142
Figura 3. 86: Configure Servo	142
Figura 3. 87: Servo Write Angle	143
Figura 3. 88: Servo read Angle:.....	143
Figura 3. 89: Close	143
Figura 3. 90: Wait	144
Figura 3. 91: Increment	144
Figura 3. 92: Equal	144
Figura 3. 93: Select	144
Figura 3. 94: Diagrama de bloques para escribir los ángulos que debe moverse el servomotor de acuerdo al color localizado.....	146

Figura 3. 95: Diagrama de bloques para el movimiento del servomotor.	147
Figura 3. 96: Iluminación posterior	148
Figura 3. 97: Iluminación Frontal Oblicua	148
Figura 3. 98: Otros tipos de iluminación	149
Figura 3. 99: Fuentes de luz.....	150
Figura 3. 100: Figuras geometricas.....	154
Figura 3. 101: Planos de conexión: Placa Arduino, protoboard, fuente de poder y servomotor	154
Figura 3.102: Modulo del proyecto de visión artificial	155
Figura 4. 1: Measurement Automation de National Instrument.	156
Figura 4. 2: Sketch Arduino 1.0.5 (tarjeta).....	157
Figura 4. 3: Sketch Arduino 1.0.5 (puerto serial)	158
Figura 4. 4: Abrir Lifa_base	159
Figura 4. 5: Cargar Lifa Base	159
Figura 4. 6: Boléanos encendido para iniciar programa	160
Figura 4. 7: Visa resource	160



ÍNDICE DE TABLAS

Tabla 2.1: Relación entre frecuencias y colores percibidos	57
Tabla 2.2: Especificación servo motor Tower pro.....	93



RESUMEN

El presente proyecto es un sistema automatizado controlado por visión artificial el cual va a identificar y clasificar diferentes objetos (figuras geométricas). Utilizando hardware (control de servomotor, tarjeta arduino, etc.), el cual se va a poder observar en una demostración, mientras el tema fuerte del proyecto es el desarrollo de la programación utilizando el software (control por Labview 2011).

El principio de este proyecto es transportar objetos de diferentes colores (rojo, verde, azul) o formas (triángulo, círculo, cuadrado), por una faja transportadora, y al momento que va ser detectado, la cámara web captará la imagen y la procesará en Labview, para clasificarlos en diferentes contenedores dependiendo del color de cada objeto o de la forma de los mismos. A la vez en la pantalla de dicho programa se visualizará el nombre del objeto que se capturo en la cámara web, por ejemplo aparecerá el nombre a quien corresponda, si en caso sea color: rojo, verde, azul o sea forma: un triángulo, círculo, cuadrado.

En el capítulo 1, se hace una introducción a visión artificial, se describe la justificación de la problemática del proyecto a nivel estudiantil y en la industria, se trazan los objetivos que se tiene que cumplir. Se identifica el problema principal del proyecto de visión, y se indican los alcances y limitaciones.

En el capítulo 2, se profundiza el tema de visión artificial, se detalla el instrumento y el microprocesador a utilizar (cámara web y placa arduino uno), se describen las características y bondades del software Labview y su plataforma de Adquisición de Imágenes, con una breve descripción de los bloques a utilizar y el control del servomotor tower pro sg5010 (ancho de pulso PWM).

El capítulo 3, destinado a analizar el diseño e implementación de la solución desarrollada, se detalla los diagramas virtuales generados, se hace una

descripción de cada una de las fases del sistema, desde su configuración, calibración, y sincronización con las herramientas externas.

En el capítulo 4, se muestran los pasos a seguir para la puesta en marcha o la iniciación del programa de visión artificial según lo desarrollado en la programación.

Al final se presentan las conclusiones, recomendaciones y bibliografía como referencia para futuras investigaciones



ABSTRACT

This project is an artificial vision which will identify and classify different objects (geometric figures) controlled automated system. Using hardware (servo motor control, arduino card, etc.), Which will be able to observe a demonstration while the strong theme of the project is the development of programming using the software (control by Labview 2011).

The principle of this project is to carry objects of different colors (red, green, blue) or shapes (triangle, circle, square) on a conveyor belt , and when it will be detected, capture the webcam image and the processed Labview in order to classify into different containers depending on the color of each object or the shape thereof. While in the program screen the name of the object captured in the webcam was displayed, for example the name to the appropriate appears, if in case it is color: red, green, blue or is shape: a triangle, circle, square.

In Chapter 1, an introduction to computer vision is the justification for the project issues described student-level and industry, the objectives that must be met are plotted. The main problem of the proposed vision is identified, and the scope and limitations are indicated.

In Chapter 2, the issue of artificial vision deepens, the instrument used and the microprocessor (webcam and arduino card) detailing the characteristics and benefits of Labview software platform is described and Image Acquisition with a brief description of the blocks to use and control tower pro SG5010 servo (PWM pulse width).

Chapter 3, designed to analyze the design and implementation of the developed solution, the generated virtual detailed diagrams, a description of each of the phases of the system from setup, calibration, and synchronization with external tools is made.

In Chapter 4, the steps for starting or program initiation as machine vision developed in the programming is.

Finally, conclusions, recommendations and literature as reference for future research are presented.



CAPITULO I

MARCO METODOLOGICO

1.1. INTRODUCCIÓN

Se puede definir la “Visión Artificial” como un campo de la “Inteligencia Artificial” que, mediante la utilización de las técnicas adecuadas, permite la obtención, procesamiento y análisis de cualquier tipo de información especial obtenida a través de imágenes digitales.

La visión artificial la componen un conjunto de procesos destinados a realizar el análisis de imágenes. Estos procesos son: captación de imágenes, memorización de la información, procesamiento e interpretación de los resultados.

Con la visión artificial se pueden:

- Automatizar tareas repetitivas de inspección realizadas por operadores.
- Realizar controles de calidad de productos que no era posible verificar por métodos tradicionales. Realizar inspecciones de objetos sin contacto físico.
- Realizar la inspección del 100% de la producción (calidad total) a gran velocidad.
- Reducir el tiempo de ciclo en procesos automatizados.
- Realizar inspecciones en procesos donde existe diversidad de piezas con cambios frecuentes de producción.

Las principales aplicaciones de la visión artificial en la industria actual son:

- Identificación e inspección de objetos.
- Determinación de la posición de los objetos en el espacio.
- Establecimiento de relaciones espaciales entre varios objetos (guiado de robots)
- Determinación de las coordenadas importantes de un objeto.
- Realización de mediciones angulares.
- Mediciones tridimensionales.

Métodos de captación de las imágenes.

Digital. La función obtenida tras el resultado de la medida o muestreos realizados a intervalos de tiempo espaciados, siendo el valor de dicha función un número positivo y entero. Los valores que esta función toma en cada punto dependen del brillo que presenta en esos puntos la imagen original.

Píxel. Una imagen digital se considera como una cuadrícula. Cada elemento de esa cuadrícula se llama Píxel (Picture element). La resolución estándar de una imagen digital se puede considerar de 512x484 Pixel.

Nivel de grises. Cuando una imagen es digitalizada, la intensidad del brillo en la escena original correspondiente a cada punto es cuantificada, dando lugar a un número denominado “nivel de gris”.

Imagen binaria. Es aquella que sólo tiene dos niveles de gris: negro y blanco. Cada píxel se convierte en negro o blanco en función del llamado nivel binario

1.2. JUSTIFICACIÓN

En el Programa Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatronica no se han realizado trabajos utilizando Visión artificial, este proyecto ayuda al desarrollo de la programación para la identificación y clasificación tanto por color o de la forma de las figuras geométricas.

Este proyecto contara con una demostración física para certificar que los algoritmos utilizados son correctos y así ayudar, a incentivar a los alumnos que se interesen más por la investigación y pueda ser utilizado como material de consulta, ya que llevamos cursos de Procesamiento digital de señales, Procesos Inteligentes, etc.

Las inspecciones realizadas por los seres humanos, a menudo no pueden cumplir con los requisitos de la industria moderna respecto a la velocidad de producción, calidad de producto y costos de producción. Los humanos se cansan, cometen errores y los criterios que se aplican durante las inspecciones son inevitablemente subjetivos. En algunos casos, no es humanamente posible llevar a cabo las tareas de inspección debido a las condiciones ambientales. Las cámaras y el resto de componentes de un sistema de visión artificial, por el contrario, llevan a cabo las mediciones con una precisión constante y a un ritmo que es establecido por el propio proceso de producción. Estas ventajas han llevado a una creciente aceptación de la visión por industrias de todo el mundo.

1.3. ANTECEDENTES

Tesis para obtener grado de magister en la “PONTIFICIA UNIVERSIDAD CATOLICA DEL PERU” realizado por el Ing. EDDIE ANGEL SOBRADO MALPARTIDA

Tema: Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot. Lima-Perú

Fuente: Pagina web de tesis de la Universidad Pontificia Católica del Perú.

Esta tesis se enfoca en el reconocimiento y manipulación de objetos mediante la utilización de un brazo robótico, utilizando el principio de redes neuronales y el software Matlab.

1.4. OBJETIVOS

Objetivos Primarios

- Diseñar un sistema de identificación y clasificación por visión artificial

Objetivos secundarios

- Implementar, adecuar algoritmos y diagrama de bloques para realizar un correcto control que permita comparar patrones para la identificación y clasificación de objetos.
- Implementar un prototipo de visión artificial en el cual se pueda visualizar los objetos mediante el software Labview y a su vez identificarlos y clasificarlos mediante controladores.

1.5. HIPÓTESIS

Es probable realizar la identificación y clasificación de objetos utilizando el software labview con su asistente de adquisición de imágenes y así demostrar que mediante la utilización de filtros y comparación de patrones utilizando imágenes predeterminadas con imágenes recién captadas vía cámara web y a su vez desarrollando una programación con diagramas de bloques de visión es posible la identificación de diferentes objetos por forma o color.

1.6. IDENTIFICACIÓN DEL PROBLEMA

Problema

Los problemas principales para el desarrollo de este proyecto, son:

La identificación de los algoritmos, el desarrollo del asistente de visión artificial de labview, reconocimiento imágenes utilizando el software labview 2011, la interface de comunicación y el movimiento del equipo de transporte y clasificador, para la identificación y la clasificación de las figuras geométricas

1.7. ESQUEMA

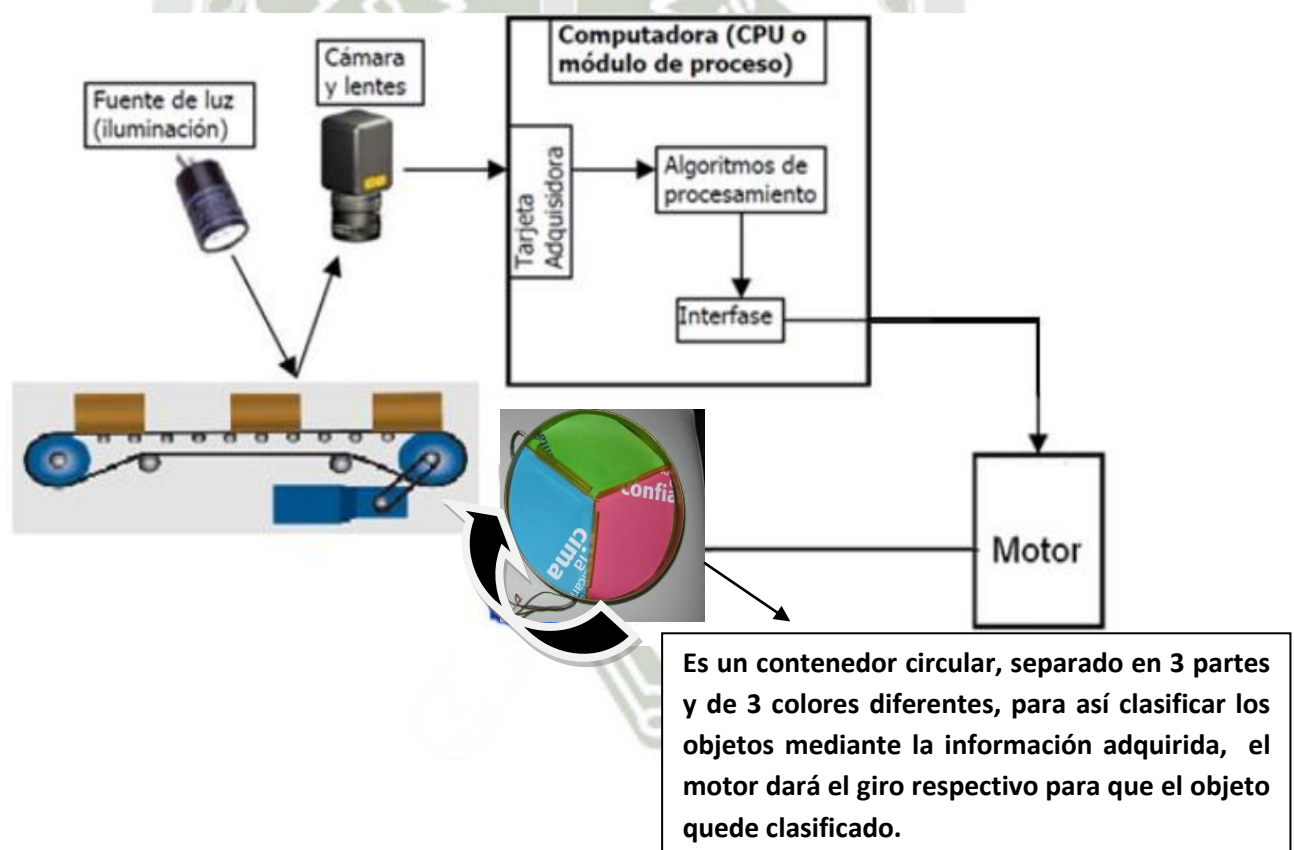


Figura 1. 1: Esquema del proyecto de visión artificial
Fuente: Elaboración propia

Diagrama de bloques

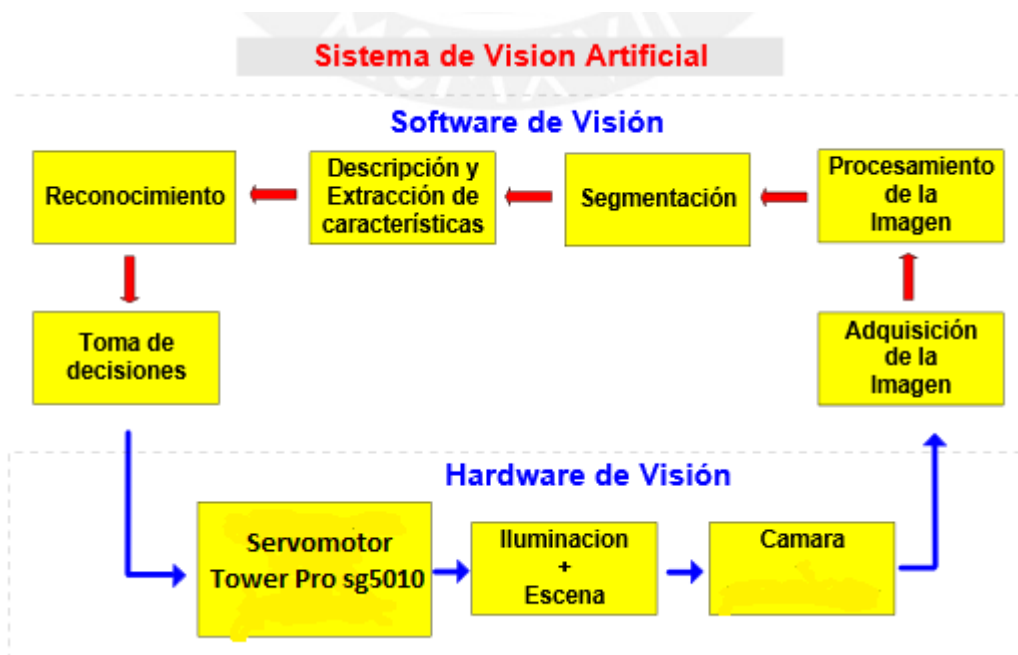


Figura 1. 2: Diagrama de bloques del sistema a Implementar
Fuente: Elaboración propia

1.8. ALCANCES Y LIMITACIONES

ALCANCES

- El desarrollo neto de este proyecto es la investigación del lenguaje de programación utilizando visión artificial. Realizar un algoritmo el cual me permita identificar, visualizar el nombre del objeto captado y clasificarlo tanto por su color o forma. El proyecto va a demostrar que los algoritmos, patrones, parámetros, comparaciones realizadas son correctas.
- Se contara con una faja transportadora, la cual transportara los objetos hasta su posición final de un clasificador dependiendo de las características del objeto.

- Sincronizar recipientes, con la faja transportadora según la velocidad y distancias del proceso de visión artificial.

LIMITACIONES

- Solo idéntica objetos si se realiza el proceso en las mismas condiciones que se tomaron los patrones.
- Identifica y clasifica figuras geométricas, cuando se transporte a bajas velocidades.
- Se utiliza figuras geométricas pequeñas, debido que la plataforma de clasificación es de tamaño reducido.
- El servomotor solo gira 180°.
- Posibles errores en cambios de iluminación, siempre y cuando se mantenga un mismo ambiente.
- Mantener el mismo fondo en la captación con cámara web.

CAPITULO II

MARCO TEÓRICO

2.1. VISIÓN ARTIFICIAL

2.1.1. Introducción

Dentro de la Mecatrónica existe un área que se está desarrollando cada vez más, que viene hacer la Visión artificial. Esta área se ocupa de simular el comportamiento del ojo del ser humano para poder desarrollar proyecto que involucren un alto grado de precisión visual.

Por otra parte, la visión artificial se puede aplicar en diferentes campos como son el médico, el electrónico, mecánico, etc. Cada uno de estos campos requiere una percepción visual artificial con un alto grado de precisión en virtud que si se presentan errores, esto no permitirá que el sistema tome acciones correctas de acuerdo a lo que ha captado.

2.1.2. Origen de la visión

Gracias a la existencia de moléculas sensibles a la luz, los seres vivos se han adaptado y han desarrollado mecanismos para captar información de su entorno haciendo uso de éstas. Es así como diversos animales desarrollaron células detectoras de luz por todo su organismo o en puntos concretos. A partir de aquí comienza la evolución de dichas moléculas hasta crear lo que hoy conocemos como el ojo, en el que las superficies puntuales de las células receptoras de luz acabaron cerrándose en una cavidad interior abierta al exterior por una pequeña apertura, desdoblándose así una lente capaz de enfocar en la superficie fotosensible la luz que reflejarían los objetos del medio.

2.1.3. Visión humana

Las partes del ojo son esenciales para la existencia humana porque gracias a ellas captamos y percibimos nuestro entorno. El sistema de Visión humana es capaz de detectar ondas dentro del rango del espectro electromagnético que van aproximadamente desde los 400 nm (color violeta) hasta los 700 nm (color rojo), percibiendo este rango en un continuo gradiente de colores conocido como espectro visual tal como se aprecia en la Figura 2.1.

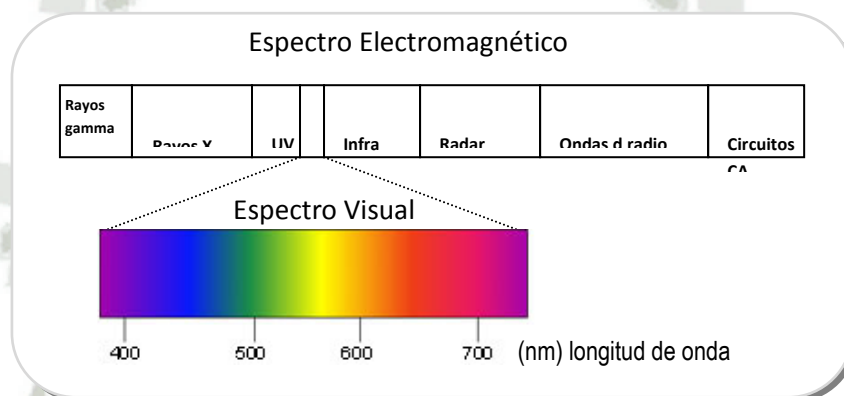


Figura 2. 1: Espectro Electromagnético

Fuente: Frenzel, Louis L. (mayo de 2003). Sistemas electrónicos de comunicaciones

2.1.4. Anatomía de la visión humana

En la Figura 2.2. se aprecian los componentes principales del ojo:

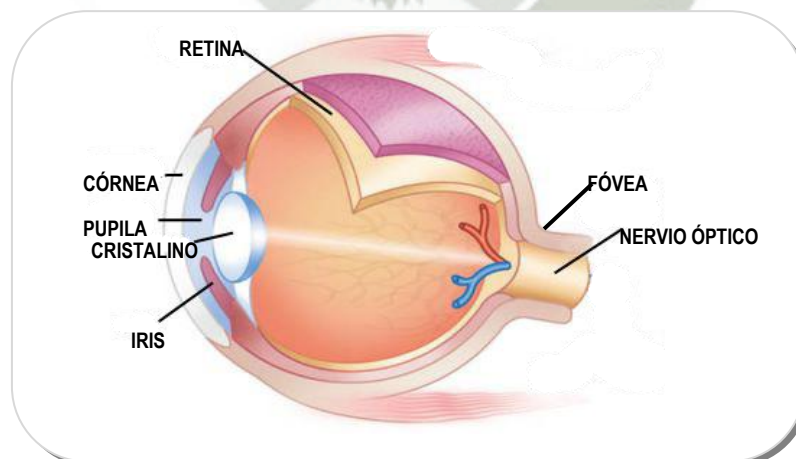


Figura 2. 2: Partes del Ojo

Fuente: wikipedia, Anatomía de la visión humana

- Esclerótica: es la parte blanca que se ve en nuestros ojos. Su propósito es servir como estructura, soporte y protección al ojo, como un cascarón.
- Córnea: proporciona gran parte del enfoque a la luz que entra en nuestros ojos. Se compone de 5 capas, siendo la más superficial el epitelio corneal que es quien protege al ojo de agentes externos y está formado por células altamente regenerativas.
- Pupila: controla la cantidad de luz que entra al ojo. En presencia de mucha luz la pupila se cierra, mientras que con poca luz se dilata, aumentando su tamaño.
- Iris: es la parte que da color a nuestros ojos. Su función es controlar el tamaño de la pupila. Esto lo logra contrayendo o expandiendo sus músculos.
- Cristalino: da el toque fino al enfoque. Realiza su tarea cambiando su forma, haciéndose más gruesa o delgada según sea necesario.
- Retina: es aquí donde se realiza la percepción de la imagen, es una finísima capa de tejido nervioso que cubre la pared interna del ojo y capta las imágenes. Cuando la visión es adecuada, los rayos de luz se enfocan exactamente en esta parte del ojo.
- Fóvea: es una pequeña depresión en la retina cuya área total es un poco mayor a 1 mm².
- Nervio óptico: transmite al cerebro las imágenes que se captan en la retina en forma de impulsos eléctricos.
- Humor vitrio: Es una sustancia transparente que se encuentra en la cavidad del globo ocular. Su función es darle estructura al ojo, como si fuera su esqueleto.

Lejos de ahondar en la anatomía y fisiología del ojo humano, mencionaremos que el proceso de captura de imágenes se inicia en la retina, aquí existen dos tipos de células neuronales sensibles a la luz: los bastones y los conos, ambos con propiedades totalmente distintas. La información captada por dichos fotorreceptores provoca complejas reacciones químicas que conducen a una hiperpolarización produciendo una señal nerviosa que se transmite a las células ganglionales, donde se interpreta y se manda al cerebro a través del nervio óptico.

Para distinguir las ondas de color, el ojo recoge la luz y la enfoca en su superficie posterior, en la Figura 2.3 vemos cómo se realiza la percepción de un objeto situado en un lugar lejano y otro cercano.

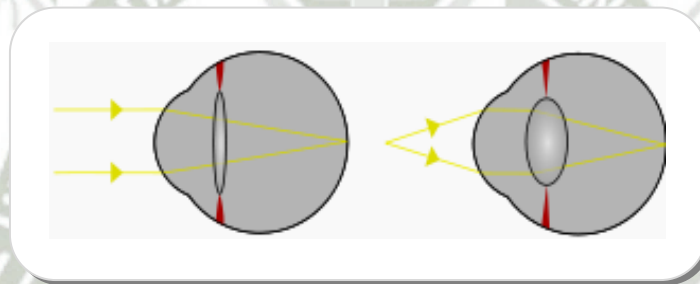


Figura 2. 3: Percepción de Objetos
Fuente: Wikipedia, la enciclopedia libre

En el fondo del ojo, los conos se concentran en la fóvea. Éstos son los responsables de la definición espacial y de la visión del color; debido a que son poco sensibles a la intensidad de la luz proporcionan visión fotópica. Por el contrario, los bastones se concentran en zonas alejadas de la fóvea y son los responsables de la visión escotópica, sensible a alta intensidad de luz, en realidad no son sensibles al color. Los bastones son mucho más sensibles que los conos a la intensidad luminosa, por lo que aportan a la visión del color aspectos como el brillo y el tono. En la Figura 2.4 se muestran los conos y bastones. Las células amarillas son los conos, mientras que las células blancas (en la parte inferior de la fotografía) son los bastones.

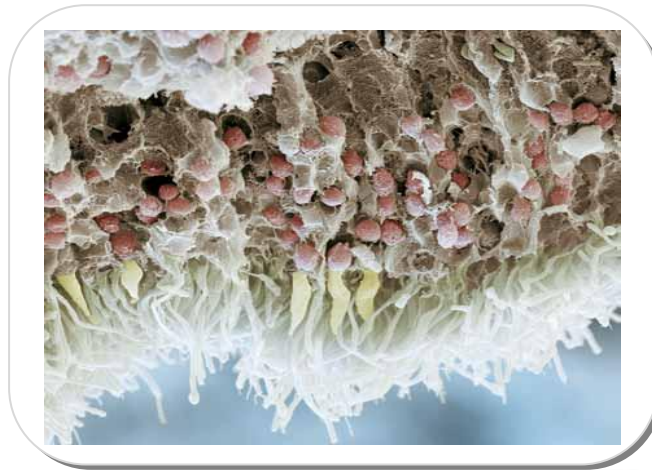


Figura 2. 4: Conos y bastones
Fuente: El mundo salud

La visión está relacionada con la percepción del color, la forma, la distancia y las imágenes en 3 dimensiones. El proceso de generación de imágenes inicia cuando las ondas luminosas inciden sobre la retina del ojo, como lo habíamos denotado anteriormente, si estas ondas son superiores o inferiores a determinados límites, no producen impresión visual, el color depende de la longitud de onda incidente. La imagen percibida por la retina es una imagen invertida, la proximidad aparente es resultado de una asociación inconsciente con el incremento en la magnitud de la imagen retiniana.

Es así como el proceso visual resultante no es más que la combinación de la percepción de cada uno de nuestros dos ojos sobre un mismo objeto, el cual muestra una imagen desde dos ángulos ligeramente diferentes que los ojos unen para formar una imagen tridimensional única, este tipo de visión, llamada visión estereoscópica, es importante ya que permite la percepción de la profundidad.

El color puede ser modelado mediante distribuciones gaussianas como podemos apreciar en la Figura 2.5. En el hombre, los máximos de absorción de los tres receptores del color son 426, 530 y 560 nm para el azul, verde y rojo respectivamente. La longitud de onda de absorción máxima de la rodopsina, la molécula de los bastones, es de 500 nm.

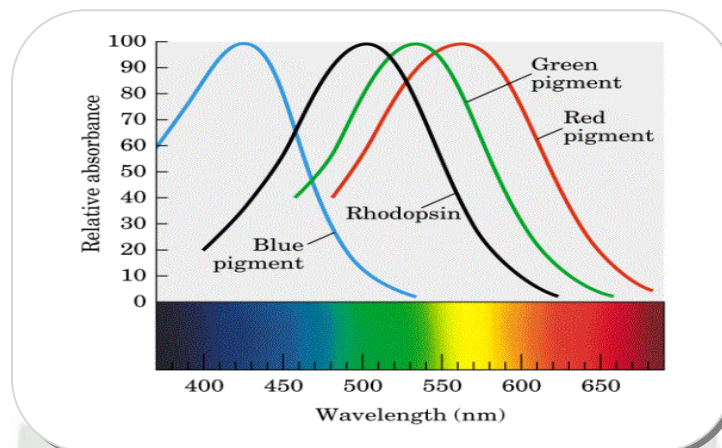


Figura 2. 5: Sensibilidad Espectral humana al color
Fuente: Wikipedia, la formación de la visión humana del color

2.1.5. Que es la visión artificial

La visión artificial es una herramienta que nos sirve para establecer la relación que existe entre un mundo tridimensional y tomar sus vistas bidimensionales. Si tomamos estas vistas podemos hacer una reconstrucción de este espacio y mediante esto simular una característica tridimensional y pasarla en un plano bidimensional.

Es un proceso mediante el cual se extrae información del mundo físico a partir de imágenes, sean éstas análogas o digitales, utilizando para ello un dispositivo electrónico.

Aplicado al campo de la robótica, el mecanismo que permite adquirir imágenes para luego procesarlas y con ello sean comprensibles para el computador, son las cámaras.

Para que el computador pueda visualizar una imagen, es necesario primero convertirla a formato digital, independiente si son imágenes análogas, de colores o a escala de grises. Ya digitalizada la imagen por un dispositivo de adquisición de imagen, ésta corresponderá a un arreglo de matriz de píxeles que representan una serie de valores de intensidad ordenados en un sistema de coordenadas (x,y).

En cuanto al sensor de la cámara, se disponen de dos tipos: CCD usados generalmente en las cámaras digitales, que manejan un rectángulo formado por miles de células generadoras de electricidad que reciben luz y CMOS. Otros sensores comúnmente usados en aplicaciones de visión robótica incluyen sensores termales, infrarrojos y escáneres microscópicos.

Dependiendo de la aplicación, se deberá realizar la elección entre cámaras con alta resolución, alta velocidad o alta calidad, si se desea incursionar en campos como astronomía, etc. o entre cámaras con resolución media si las características a inspeccionar no son tan complejas.

Independientemente del origen de la imagen, éstas suelen requerir una serie de procesamiento debido al ruido, iluminación irregular, mal enfoque, variaciones en los colores y otros problemas que distorsionan la calidad de la imagen.

La visión artificial se consigue procesando las imágenes recibidas desde un sensor, se procesa completamente, píxel por píxel, hasta conseguir reconocer aquello que se buscaba.

Aplicaciones de la visión artificial

La visión artificial tiene un campo amplio, ya que se pueden implementar en cualquier área. Como las que siguen a continuación:

Medica

Procesamiento de imágenes tomadas de las radiografías de los pacientes, ya sean estos rayos x, rayos gamma, o ultravioleta.

Fotogrametría

En donde se requiere realizar mediaciones del espacio 3D a partir de fotografías tomadas, de tal forma que se pueden medir superficies, objetos, construcciones, etc.

Electrónica

Para la construcción, diseño e implementación de componentes eléctricos en una placa electrónica. Aquí se requiere un alto grado de precisión para que estos elementos puedan ajustar sin ningún grado de error.

Mecánica

La construcción de piezas con un alto grado de precisión. La percepción visual capta las imágenes de las piezas que se generan y con esto pueden ir observando que se vayan obteniendo de manera óptima las piezas.

Detección de fallas

Involucra la detección de defectos o artefactos no deseados que tienen una forma desconocida y están en una posición no deseada.

Estimación de movimiento

Mediante una cámara que adquiere una imagen de un objeto en movimiento, se puede hacer una aproximación de este objeto a partir de ciertos puntos que tengan correspondencia con la secuencia de imágenes que se fueron adquiriendo.

2.1.6. Generación de posibles soluciones

Para enseñar a un ordenador a identificar y clasificar objetos, en primer lugar se necesita proporcionarle una plantilla o una representación computacional específica de ese objeto concreto.

Los diversos métodos de detección de objetos pueden clasificarse en dos grandes grupos:

- Los métodos tradicionales basados en detección de características
- Los métodos más recientes que se basan en el reconocimiento de patrones que obtienen la información de manera implícita mediante el aprendizaje desde ejemplos iniciales.

Métodos basados en detección de características

Los métodos basados en detección de características hacen uso de características de bajo nivel como son los bordes, los niveles de gris, el color y el movimiento. La distribución de mínimos locales de niveles de gris puede señalar el contorno de un frasco.

Métodos basados en reconocimientos de patrones

Para resolver problemas más complejos como la detección de múltiples objetos en fondos no controlados se utilizan los detectores basados en reconocimiento de patrones o aprendizaje de ejemplos. La mayoría de estos métodos requieren como paso preliminar un costoso proceso de búsqueda por ventanas a diferentes resoluciones (diferentes escalas y posiciones). Estos detectores se basan en redes neuronales artificiales que distinguen entre las imágenes.

Los métodos basados en características son apropiados para el procesamiento de imágenes en tiempo real cuando el color y el movimiento son posibles. Para imágenes en niveles de gris estáticas, los métodos basados en reconocimiento de patrones son más adecuados.

Actualmente existen diferentes sistemas de visión que ofrecen soluciones para los ambientes de producción, el inconveniente es que dichos sistemas son costosos y tienen esquemas rígidos que limitan las posibilidades de integración y adaptación. Debido a esto, se pensó en implementar un sistema de inspección de color que fuera lo más independiente posible del factor humano. Es por esto que en la búsqueda de un sistema que ofreciera flexibilidad y un costo moderado, se optó por el módulo de programación para crear sistemas de visión que ofrece National Instruments compatible con Labview e IMAQ Visión.

2.1.7. Procesamiento de Imágenes

Aspectos de un Sistema de Visión Artificial

Tenemos como objetivo presentar brevemente los pasos fundamentales de un proyecto de visión artificial. La visión artificial es una técnica basada en la adquisición de imágenes, generalmente en dos dimensiones, para luego procesarlas digitalmente mediante algún tipo de CPU (computadora, microcontrolador, DSP, etc), con el fin de extraer y medir determinadas propiedades de la imágenes adquiridas. Se trata, por tanto, de una tecnología que combina las computadoras con las cámaras de video para adquirir, analizar e interpretar imágenes de una forma equivalente a la inspección visual humana. Actualmente se aplica en diversos procesos científicos y militares, extendiéndose su uso además, en un amplio rango de sectores industriales para la automatización de tareas anteriormente reservadas para la inspección visual humana. Este tipo de incremento de utilización a nivel industrial es debido a que se trata de una tecnología especialmente útil en labores de inspección o supervisión, siendo cuantitativamente más objetivos y consistentes que la inspección humana.

Un segundo aspecto complementario de estas técnicas es la automatización derivada de esta inspección y la posibilidad de actuar sobre el proceso de fabricación o inspección modificando parámetros de

la máquina de producción. Las técnicas de visión artificial, como demuestra su gran uso en la industria, son particularmente apropiadas para la realización de trabajos visuales altamente repetitivos que sean fatigosos o difíciles de realizar para un operario, especialmente cuando este trabajo es ineficiente o costoso en términos económicos o temporales. Un sistema de visión artificial se compone básicamente de los siguientes elementos:

a) Fuente de luz:

Es un aspecto de vital importancia ya que debe proporcionar unas condiciones de iluminación uniformes e independientes del entorno, facilitando además, si es posible, la extracción de los rasgos de interés para una determinada aplicación.

b) Sensor de imagen:

Es el encargado de recoger las características del objeto bajo estudio.

c) Tarjeta de captura o adquisición de imágenes:

Es la interfaz entre el sensor y la computadora o módulo de proceso que permite al mismo disponer de la información capturada por el sensor de imagen.

d) Algoritmos de análisis de imagen:

Es la parte inteligente del sistema. Su misión consiste en aplicar las transformaciones necesarias y extracciones de información de las imágenes capturadas, con el fin de obtener los resultados para los que haya sido diseñado.

e) Computadora o módulo de proceso:

Es el sistema que analiza las imágenes recibidas por el sensor para extraer la información de interés en cada uno de los casos implementando y ejecutando los algoritmos diseñados para la obtención de los objetivos.

Sistema de respuesta en tiempo real:

Con la información extraída, los sistemas de visión artificial pueden tomar decisiones que afecten al sistema productivo con el fin de mejorar la calidad global de producción.

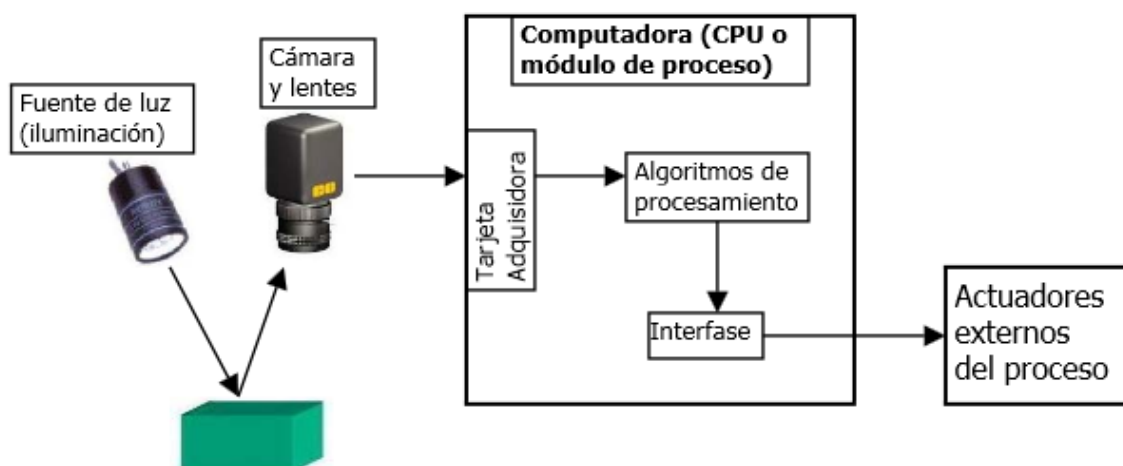


Figura 2. 6: Diseño de visión artificial

Fuente: Elaboración propia

Fuentes de luz

La fuente de luz es un factor de vital importancia en los sistemas de visión artificial y afectan de forma crucial a los algoritmos de visión que se vayan a utilizar bajo esas condiciones. Una iluminación arbitraria del entorno no suele ser aceptable ya que muy probablemente se obtengan

imágenes con un bajo contraste, reflexiones especulares, además de sombras y detalles espurios. Los objetivos de una iluminación adecuada son, básicamente, independizar las condiciones del entorno y resaltar los rasgos de interés de una determinada aplicación.

Tipos de iluminación

A continuación sólo se enuncian los diferentes tipos de iluminación, de las técnicas de iluminación más comunes y generalmente usadas.

- Iluminación posterior (backlight).
- Iluminación Frontal Oblicua y Direccional.
- Iluminación Frontal Axial (difusa).
- Iluminación de Día Nublado (cloudy day illumination, CDI).
- Iluminación de Campo Oscuro (dark field).
- Array de luces.
- Iluminación de campo claro (o brillante).

Sensores de imagen

Los sensores de imagen son componentes sensibles a la luz que modifican su señal eléctrica en función de la intensidad luminosa que perciben. La tecnología más habitual en este tipo de sensores es el CCD (charge coupled devices o dispositivos de acoplamiento de carga) donde se integra en un mismo chip los elementos fotosensibles y el conjunto de puertas lógicas y circuitería de control asociada. En éstos, la señal eléctrica que transmiten los fotodiodos es función de la intensidad luminosa que reciben, su espectro, y el tiempo de integración (tiempo durante el cual los fotodiodos son sensibles a la luz incidente). Otra tecnología que está empezando a extenderse son los sensores **CMOS (complementary metal oxide semiconductor)** dada las ventajas de

éstos sobre los CCD, y la reducción de precios de estos dispositivos. En cuanto al rango dinámico (rango de amplitud entre los valores máximos y mínimos que un elemento puede producir, medido en decibelios) se pasa de los 70dB de los sensores CCD a los 120dB de los sensores CMOS, valor más cercano a los 200 dB del ojo humano, lo que facilita la auto adaptación en el propio chip al brillo existente en el entorno. El sensor CMOS puede amplificar de forma individual la respuesta de los píxeles en función de sus respectivas condiciones de iluminación. El acceso multipíxel permite configurar individualmente distintos intervalos de tiempo de integración, mejorando la calidad de imagen mediante la selección del mejor tiempo de exposición de cada píxel.

Resolución

Existen diferentes arquitecturas de sensores. En primer lugar están los sensores lineales. En éstos, el sensor es una línea de fotodiodos. Esta arquitectura permite la utilización de sensores de 1x1024, 1x2048, 1x4096, e incluso 1x6000 píxeles, lo que la hace muy adecuada para trabajar con altas resoluciones sobre superficies en movimiento. Para condiciones de iluminación muy exigentes o velocidades de trabajo muy altas existe la posibilidad del uso de sensores TDI (time delay integrated). Esta tecnología consiste en el uso de varias líneas de captura sobre la misma línea del objeto, con el fin de sumar su carga y obtener así una mayor sensibilidad. En segundo lugar están los sensores de área. Estos alcanzan resoluciones habituales de 1024x1024, aunque existen en el mercado algunas casas que disponen de cámaras especiales con resoluciones de hasta 3072x2048. En este caso existen tecnologías de adquisición de imágenes, entrelazada y no entrelazada. El método entrelazado captura las líneas pares e impares que forman una imagen en instantes de tiempo diferentes. La tecnología de no entrelazado (progressive scan) captura todas las líneas en el mismo instante de tiempo. Es más costoso económicamente, pero indispensable para trabajar con objetos en movimiento.

Cuantización

Independientemente de la arquitectura utilizada, existen otros aspectos importantes en estos sensores. La cuantización (conversión analógica-digital) determina el número de bits usados para representar la información capturada. Por ejemplo, usando un sistema blanco y negro de 8 bits tenemos 256 niveles diferentes mientras que, usando un sistema de 10 bits, obtendríamos 1024 niveles de gris diferentes, lo que permite una mayor definición. La evolución de las capacidades y características técnicas de estos dispositivos ha seguido una tendencia muy favorable, potenciado en parte por el mayor uso industrial de estos sistemas. Por tanto, en el mercado existen hoy en día numerosos dispositivos para abordar cualquier tipo de aplicación, ya sea en blanco y negro o color, a las velocidades requeridas en los procesos de producción. Entre los avances más significativos, además de la utilización de puertos de comunicación cada vez más potentes y fiables, está una capacidad de proceso más o menos relevante e integrada en la propia cámara. Por ejemplo, la cámara lineal OPSIS 5150ALC dispone de FPGA (Field-programmable gate array, matriz de puertas programable por campo) programable y de un microprocesador Motorola Power PC con sistema operativo en tiempo real VxWorks que permite realizar procesamiento de imágenes. De esta forma se puede transmitir la imagen, la imagen pre procesada, o los resultados del procesamiento de la imagen a través de los puertos de comunicación de que dispone la cámara, haciendo innecesario el uso de tarjetas de captura.

Tarjetas de captura (o adquisidora)

Las tarjetas de captura de imagen permiten transferir la imagen de la cámara a la memoria de la computadora con el fin de que ésta pueda realizar el procesamiento adecuado a las imágenes.

Aspectos importantes en la adquisidora: Velocidad de transmisión, el formato de los datos, la profundidad de píxel, la capacidad de captura

por disparo, la capacidad de pre procesado de la imagen, la velocidad de transferencia de la imagen de la memoria de la tarjeta a la memoria de la computadora, el reset asíncrono o la capacidad de controlar parámetros de la cámara en tiempo real. Evidentemente, todos los datos relativos a velocidades de transmisión son especialmente importantes para poder trabajar en tiempo real. Este aspecto ha mejorado notablemente en los últimos años. Otro parámetro que ha evolucionado muy favorablemente es el formato de los datos. Con la estandarización definitiva de los formatos EIA-644 o LVDS, y del IEEE 1394, se consigue no sólo aumentar la velocidad de transferencia, sino que también se hace con una mayor inmunidad al ruido y con longitudes de cableado aceptadas mucho mayores (de 2-10 metros a 25 metros de distancia cámara – tarjeta). La capacidad de sincronizar la captura de las imágenes con una señal externa también se ha mejorado con el reset asíncrono, lo que habilita el inicio de la captura sin tener que esperar el siguiente vertical blank (es decir, sin necesidad de finalizar completamente la captura de anterior). La capacidad de controlar la cámara con salidas RS-232 de la tarjeta facilita la variación de parámetros de la cámara, fundamentalmente la ganancia de los canales RGB, o de ópticas motorizadas en tiempo real.

Un aspecto relativamente reciente es la capacidad de poder trabajar con multitap. Existen tarjetas capaces de trabajar con varias cámaras a la vez (hasta 6 cámaras de 8 bits por píxel), o usar los canales de que dispone para trabajar con cámaras multitap que transmiten a la vez varios píxeles por cada uno de los canales. Por ejemplo, a 40 MHz, una tarjeta puede recibir la información por un canal de 8 bits a 40 Mbytes/seg mientras que si usa dos canales, la información la puede recibir a 80Mbytes/seg, incrementándose de forma proporcional al número de canales la velocidad de transmisión.

En cuanto a la capacidad de proceso de las tarjetas, actualmente sólo existe la posibilidad de incorporar un Procesador Digital de Señal (DSP),

que permite realizar determinados algoritmos de análisis de imagen en la propia memoria de la tarjeta, acelerando la implementación de los algoritmos de visión. El desarrollo de módulos de proceso altamente especializados ha sido abandonado debido a la evidencia de la evolución superior en prestaciones de los procesadores convencionales, que hacen que el costo de desarrollo de las placas altamente especializadas no sea rentable para las casas fabricantes de la misma, al ser éstos rápidamente superados por los constantes incrementos en velocidad y mejoras internas de los procesadores convencionales.

Otra de las recientes ventajas en cuanto a las tarjetas de captura de imágenes es la integración de la tarjeta de captura, la visualización, y el módulo de proceso (CPU) en un solo elemento. Estos equipos se proporcionan en un chasis adecuado para trabajar en entornos industriales.

Algoritmos de análisis de imagen

Los algoritmos relacionados con visión artificial son muy variados y abarcan numerosas técnicas y objetivos. Los pasos más comunes en que consiste el análisis o procesamiento de imagen se muestran en la figura.

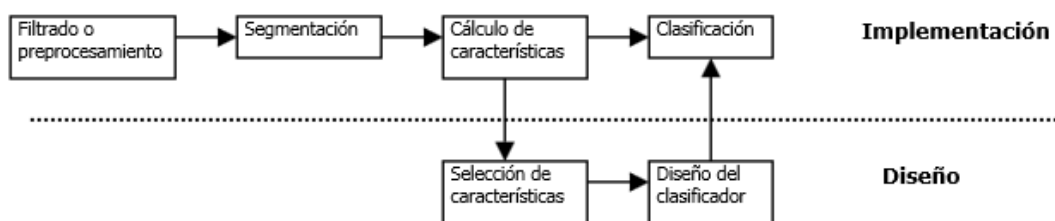


Figura 2. 7: Algoritmos de análisis de imagen

Fuente: Digital Image Processing Second Edition, Rafael C. Gonzalez – Richard E. Woods

Filtrado o Pre-procesamiento

Cuando se adquiere una imagen mediante cualquier sistema de captura, por lo general esta no es directamente utilizable por el sistema de visión.

La aparición de variaciones en intensidad debidas al ruido, por deficiencias en la iluminación, o la obtención de imágenes de bajo contraste, hace necesario un pre procesamiento de la imagen con el objetivo fundamental de corregir estos problemas, además de aplicar aquellas transformaciones a la imagen que acentúen las características que se deseen extraer de las mismas, de manera que se facilite las operaciones de las etapas posteriores. A este campo pertenecen las técnicas de transformaciones geométricas, las basadas en el histograma, el filtrado espacial y el filtrado frecuencial. Aunque existen líneas de investigación abiertas, no es el campo más activo en ese sentido, y las técnicas existentes son bastante útiles para la mayoría de las aplicaciones. Algunas de las técnicas más habituales son:

- **Conversión de los niveles de gris:** Su objetivo es la mejora de la calidad de la imagen
- **Transformaciones geométricas:** Los objetivos de ésta son la corrección de la perspectiva y la reconstrucción tridimensional de los objetos de la escena.
- **Transformación del histograma:** las transformaciones del histograma pueden facilitar la segmentación de objetos de la imagen, aunque habitualmente sólo sirve para modificar el contraste de la imagen y el rango dinámico de los niveles de gris.
- **Filtrado espacial y frecuencial:** Dado que pueden suponer una importante modificación de la imagen original, generalmente sólo se consideran filtrados aquellas operaciones realizadas en el dominio espacial o frecuencial que suponen transformaciones pequeñas de la imagen original.

Segmentación o aislamiento de los objetos de interés

Cuando ya se dispone de la imagen capturada y filtrada, es necesario aislar o separar los objetos de interés de la escena. Se pretende por

tanto dividir una imagen en diferentes regiones, o dicho de otra forma, detectar automáticamente los bordes entre los elementos o regiones. Las operaciones de segmentación de una escena dependen de la propia escena y de la información que se busque dentro de la imagen. Por ejemplo, en fotografía aérea se utiliza para separar regiones pobladas, agrícolas, bosques, aguas, etc. En imagen médica se usa para aislar zonas enfermas de los órganos, o para detectar y contar microorganismos en microscopía. **Industrialmente**, se usa como primer paso para la medición de piezas, para determinar la correcta disposición del etiquetado de los envases (aislar el envase, aislar la etiqueta, determinar si la relación entre estos dos objetos es correcta), para la medición de irregularidades en superficies homogéneas, para la detección de cambios de texturas en las que el sistema ha de determinar si es un defecto o no lo es, etc. Las técnicas básicas de segmentación se pueden dividir en tres grupos:

- Aplicación de umbrales de niveles de gris
- Agrupación por rasgos comunes.
- Extracción de bordes.

La segmentación de imagen es una temática actual de investigación y se han definido numerosos métodos diseñados para aplicaciones particulares. No obstante, no hay métodos que sean útiles en una amplia variedad de situaciones, por lo que cada problema necesita la adaptación o desarrollo de nuevas técnicas. Extracción y selección de características Con los objetos a estudiar ya definidos, es necesario extraer una serie de medidas que los caractericen adecuadamente, en forma de vector de características. Como norma general, para la elección de características se intenta que:

- Sean discriminantes, es decir, tengan valores numéricos diferentes para clases diferentes.

- Sean fiables, es decir, tengan valores parecidos dentro de una misma clase.
- Estén incorreladas, es decir, obtener la misma información con el mínimo número de características.
- Se calculen en un tiempo aceptable, de manera que sea utilizable en problemas de tiempo real. Dado que las características relevantes son a menudo desconocidas a priori, habitualmente se introducen muchas características candidatas para intentar representar mejor las clases, aun cuando muchas de ellas puedan ser parcial o completamente redundantes o irrelevantes. No obstante, existen unas técnicas para mejorar las prestaciones del conjunto de características escogido inicialmente. Éstas están íntimamente ligadas al reconocimiento de formas en su etapa de diseño del clasificador.

Reconocimiento de formas e inteligencia artificial

Normalmente, la visión artificial se utiliza para tomar decisiones a partir de la información proporcionada por el sistema de adquisición de imágenes y las transformaciones y operaciones realizadas con ellas. La información extraída se puede considerar como un vector que recoge las características o rasgos diferenciadores de la imagen analizada. En el caso de que se trate de una aplicación de medición, este vector recoge todas aquellas medidas que se deseaban obtener. En aplicaciones de inspección, y sobre todo, en aplicaciones de clasificación, este vector es el conjunto de datos con los que ha de trabajar un reconocedor o clasificador encargado de extraerlas conclusiones posibles a partir del vector de entrada. Para el diseño del clasificador es necesaria una etapa de selección de características y una etapa de aprendizaje o entrenamiento. Generalmente, se usa el propio clasificador como evaluador del conjunto de características en prueba dado que el método que proporciona la mayor fiabilidad, aunque sea el más costoso

temporalmente y de menor generalidad o capacidad de aplicación del clasificador a otros problemas.

Respecto a los métodos clásicos de búsqueda secuencial, actualmente uno de los campos de investigación abiertos es el uso de algoritmos genéticos para la selección en paralelo de características.

En el reconocimiento de formas aplicado a la visión artificial se utilizan técnicas de re-conocimiento geométrico de formas, como el aprendizaje supervisado (se conoce la clase a la que pertenece cada vector) en condiciones estadísticas o algoritmos de clasificación no supervisados oclustering, además, las redes neuronales, siendo estas últimas especialmente interesantes por su capacidad de aprendizaje adaptativo. Existen numerosos algoritmos para cada una de estas técnicas y se investiga ampliamente el desarrollo de nuevos algoritmos e implementaciones hardware de éstos. El problema fundamental de estas técnicas, es que cada una de ellas suele ser la más adecuada para las características de un tipo específico de problema no conocidas priori, lo que dificulta la selección de la técnica con la que se desea abordar inicialmente el problema.

CPU o módulo de proceso

Se trata del sistema encargado de recibir y analizar, mediante los algoritmos adecuados, las imágenes a la velocidad necesaria para poder interactuar con el entorno en tiempo real. Es, pues, la parte del sistema encargada de implementar las funciones y objetivos para los que se haya diseñado el sistema.

Sistemas de respuesta

Se trata de automatismos que responden electromecánicamente con el fin de corregir o evitar, por ejemplo, en los sistemas de producción, las causas generadoras de los problemas de detección. Otra de las funciones de estos sistemas es la generación de estadísticas e informes

de del proceso al que están asociados, lo que permite disponer de una información global de los procesos para facilitar la toma de decisiones. En cuanto a los dispositivos electromecánicos utilizables en los sistemas de visión, destacan los variadores de frecuencia para el control de motores y la comunicación entre las PC's y los autómatas programables que gestionan los distintos dispositivos mediante redes de campo como "Profibus"o "WorldFIP", y más recientemente mediante "Industrial Ethernet", y protocolos Wireless. Estos buses permiten la integración de los sistemas de visión y sus acciones de respuesta dentro de la estructura de las empresas, lo que supone un ahorro económico en los casos en que ya se disponga de esta estructura

Filtrado Espacial

Introducción

El filtrado espacial es la operación que se aplica a una imagen para resaltar o atenuar detalles espaciales con el fin de mejorar la interpretación visual o facilitar un procesamiento posterior, y constituye una de las técnicas comprendidas dentro del realce de imágenes. Ejemplos comunes incluyen aplicar filtros para mejorar los detalles de bordes en imágenes, o para reducir o eliminar patrones de ruido. El filtrado espacial es una operación "local" en procesamiento de imagen, en el sentido de que modifica el valor de cada píxel de acuerdo con los valores de los píxeles que lo rodean; se trata de transformar los niveles de gris originales de tal forma que se parezcan o diferencien más de los correspondientes a los píxeles cercanos.

Frecuencia Espacial

La frecuencia espacial define la magnitud de cambios en el nivel de gris por unidad de distancia en una determinada zona de la imagen. Las áreas de la imagen con pequeños cambios o con transiciones graduales en los valores de los datos se denominan áreas de bajas frecuencias.

Las áreas de grandes cambios o rápidas transiciones se conocen como áreas de altas frecuencias (por ejemplo, los bordes). Así, los filtros espaciales se pueden dividir en tres categorías:

- **Filtros pasa bajos:** Enfatizan las bajas frecuencias, suavizando las imágenes y atenuando ruidos. Se trata de asemejar el nivel de gris de cada píxel al nivel de gris de los píxeles vecinos, reduciendo la variabilidad espacial de la imagen. Ello produce un borronado de los bordes, perdiéndose en nitidez visual de la imagen, pero ganando en homogeneidad.
- **Filtros pasa altos:** Enfatizan las altas frecuencias, para mejorar o afilar las características lineales como los límites en general. Realizan por tanto el efecto contrario a los filtros pasa bajos, eliminando las bajas frecuencias.
- **Filtro detectores de bordes:** Realizan otro tipo de operaciones con los píxeles, pero siempre con el resultado de enfatizar los bordes que rodean a un objeto en una imagen, para hacerlo más fácil de analizar. Estos filtros típicamente crean una imagen con fondo gris y líneas blancas y negras rodeando los bordes de los objetos y características de la imagen.

Convolución Espacial

El filtrado espacial se realiza trasladando una matriz rectangular de dos dimensiones (también llamada ventana, kernel, máscara o núcleo) que contiene "pesos" o ponderaciones sobre la imagen en cada localización de píxel. Se evalúa el píxel central de la ventana de acuerdo con los píxeles de alrededor y sus valores de ponderación. Cuando un nuevo valor es así calculado, se desplaza la ventana sobre el siguiente píxel, realizando la misma operación. Este proceso de evaluarla vecindad ponderada del píxel se denomina "convolución bidimensional", y a la matriz de filtro se le conoce como "kernel de convolución".

En general, la convolución de una imagen f de $M \times N$ con una máscara h de $m \times n$ está dada por la siguiente expresión:

$$g(x, y) = \sum_{i=-a}^a \sum_{j=-b}^b f(x+i, y+j) \cdot h(i, j) \quad a = \frac{m-1}{2}, b = \frac{n-1}{2}$$

$h(-1,-1)$	$h(-1,0)$	$h(-1,1)$
$h(0,-1)$	$h(0,0)$	$h(0,1)$
$h(1,-1)$	$h(1,0)$	$h(1,1)$

Figura 2. 8: Mascara de convolución

Fuente: Digital Image Processing Second Edition, Rafael C. Gonzalez – Richard E. Woods

Por ejemplo, considerémosla máscara de convolución mostrada en la figura 2.8 y una imagen genérica representada por f . Dado que $m=3$ y $n=3$, a y b son iguales a 1. Teniendo en cuenta todo lo anterior, la respuesta a la máscara de convolución dada, o sea, $g(x,y)$, está dada por la expresión anterior. Luego, evaluando la respuesta para un punto cualquiera, por ejemplo (Figura 2.9), obtenemos lo siguiente:

$$g(5,4) = \sum_{i=-1}^1 \sum_{j=-1}^1 f(5+i, 4+j) \cdot h(i, j)$$



Figura 1: Máscara de convolución.

$$g(5,4) = f(4,3) \cdot h(-1,-1) + f(4,4) \cdot h(-1,0) + f(4,5) \cdot h(-1,1) + f(5,3) \cdot h(0,-1) + f(5,4) \cdot h(0,0) + f(5,5) \cdot h(0,1) + f(6,3) \cdot h(1,-1) + f(6,4) \cdot h(1,0) + f(6,5) \cdot h(1,1)$$

Figura 2. 9: Ejemplo mascara de convolucion

Fuente: Digital Image Processing Second Edition, Rafael C. Gonzalez – Richard E. Woods

Filtros Pasa bajos

El resultado de aplicar un filtro pasa bajos a una imagen es simplemente el promediado de los píxeles contenidos en el vecindario(o entorno) de la máscara utilizada.

Si bien anteriormente se dieron los usos de los filtros pasa bajos, generalmente se los utiliza para atenuar los detalles irrelevantes en una imagen. Otra de las utilidades del filtro pasa bajos, aparte de la más obvia que es la atenuación del ruido, es el suavizado de los falsos contornos producidos por una cuantización con un número insuficiente de niveles de gris.

El procedimiento básico del filtro pasa bajos es reemplazar el valor de cada píxel en una imagen por el promedio de los niveles de gris del vecindario definido por la máscara.

La figura 2.10 muestra dos máscaras (o núcleos) de 3x3. Usando la primera se obtiene el promediado estándar de los píxeles abarcados por aquella. Se debe notar que, en lugar de ser 1/9, los coeficientes de la máscara son todos 1's. Esto se hace así debido a que es más eficiente computacionalmente tener todos los coeficientes con el valor 1. Luego, la imagen completa se divide por 9 (o sea, por la suma de todos los coeficientes).

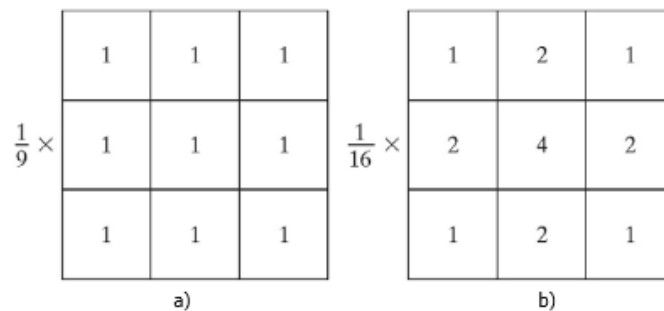


Figura 2. 10: Dos mascaras de 3x3 para realizar un filtrado pasa bajos (suavizado, promediado)

Fuente: Digital Image Processing Second Edition, Rafael C. Gonzalez – Richard E. Woods

La segunda máscara de la figura 2.10 produce un promedio ponderado. Como se puede ver, los píxeles son multiplicados por diferentes coeficientes, dándoles así más importancia (peso) a algunos píxeles que a otros. En el caso particular de la máscara de la figura 2, el píxel del

centro se multiplica por el valor más alto, mientras que los demás píxeles son pesados teniendo en cuenta la inversa de su distancia al centro. La razón de darle más importancia al píxel central y menos importancia a los píxeles más alejados al centro, radica en la necesidad de reducir el borronado durante el proceso de suavizado. Como se hacía en el caso anterior, primero se multiplican los píxeles por los coeficientes de la máscara, que son números enteros potencias de 2 (mayor eficiencia, rapidez), y luego se divide por la suma de todos los coeficientes de la máscara, o sea, 16. En la figura 2.11 pueden verse los efectos del filtro espacial pasa bajos, o suavizado. Teniendo en cuenta la imagen original, figura 2.11 (a), las figuras 2.11 (b) a 2.11 (f) son el resultado de filtrar la imagen original utilizando núcleos cuadrados de $n=3, 5, 9, 15$ y 35 , respectivamente. Para la máscara de 3×3 , notamos un leve borronado general de la imagen, pero los detalles que tienen aproximadamente el mismo tamaño que el núcleo han sido afectados considerablemente. Para $n=9$, se ve un borronado mucho mayor, y también que el 20% del círculo negro casi se confunde con el fondo, ilustrando así el efecto de mezclado que el borronado produce sobre aquellos objetos cuyos niveles de gris son parecidos a los de sus vecinos. Otro efecto que se puede notar es la reducción del ruido en los rectángulos. Para $n=15$ y 35 , se puede ver que el borronado es excesivo. Este tipo de borronado se utiliza para eliminar los objetos pequeños en una imagen

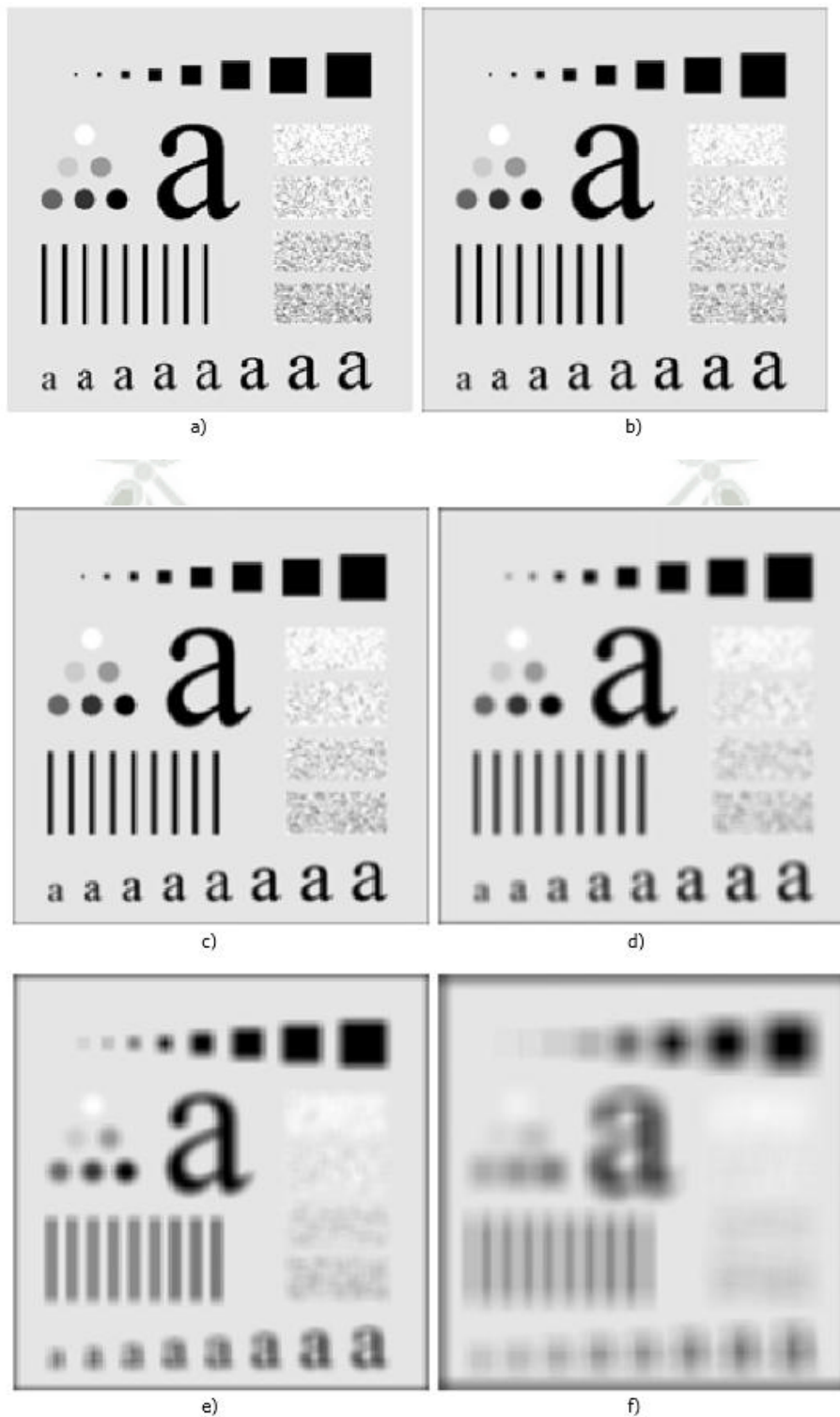


Figura 3: a) Imagen original, 500x500 píxeles. (b) – (f) Resultados de suavizar con máscaras cuadradas de $n=3, 5, 9, 15$ y 35 , respectivamente.

Figura 2. 11: Resultados de suavizar con máscaras cuadradas

Fuente: Digital Image Processing Second Edition, Rafael C. Gonzalez – Richard E. Woods

Como ya lo mencionamos, una aplicación importante de este tipo de filtrado es el borronado de una imagen con el fin de obtener una presentación burda de los objetos de interés, de manera que la intensidad de los objetos pequeños se confunda con la del fondo y los objetos grandes sean fáciles de detectar. Por último, presentaremos la implementación general para filtrar una imagen de $M \times N$ con una máscara de promedio ponderado de $m \times n$ (m y n impares):

$$g(x, y) = \frac{\sum_{i=-a}^a \sum_{j=-b}^b f(x+i, y+j) \cdot h(i, j)}{\sum_{i=-a}^a \sum_{j=-b}^b h(i, j)}$$

Donde el denominador es simplemente la suma de los coeficientes de la máscara y, por lo tanto, una constante que sólo debe calcularse una vez.

Filtros estadísticos: filtro de mediana

Los filtros estadísticos son filtros espaciales no lineales cuya respuesta está basada en ordenar los píxeles abarcados por una máscara y luego reemplazar el valor del píxel central con el valor determinado por el resultado del ordenamiento. El más conocido de estos filtros es el filtro de mediana, el cual reemplaza el valor del píxel central por la mediana de los niveles de gris del vecindario de ese píxel (el valor original del píxel es incluido en el cálculo de la mediana). Los filtros de mediana son muy usados debido a que, para ciertos tipos de ruidos aleatorios, proveen una excelente reducción de ruido y un borronado considerablemente menor que los filtros lineales de suavizado del mismo tamaño. Los filtros de mediana son particularmente efectivos cuando el ruido es del tipo impulso (también llamado ruido sal y pimienta) debido a que aparece como puntos negros o blancos sobrepuestos en la imagen.

La mediana, ξ , de un conjunto de valores es aquella en la que la mitad de los valores en el conjunto son menores o iguales que ξ , y la otra

mitad es mayor o igual a ξ . Por ejemplo, si en una imagen tomamos un conjunto de píxeles de 3×3 con valores $\{1, 9, 5, 0, 8, 7, 1, 2, 4\}$ la mediana para este caso será el valor 4, ya que la mitad de este conjunto es menor (o igual) y la otra mitad es mayor (o igual) a éste: $\{0, 1, 1, 2, 4, 5, 7, 8, 9\}$.

El procedimiento general para realizar el filtro de mediana en cualquier punto consiste en ordenar los valores de dicho píxel y los de su vecindario, determinar la mediana, y asignar éste último valor al píxel en cuestión. Para un vecindario de 3×3 , la mediana es el 5to valor más grande, para uno de 5×5 es el 13mo valor más grande, y así. Cuando algunos valores en un vecindario son iguales, éstos se ven agrupados. Por ejemplo, supongamos que un vecindario de 3×3 tiene los valores $\{10, 20, 20, 20, 15, 20, 20, 25, 100\}$. Ordenando este conjunto nos queda $\{10, 15, 20, 20, 20, 20, 20, 25, 100\}$, de donde se ve que la mediana es 20. Así, la principal función de los filtros de mediana es hacer que los puntos con niveles de gris distintos sean más parecidos a los de su vecindario. En la figura 2.12 puede verse los efectos que produce el filtro de mediana en una imagen contaminada con ruido sal y pimienta.

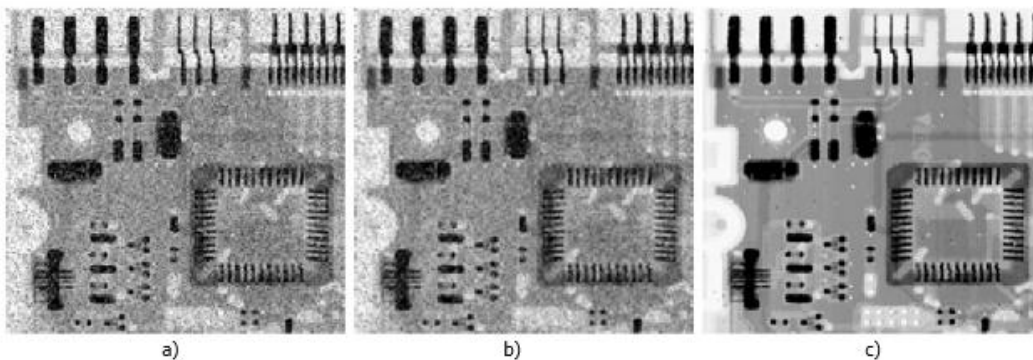


Figura 4: a) Imagen con ruido sal y pimienta. b) Filtrado de (a) con un filtro de suavizado de 3×3 . c) Reducción de ruido con un filtro de mediana de 3×3 .

Figura 2. 12: Filtro de mediana

Fuente: Digital Image Processing Second Edition, Rafael C. Gonzalez – Richard E. Woods

La figura 2.12 (b) es el resultado de promediar la imagen original, figura 2.12 (a), con una máscara de 3x3, mientras que la 2.12 (c) es la consecuencia de filtrar la original con un filtro de mediana de 3x3. Se puede ver claramente que el filtro de mediana es más efectivo a la hora de remover ruido sal y pimienta que los filtros de promediado. No obstante, el precio que se paga es un borronado considerable.

Filtros Pasa altos

El objetivo principal de estos filtros es resaltar los detalles delicados de una imagen (por ejemplo, los bordes) o realzar los detalles de una imagen borronada. Anteriormente, vimos que el suavizado de una imagen, en el dominio espacial, se lograba promediando el vecindario de cada píxel. Dado que el promediado es análogo a la integración, es lógico pensar que en el proceso de filtrado con un pasa altos se realicen operaciones de diferenciación espacial. Fundamentalmente, la magnitud de la respuesta de un operador derivada es proporcional al grado de discontinuidad de la imagen en el punto en donde este operador está siendo aplicado. Por esta razón, la diferenciación de una imagen realza o enfatiza los bordes y otras discontinuidades tales como el ruido y atenúa aquellas áreas cuyos niveles de gris varían lentamente.

Los filtros pasa altos que estudiaremos aquí están basados en la primer y segunda derivada. Las derivadas de una función discreta se definen en términos de diferencias. Existen muchas maneras de definir estas diferencias. Sin embargo, existen requerimientos que debe cumplir cualquier definición de aquéllas. La primera derivada debe ser:

- Cero en las zonas que tienen un nivel de gris constante
- Distinta de cero ante un cambio del tipo escalón o rampa en el nivel de gris
- Y distinta de cero en toda la zona que tiene un cambio del tipo rampa.

Similarmente, cualquier definición de segunda derivada debe ser:

- Cero en zonas con nivel de gris constante;
- Distinta de cero al principio o al final de un cambio del tipo escalón o rampa;
- Y debe ser cero en toda la zona que tiene un cambio del tipo rampa con pendiente constante.

Las definiciones básicas para la primera y segunda derivada de una función bidimensional discreta con respecto a x y a y son las siguientes:

$$\frac{df}{dx} = f(x+1, y) - f(x, y) \quad \frac{d^2 f}{dx^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$
$$\frac{df}{dy} = f(x, y+1) - f(x, y) \quad \frac{d^2 f}{dy^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

Fuente: Digital Image Processing Second Edition, Rafael C. Gonzalez – Richard E. Woods

Filtrado en el Dominio de la Frecuencia

En el Filtrado Espacial se presentaron las diferentes técnicas usadas generalmente para el realce de una imagen en el dominio espacial. Ahora afrontaremos el problema desde el dominio de la frecuencia, más precisamente, con la transformada de Fourier.

Introducción a la Transformada de Fourier y al dominio de la frecuencia

La transformada de Fourier, $F(u)$, de una sola variable, de una función continua, $f(x)$, se define como:

$$F(u) = \int_{-\infty}^{+\infty} f(x).e^{-j2\pi.u.x} dx$$

De la misma forma, dada F(u), podemos obtener f(x) a través de la transformada inversa de Fourier

$$f(x) = \int_{-\infty}^{+\infty} F(u).e^{j2\pi.u.x} du$$

Las expresiones anteriores pueden extenderse fácilmente a dos variables, u y v:

$$F(u, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y).e^{-j2\pi.(u.x+v.y)} dx dy$$

Y, similarmente, la transformada inversa queda definida como:

$$f(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F(u, v).e^{j2\pi.(u.x+v.y)} dudv$$

Ahora bien, en el procesamiento de imágenes se trata con funciones discretas. Por esta razón, presentamos la transformada de Fourier de una función discreta (TDF) de una variable, f(x), con x=0, 1, ..., M-1:

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x).e^{\frac{-j2\pi.u.x}{M}}$$

Para u=0, 1, ..., M-1. Análogamente, dada F(u), la transformada discreta de Fourier inversa está definida como:

$$f(x) = \sum_{u=0}^{M-1} F(u).e^{\frac{j2\pi.u.x}{M}}$$

Para $x=0, 1, \dots, M-1$. Cada uno de los M términos de $F(u)$ es llamado componente de frecuencia de la transformada. Los valores de u en los que la $F(u)$ está definida se llama dominio de la frecuencia. Una propiedad importante de la TDF (y su inversa) es que, a diferencia del par continuo, aquéllas siempre existen. Lo mismo sucede con la TDF para una función de dos variables. En algunos casos, es conveniente expresar la TDF, $F(u)$, en coordenadas polares:

$$F(u) = |F(u)| \cdot e^{-j\phi(u)}$$

Dónde:

$$|F(u)| = \sqrt{R^2(u) + I^2(u)}$$

Es llamado magnitud de la transformada de Fourier, y

$$\phi(u) = \tan^{-1} \left[\frac{I(u)}{R(u)} \right]$$

Se denomina fase o argumento de la transformada de Fourier. $R(u)$ e $I(u)$ son la parte real y la imaginaria de $F(u)$, respectivamente.

Transformada discreta de Fourier bidimensional

La transformada discreta de Fourier de una función (imagen) $f(x,y)$ de tamaño $M \times N$ está dada por la siguiente expresión:

$$F(u, v) = \frac{1}{M \cdot N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cdot e^{-j2\pi \cdot \left[\frac{u \cdot x}{M} + \frac{v \cdot y}{N} \right]} \quad (1)$$

Similarmente, dada $F(u,v)$, obtenemos $f(x,y)$ a través de la transformada de Fourier inversa, definida por la siguiente expresión:

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \cdot e^{j2\pi \left[\frac{u \cdot x}{M} + \frac{v \cdot y}{N} \right]}$$

Para $x=0, 1, \dots, M-1$ e $y=0, 1, \dots, N-1$. Las variables u y v son las variables transformadas o de frecuencia, y x e y son las variables espaciales o de imagen. Como en la sección anterior, definimos el espectro de Fourier, la fase y la magnitud:

$$|F(u, v)| = \sqrt{R^2(u, v) + I^2(u, v)} \quad \phi(u, v) = \tan^{-1} \left[\frac{I(u, v)}{R(u, v)} \right]$$

Donde $R(u, v)$ e $I(u, v)$ son la parte real y la imaginaria de $F(u, v)$, respectivamente. En la práctica es común multiplicar a función $f(x, y)$ (imagen) por $(-1)^{x+y}$ antes de calcular la transformada de Fourier. Debido a las propiedades de las exponenciales se puede demostrar que

$$TDF \{ f(x, y) \cdot (-1)^{x+y} \} (u, v) = F(u - M/2, v - N/2) \quad (2)$$

Donde TDF () representa la transformada discreta de Fourier. Esta expresión coloca al origen de la TDF de $f(x, y) \cdot (-1)^{x+y}$ [o sea, $F(0, 0)$] en $u=M/2$ y $v=N/2$. En otras palabras, multiplicando $f(x, y)$ por $(-1)^{x+y}$ se traslada el origen de $F(u, v)$ a las coordenadas $(M/2, N/2)$, las cuales coinciden con el centro del área de $M \times N$ ocupada por la transformada discreta de Fourier bidimensional.

El valor de la transformada en $(u, v) = (0, 0)$ está dado por la ecuación (1),

$$F(0, 0) = \frac{1}{M \cdot N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \quad (3)$$

de la cual vemos que es el promediado de $f(x, y)$. En otras palabras, si $f(x, y)$ es una imagen, el valor de la transformada en el origen es promediado de los niveles de gris de la imagen. Dado que ambas

frecuencias son cero en el origen, $F(0,0)$ es llamado a veces componente continua (dc) del espectro.

Si $f(x,y)$ es real, es fácil demostrar que su TDF es hermítica y que la magnitud de ésta es simétrica.

La figura 2.13 (a) muestra un rectángulo blanco de 20×40 píxeles sobre un fondo negro de 512×512 . La imagen fue multiplicada por $(-1)^{x+y}$ antes de aplicarle la TDF de manera que la frecuencia cero quede en el centro del espectro (figura 2.13 (b)). Antes de mostrar la TDF, se le aplicó una transformación logarítmica, $F'(u,v) = c \cdot \log(1 + F(u,v))$, para realzar el detalle en los niveles de gris.

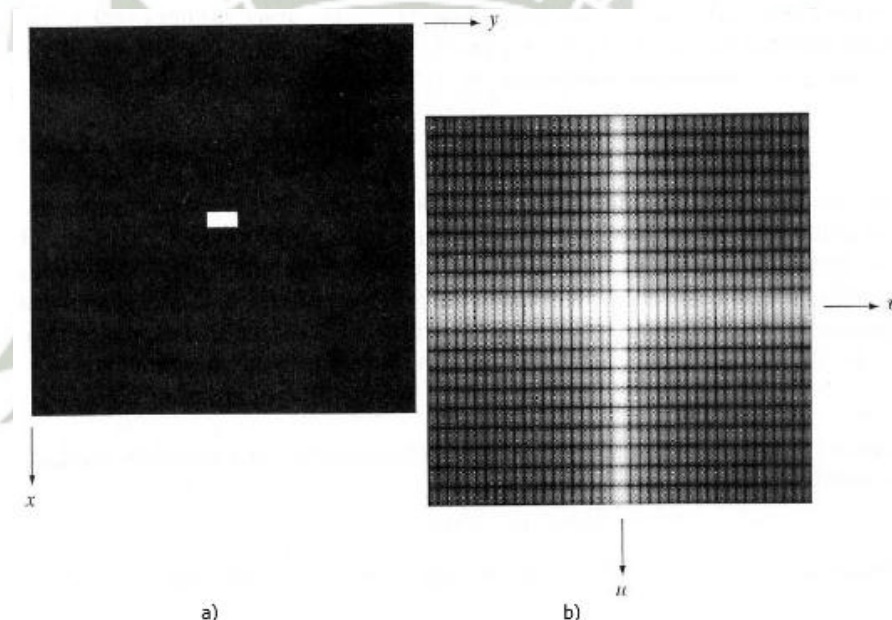


Figura 1: a) Imagen de un rectángulo de 20×40 sobre un fondo negro. b) Espectro de Fourier centrado (con la aplicación de una transformación logarítmica).

Figura 2. 13: Rectángulo blanco

Fuente: Digital Image Processing Second Edition, Rafael C. Gonzalez – Richard E. Woods

Procedimiento para filtrar en el dominio de la frecuencia

Para filtrar en el dominio de la frecuencia se deben seguir los siguientes pasos:

- i) Multiplicar la imagen de entrada por $(-1)^{x+y}$ para centrar la transformada, como lo indica la ecuación (2).
- ii) Calcular la TDF de la imagen, $F(u,v)$, mediante la ecuación (1).
- iii) Multiplicar $F(u,v)$ por una función $H(u,v)$ que actúe como filtro.
- iv) Calcular la TDF inversa del resultado hallado en (iii).
- v) Obtener la parte real del resultado de (iv).
- vi) Multiplicar el resultado de (v) por $(-1)^{x+y}$.

La razón por la que $H(u,v)$ actúa como filtro es que esta función elimina o atenúa ciertas frecuencias en la transformada y deja a otras sin cambios.

Consideremos que $f(x,y)$ es la imagen de entrada del paso (i) del procedimiento antes descrito, y $F(u,v)$ su transformada de Fourier. Entonces, la transformada de Fourier de la imagen filtrada está dada por:

$$G(u, v) = H(u, v) \cdot F(u, v)$$

La multiplicación de H y F involucra funciones bidimensionales, o sea, es el resultado de multiplicar elemento a elemento dos matrices.

En general, los elementos de F son números complejos pero los filtros ($H(u,v)$) que se usan frecuentemente son reales. Estos filtros multiplican de igual forma la parte real y la imaginaria de los elementos de F y, por lo tanto, sólo modifican la magnitud de la transformada manteniendo invariable el argumento.

La imagen filtrada se obtiene aplicando la transformada inversa de Fourier a $G(u,v)$ y multiplicando el resultado por $(-1)^{x+y}$. Cuando la imagen $f(x,y)$ y el filtro $H(u,v)$ son reales, la transformada inversa de Fourier de $G(u,v)$ debería ser real, pero en la práctica no es así ya que existen componentes complejos parásitos que aparecen como

consecuencia del error numérico computacional. Estos componentes deben ser ignorados (son despreciables pero pueden influir en cálculos posteriores).

Algunos filtros básicos

Filtro Notch

Supongamos que queremos poner a cero el nivel de gris promedio de una imagen. De acuerdo con la ecuación (3), el nivel de gris promedio de una imagen está dado por $F(0,0)$. Si hacemos que esta componente sea cero en el dominio de la frecuencia y calculamos la TDF inversa, entonces el nivel de gris promedio de la imagen resultante será ce-ro. Consideremos ahora que hemos centrado la transformada según la ecuación (2). El procedimiento anterior de filtrado se realiza multiplicando $F(u,v)$ por la siguiente función filtro:

$$H(u,v) = \begin{cases} 0 & \Leftrightarrow (u,v) = (M/2, N/2) \\ 1 & \Leftrightarrow (u,v) \neq (M/2, N/2) \end{cases}$$

Los filtros notch son herramientas excepcionalmente útiles cuando es posible identificar efectos espaciales en la imagen causados por componentes de frecuencias específicos y localizados.

Como lo discutimos en el apunte de “Filtrado Espacial”, las frecuencias bajas en la TDF son responsables del nivel de gris general de las áreas suaves en una imagen, mientras que las frecuencias altas son responsables de los detalles (bordes y ruido). Un filtro que atenúa las altas frecuencias mientras que deja pasarlas bajas lo denominábamos filtro pasa-bajos. Un filtro que tiene las características opuestas a las del anterior lo denominábamos pasa-altos.

La figura 2.14 muestra los efectos del filtrado con un pasa bajos y con un pasa altos. Ambos filtros $H(u,v)$, figura 2.14 (b) y 2.14 (d), son

circularmente simétricos. Como era de esperar, la imagen filtrada con el pasa bajos de la figura 2.14 (b) que se exhibe en 2.14 (c) se ve que está suavizada o borroneada con respecto a la original de la figura 2.14 (a). De igual forma, la imagen filtrada con el pasa-altos de la figura 2.14 (d) que se exhibe en 2.14 (e) muestra muy poca variación suave de nivel de gris que la original ($F(0,0)$ fue forzada a cero) pero con mayor detalle en los bordes y las formas.

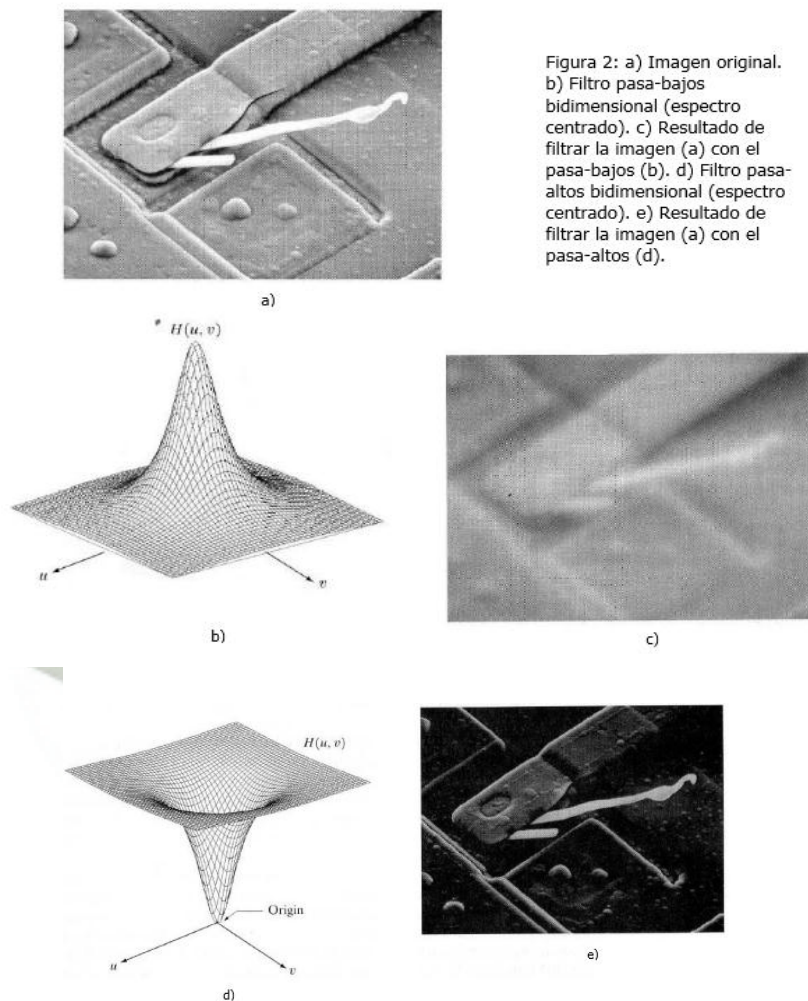


Figura 2. 14: Filtros Notch

Fuente: Digital Image Processing Second Edition, Rafael C. Gonzalez – Richard E. Woods

Filtros Ideales

Pasa-bajos

El filtro pasa-bajos más simple que podemos imaginar es aquel que elimina todas las componentes de alta frecuencia de la transformada de Fourier que se encuentren a una distancia D_0 del origen del espectro (centrado). Este filtro está dado por

$$H(u, v) = \begin{cases} 1 & \Leftrightarrow D(u, v) \leq D_0 \\ 0 & \Leftrightarrow D(u, v) > D_0 \end{cases} \quad D(u, v) = \sqrt{(u - M/2)^2 + (v - N/2)^2}$$

Donde D_0 es un número no negativo (a veces llamado frecuencia de corte) y $D(u, v)$ es la distancia del punto (u, v) al centro del espectro. Debido a que, como mencionamos anteriormente en este apunte, es más práctico trabajar con un espectro centrado (la frecuencia cero de la TDF en el centro), el centro del espectro está dado por $(u, v) = (M/2, N/2)$.

El nombre de filtro ideal se debe a que todas las frecuencias que se encuentran en el interior del círculo de radio D_0 pasan sin atenuación mientras que las que se encuentran fuera del círculo son completamente atenuadas. Como se sabe, los filtros ideales no pueden realizarse físicamente mediante componentes electrónicos, pero sí pueden ser implementados en una computadora.

Pasa-altos

El filtro pasa-altos ideal es el opuesto al pasa-bajos ideal en el sentido de que el primero fuerza a cero todas aquellas frecuencias interiores al círculo de radio D_0 mientras 'deja pasar' sin atenuación a aquellas que se encuentren afuera. Su definición es la siguiente:

$$H(u, v) = \begin{cases} 0 & \Leftrightarrow D(u, v) \leq D_0 \\ 1 & \Leftrightarrow D(u, v) > D_0 \end{cases} \quad D(u, v) = \sqrt{(u - M/2)^2 + (v - N/2)^2}$$

Los filtros ideales tienen características que generalmente son contraproducentes. Un ejemplo de esto son las oscilaciones indeseadas que aparecen en la imagen filtrada. Estos efectos son conocidos como ringing en el idioma inglés. La figura 2.15 ilustra lo comentado en este párrafo.

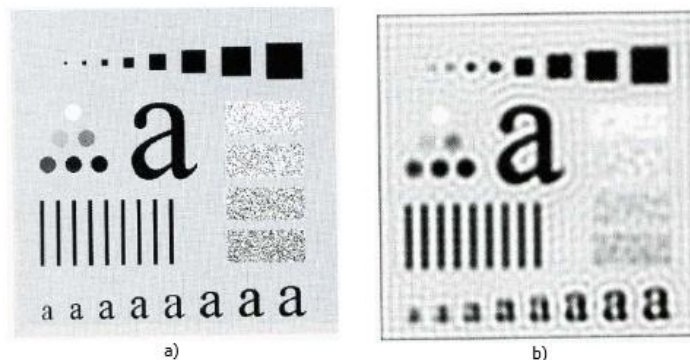


Figura 3: a) Original. b) Resultado de filtrar la imagen original con un pasa-bajos ideal con $D_0 = 30$. Aquí se puede ver el efecto conocido como 'ringing' (oscilaciones).

Figura 2. 15: Filtros ideal (pasa bajos)

Fuente: Digital Image Processing Second Edition, Rafael C. Gonzalez – Richard E. Woods

Filtro Butterworth

Pasa-bajos

La función de transferencia de un filtro Butterworth pasa bajos de orden n , y con una frecuencia de corte a una distancia D_0 del origen, está definida por

$$H(u, v) = \frac{1}{1 + \left[\frac{D(u, v)}{D_0} \right]^{2n}} \quad D(u, v) = \sqrt{(u - M/2)^2 + (v - N/2)^2}$$

En la figura 2.16 (a) se puede ver la forma de un filtro Butterworth pasa bajos, en la 2.16 (b) se muestra al filtro como una imagen, mientras que en la 2.16 (c) se presenta el perfil del filtro Butterworth para distintos n .

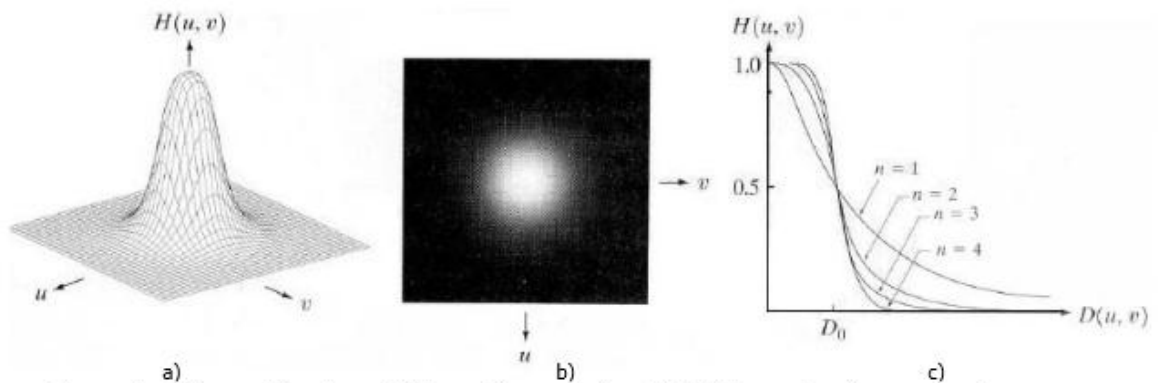


Figura 4: a) Perspectiva de un Butterworth pasa-bajos. b) El filtro mostrado como una imagen. c) Perfil del filtro para distintos n .

Figura 2. 16: Filtro Butterworth

Fuente: Digital Image Processing Second Edition, Rafael C. Gonzalez – Richard E. Woods

A diferencia del filtro ideal pasa-bajos, la función de transferencia del filtro Butterworth no tiene una discontinuidad evidente que establezca una frecuencia de corte bien determinada. Para aquellos filtros que tienen una función transferencia continua (o suavizada) es común definir una frecuencia de corte en un punto tal que $H(u, v)$ esté por debajo de una cierta fracción de su máximo. En el caso del filtro Butterworth, $H(u, v) = 0.5$ (50% de su máximo valor, o sea, 1) cuando $D(u, v) = D_0$.

Para $n=1$, el filtro Butterworth no presenta oscilaciones, para $n=2$, las oscilaciones son casi imperceptibles; mientras que para $n > 2$, los filtros Butterworth presentan oscilaciones indeseadas. Si bien se utilizan diferentes n para el cálculo del filtro Butterworth, el de segundo orden ($n=2$) es el que generalmente se elige ya que tiene un buen compromiso entre un filtrado pasa bajos efectivo y oscilaciones de intensidad aceptable.

Pasa-altos

La función de transferencia de un filtro Butterworth pasa altos de orden n , y con una frecuencia de corte a una distancia D_0 del origen, está definida por:

$$H(u,v) = \frac{1}{1 + \left[\frac{D_0}{D(u,v)} \right]^{2n}} \quad D(u,v) = \sqrt{(u - M/2)^2 + (v - N/2)^2}$$

Filtros basados en funciones Gaussianas

Este tipo de filtro es de particular importancia ya que su forma está fácilmente especificada y porque tanto su TDF como su TDF inversa son funciones Gaussianas reales.

Filtro Gaussiano Pasa-bajos

La forma de este filtro está dada por:

$$H(u,v) = e^{-D^2(u,v)/2\sigma^2} \quad D(u,v) = \sqrt{(u - M/2)^2 + (v - N/2)^2}$$

Donde $D(u,v)$ es la distancia al origen del punto (u,v) (asumiendo que se ha centrado la transformada). Como vemos, este tipo de filtro tiene su máximo en $(0,0)$ que es 1, y luego su valor comienza a disminuir según la función Gaussiana cuya dispersión (o apertura) está dada por la constante σ . Una expresión más familiar del filtro Gaussiano es aquella en donde aparece la frecuencia de corte D_0 :

$$H(u,v) = e^{-D^2(u,v)/2D_0^2} \quad D(u,v) = \sqrt{(u - M/2)^2 + (v - N/2)^2}$$

Cuando $D(u,v) > D_0$, los valores del filtro están por debajo del 60,7% de su máximo.

Otra de las particularidades del filtro Gaussiano es que la TDF inversa de este tipo de filtro no presenta los efectos conocidos como ringing, algo que sí sucedía para el caso de los filtros Butterworth (para $n > 1$).

Filtro Gaussiano Pasa-altos

La función transferencia del filtro Gaussiano pasa altos con frecuencia de corte ubicada a una distancia D_0 del origen está dada por la siguiente expresión:

$$H(u, v) = 1 - e^{-D^2(u, v) / 2 \cdot D_0^2} \quad D(u, v) = \sqrt{(u - M/2)^2 + (v - N/2)^2}$$

Herramientas a utilizar para el desarrollo del proyecto de visión artificial

Para nuestro caso de estudio nos desenvolveremos con IMAQ Visión de National Instruments (NI-IMAQ).

- Software Labview 2011
- Visión development module (visión acquisition software, visión assistant)
- Software Arduino 1.0.5
- VI Package Manager –Labview Interface for Arduino (Librerías)
- Measurement automation.

2.2. INSTRUMENTACIÓN (CÁMARA WEB)

Instrumentación industrial: es el grupo de elementos que sirven para medir, convertir, transmitir, controlar o registrar variables de un proceso con el fin de optimizar los recursos utilizados en éste.

Las variables a medir o controlar pueden ser:

- Variables físicas:
 - Caudal.
 - Caudal másico.
 - Caudal volumétrico.
 - Presión.
 - Temperatura.
 - Nivel.
 - Nivel de líquidos.
 - Nivel de sólidos.
 - Velocidad.
 - Peso.
 - Humedad.
 - Punto de rocío.
- Variables químicas:
 - pH.
 - Conductividad eléctrica.

En este proyecto se utiliza una cámara web, como instrumento para detectar y mediante los algoritmos desarrollados, puedan ser clasificados.

2.2.1. Definición de cámara web

Webcam" significa cámara para uso en red. Es un dispositivo que se conecta al puerto USB de la computadora, y así permite captar video y tomar fotos digitales con resolución baja, por lo que no ofrece una gran calidad de gráficos a diferencia de una cámara fotográfica digital, videocámara digital ó un teléfono celular moderno. El video que capta, lo codifica especialmente para enviarlo por Internet (red mundial de redes) en tiempo real (lo más instantáneamente posible), hacia otra computadora dónde otro usuario puede visualizarlo al momento. Son

muy utilizadas para conversaciones vía Internet y hacer más personalizada la charla, así como también para actividades de vigilancia.

Funcionamiento interno de una cámara Web:

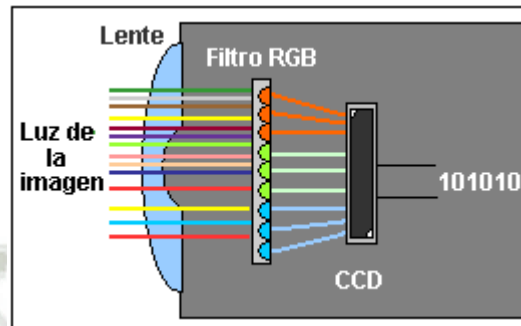


Figura 2. 17: Funcionamiento interno
Fuente: Pagina de especificaciones de cámara web

La luz de la imagen pasa por la lente, esta se refleja en un filtro RGB (Red-Green-Blue), el cuál descompone la luz en tres colores básicos: rojo, verde y azul. Esta división de rayos se concentra en un chip sensible a la luz denominada CCD ("Charged Coupled Device"), el cuál asigna valores binarios a cada píxel y envía los datos digitales para su codificación en video y posterior almacenamiento.

Características de una cámara web

- Tiene una resolución por lo general baja, aproximadamente 640 X 480 píxeles, ya que las imágenes transmitidas instantáneamente por Internet deben de tener un tamaño muy bajo archivo.
- Dependiendo el modelo, tienen la lente giratoria de hasta 360° horizontales, una base adaptable a la superficie, e incluso micrófono integrado.
- Pueden tomar fotos al instante pero con baja resolución.
- Su diseño es muy específico para aplicaciones de entretenimiento y en algunos casos como cámara de vigilancia.

Conector

El conector con que cuenta es un USB, hay 2 versiones de este conector, el USB 1.1 y el USB 2.0. En este tipo de conectores, el macho se distingue por ser el que viene en los dispositivos extraíbles, y el conector hembra es el que se encuentra montado en la computadora, e inclusive en equipos de sonido

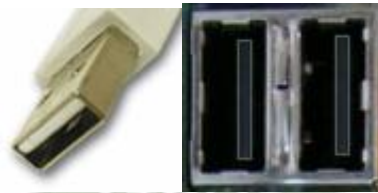


Figura 2. 18: Conectores

Fuente: Pagina de especificaciones de cámara web

Capacidades de resolución

Se refiere a la cantidad máxima de píxeles que es capaz de capturar para generar la imagen. Un píxel es cada uno de los puntos que conforman la imagen y a medida de que tenga mayor cantidad de ellos, se tendrá un mayor detalle de la imagen. Se pueden abreviar como Kpx y Mpx.

- Cámara Web, marca Genius®, modelo Gigaware 2500465, USB.

Esto es: $(640) \times (480 \text{ píxeles}) = 307,200$ píxeles de resolución ó 300 Kilopíxeles.

Especificaciones técnicas

Sensor de imagen: Sensor CMOS de 640×480 píxeles

Especificación del lente: $F = 2,4$, $f = 3,8$ mm

Balance de blancos: Automático

Exposición: Automático

Velocidad de fotogramas: 640 x 480 píxeles @ 30fps / 352 x 288 píxeles @ 30fps / 320 x 240 píxeles @ 30fps

Distancia de enfoque: Enfoque manual, 0,4 pulg. (1 cm) hasta Infinito

Campo de profundidad: 2 pulg. (5 cm) hasta Infinito

Formato vídeo: WMV / MPG

Resolución vídeo captura: 640 x 480 / 352 x 288 / 320 x 240 / 176 x 144/160 x 120 píxeles

Formato de imágenes inmóviles: JPG / BMP

Resolución captura imágenes: Hasta VGA píxeles (1,3 megapíxeles interpolados)

Control de flicker (parpadeo): 60Hz



Figura 2. 19: Cámara web Gigaware

Fuente: Pagina de especificaciones de cámara web Gigaware 2500465

Imágenes digitales

Una imagen digital es un arreglo en dos dimensiones de valores que representan la intensidad luminosa, $f(x,y)$, en un sistema de coordenadas x, y tal como se muestra en la Figura 2-22.

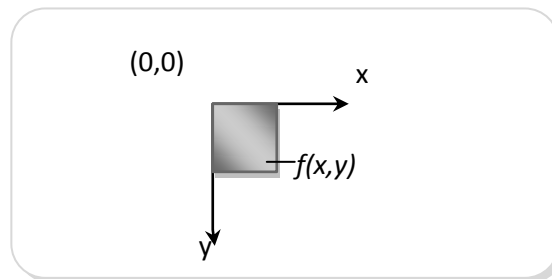


Figura 2. 20: Representación de un Pixel
Fuente: Wikipedia Imágenes digitales

Donde f representa la intensidad luminosa en el punto (x,y) , el cual se denomina píxel o unidad básica de una imagen. El punto $(0,0)$ es una referencia para las coordenadas en la imagen. En procesamiento de una imagen digital, un sensor convierte una imagen en un número finito de píxeles. El sensor asigna a cada píxel una ubicación (coordenada) y un nivel de gris o un determinado color que especifique el brillo o el color del píxel, la Figura 2.21 muestra una imagen a color y su representación en píxeles tanto en escala de colores como en escala de grises.

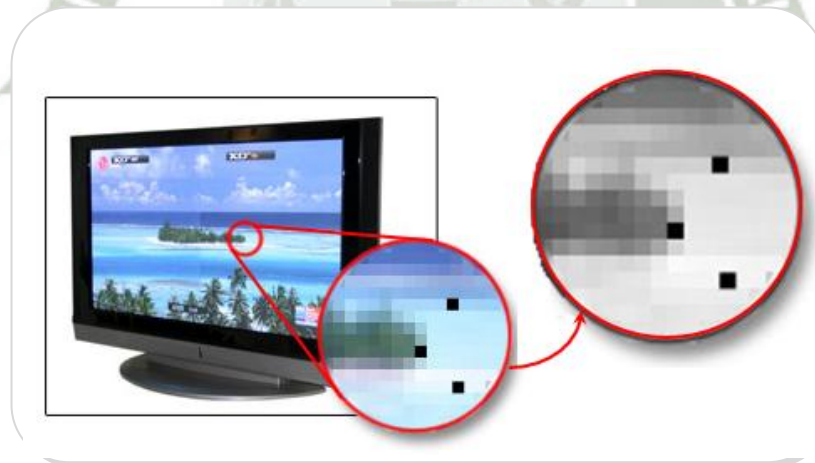


Figura 2. 21: Píxeles extraídos de una imagen
Fuente: Wikipedia Imágenes digitales

Una imagen digital tiene tres propiedades básicas: resolución, definición y número de planos.

Resolución de una imagen

Está determinada por su número de filas y columnas de píxeles. Una imagen compuesta por m columnas y n filas tiene una resolución de $m \times n$. Esta imagen tiene m píxeles a lo largo de su eje horizontal y n píxeles a lo largo de su eje vertical.

Definición de una imagen

Indica el número de tonos que se puede ver en la imagen. La profundidad de bits de una imagen es el número de bits utilizados para codificar el valor de un píxel. Para una representación en n bits, la imagen tiene una definición de 2^n , lo cual significa que un píxel puede tener 2^n valores diferentes. Por ejemplo, si n es igual a 8 bits, un píxel puede tener 256 valores diferentes, que van desde 0 hasta 255. Si n es igual a 16 bits, un píxel puede tener 65536 valores diferentes que van desde 0 hasta 65535 o desde -32768 hasta 32767. Actualmente, NI Vision sólo soporta un rango de -32768 a 32767 para imágenes de 16 bits.

NI Vision puede procesar imágenes con 8 bits, 10 bits, 12 bits, 14 bits, 16 bits, punto flotante, codificación de colores. La manera en que se codifica la imagen depende de la naturaleza del dispositivo de adquisición de imágenes, el tipo de procesamiento de imágenes que se necesita usar, y el tipo de análisis que necesita realizar. Por ejemplo, la codificación de 8 bits es suficiente si necesita obtener la información acerca del contorno de los objetos que se encuentran en una imagen. Sin embargo, si la aplicación hace uso de reconocimiento de colores, de debe usar imágenes digitales a color.

Número de Planos

El número de planos en una imagen corresponde al número de matrices de los píxeles que componen la imagen. Una imagen a escala de grises

o pseudo-color se compone de un solo plano. Una imagen de color se compone de tres planos: uno para el componente rojo, componente azul, y componente verde, es así que una imagen en color es la combinación de los tres arreglos de píxeles correspondientes a los colores rojo, verde y azul en una imagen RGB. En imágenes HSL se definen por los valores correspondientes a tono, saturación y luminosidad, con respecto a los modelos de color mencionados se tratará más adelante.

Debido a que los planos de una imagen a color se basan en los colores rojo, azul, y verde, es conveniente definir a dichos colores como los elementos básicos de una imagen en color. La percepción del color difiere de persona en persona, por ello es necesario usar modelos en color que aprovechen otras propiedades como lo son la intensidad, saturación, matiz o la iluminación.

Color	Longitud de onda	Frecuencia
rojo	625-740 nm	480-405 THz
naranja	590-625 nm	510-480 THz
amarillo	565-590 nm	530-510 THz
verde	520-565 nm	580-530 THz
azul	450-500 nm	670-600 THz
cian	430-450 nm	700-670 THz

Tabla 2.1: Relación entre frecuencias y colores percibidos[1]

Fuente: Wikipedia, la formación de la visión humana del color

Los colores primarios aditivos son el rojo, el verde, y el azul. Permiten generar los demás colores del espectro visible por medio de la suma de

emisión de luz de diferente longitud de onda. La ausencia de los tres da el negro, y la suma de los tres da el blanco. Estos tres colores se corresponden con los tres picos de sensibilidad de los tres sensores de color en nuestros ojos.

Los colores primarios substrativos son el cian, el magenta y el amarillo, comúnmente confundidos con rojo y azul los dos primeros por su similitud. Se utilizan para generar los otros colores, la ausencia de los tres produce el blanco, y la suma de los tres producen el negro.

Modelo de color RYB

Por sus siglas en inglés Red, Yellow and Blue. Éste modelo se basa en los colores primarios rojo, amarillo y azul, a partir de los cuales se crean el resto de colores mediante su combinación.



Figura 2. 22: Modelo de color RYB
Fuente: Wikipedia modelo de color RYB

Modelo de color RGB

Se basa en la mezcla de los colores rojo, verde y azul (Red, Green and Blue), tal como lo muestra la Figura 2.23, RGB se base en la mezcla de colores aditivos, siendo (1,1,1) el blanco.

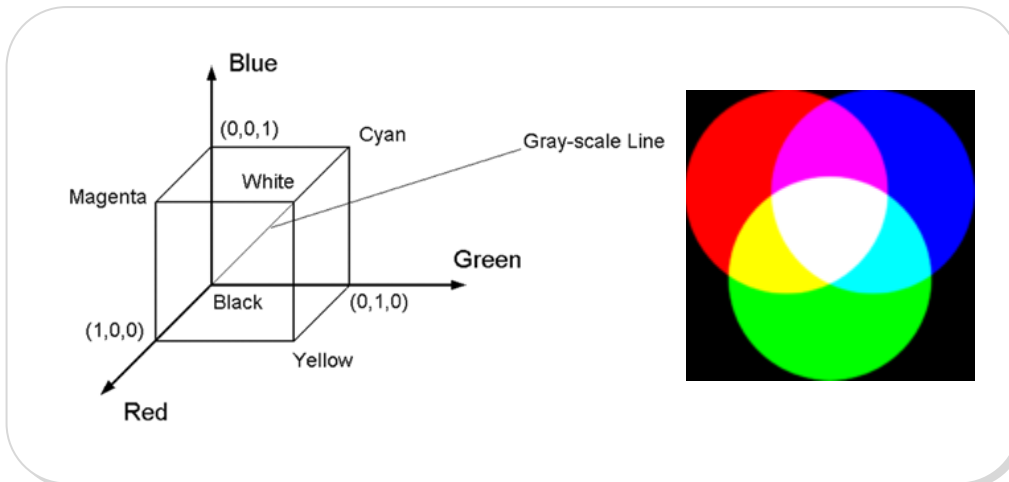


Figura 2. 23: Modelo de color RGB

Fuente: IMAGE PROCESSING WITH LABVIEW AND IMAQ VISION, Thomas Klinger, Chapter 2: Color Images

Para convertir una imagen RGB a escala de grises se aplica la ecuación:

$$GS = 0.333R + 0.333G + 0.333B$$

Tomando en cuenta la luminancia percibida por el ojo humano, tenemos:

$$GS = 0.299R + 0.587G + 0.114B$$

Modelo de color CMY o CMYK

Los complementos del rojo, azul y verde son los colores básicos para un modelo substrativo: el cian, magenta y amarillo. Es fácil ir de RGB a CMY con las ecuaciones mostradas a continuación:

$$C = 1.0 - R$$

$$M = 1.0 - G$$

$$Y = 1.0 - B$$

En el modelo CMY, el negro es creado por mezcla de todos los colores, y el blanco es la ausencia de todos. Cuando el color negro es añadido en el modelo, este modelo de color es denominado modelo CMYK.

Se debe tener en cuenta que sólo con unos colores "primarios" ficticios se puede llegar a conseguir todos los colores posibles. Estos colores

primarios son conceptos arbitrarios utilizados en modelos de color matemáticos que no representan las sensaciones de color reales o incluso los impulsos nerviosos reales o procesos cerebrales. En otras palabras, todos los colores "primarios" perfectos son completamente imaginarios, lo que implica que todos los colores primarios que se utilizan en las mezclas son incompletos o imperfectos.



Figura 2. 24: Modelo de color CMY
Fuente: Wikipedia modelo de color RYB

Modelo de color HSL o HSI

Llamado así por sus siglas en inglés Hue, Saturation, Lightness: Tonalidad, Saturación, Luminancia, también llamado **HSI**, Hue, Saturation, Intensity: Tonalidad, Saturación, Intensidad.

Es útil usar un sistema diferente en color, basado en propiedades más naturales como el tono, la saturación, y la intensidad en color. La Figura 2.25 muestra que este modelo puede ser representado por un sólido mostrado en el lado derecho de la figura. Cualquier punto en color sobre la superficie sólida representa un color totalmente saturado.

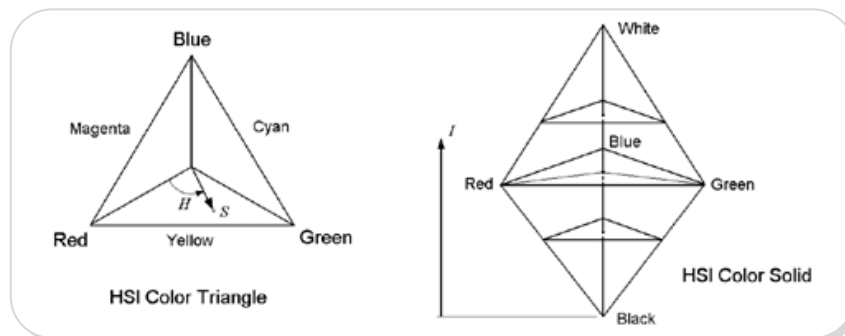


Figura 2. 25: Triángulo de color HSI

Fuente: IMAGE PROCESSING WITH LABVIEW AND IMAQ VISION, Thomas Klinger, Chapter 2: Color Images

El matiz o tono es definido como el ángulo que comienza del eje rojo; la intensidad es representada por la distancia desde el punto negro. Las fórmulas siguientes pueden ser usadas para convertir valores de RGB a HSI:

$$H = \cos^{-1} \left[\frac{\frac{1}{2} [(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right]$$

$$S = 1 - \frac{3}{R + G + B} [\min(R, G, B)]$$

$$I = \frac{1}{3} (R + G + B)$$

Modelo HSV

El modelo **HSV**, del inglés Hue, Saturation, Value: Tonalidad, Saturación, Valor; también llamado **HSB**, Hue, Saturation, Brightness: Tonalidad, Saturación, Brillo. Define un modelo de color en términos de sus componentes constituyentes en coordenadas cilíndricas:

La tonalidad se representa como un grado de ángulo cuyos valores posibles van de 0 a 360°. La Saturación se representa como la distancia al eje de brillo negro-blanco. El valor o brillo del color se representa por la altura en el eje blanco-negro.

El modelo es una transformación no lineal del espacio de color RGB, y se puede usar en progresiones de color, la Figura 2.26 muestra la representación gráfica de este modelo.

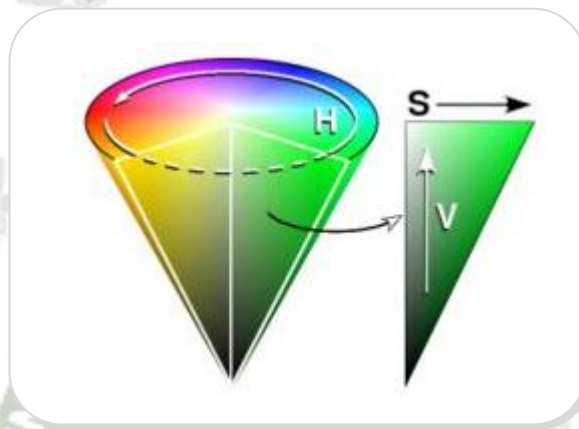


Figura 2. 26: Modelo HSV
Fuente: Wikipedia modelo de color HSV

HSL es similar al modelo HSV pero refleja mejor la noción intuitiva de la saturación y la luminancia como dos parámetros independientes, y por tanto es un modelo más adecuado.

2.3. MICROPROCESADOR (PLACA ARDUINO)

2.3.1. Definición Arduino Uno

Arduino es una herramienta para hacer que los ordenadores puedan sentir y controlar el mundo físico a través de tu ordenador personal. Es una plataforma de desarrollo de computación física (physical computing) de código abierto, basada en una placa con un sencillo microcontrolador

y un entorno de desarrollo para crear software (programas) para la placa. Puedes usar Arduino para crear objetos interactivos, leyendo datos de una gran variedad de interruptores y sensores y controlar multitud de tipos de luces, motores y otros actuadores físicos. Los proyectos con Arduino pueden ser autónomos o comunicarse con un programa (software) que se ejecute en tu ordenador.

El Arduino puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente.

Elementos de la placa Arduino Uno

Es una placa de circuito impreso donde va instalado el microprocesador, la memoria, las conexiones de entrada y salida y la conexión para el puerto usb.

- a) **Puerto usb:** A través de él se cargan las instrucciones a ejecutar, el programa es realizado en el entorno de programación de Aduino-Ordenador.
- b) **Microprocesador:** Realiza las instrucciones almacenadas en el programa de forma cíclica, es un circuito integrado que contiene muchas de las mismas cualidades que una computadora.
- c) **Botón reset:** Permite resetear el programa y permite cargar uno nuevo.
- d) **Pines de entrada y salida:** Permite conectar elementos que dan información y crean actuaciones.

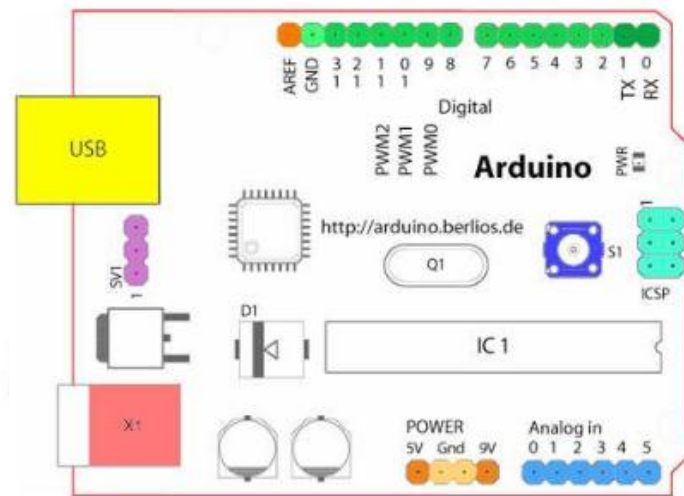


Figura 2. 27: Placa Arduino Uno
Fuente: página web Arduino Uno

- Aref: Pin de referencia analógica (naranja)
- GND: Señal de tierra digital (verde claro)
- Pines digitales 2-13: Entrada y salida (verde)
- Pines digitales 0-1: Entrada y salida del puerto serie TX/TR (verde oscuro)
- Botón de reset: Pulsador (azul oscuro)
- Pines Analógicos 0-5: entradas (azul claro)
- Pines de alimentación y tierra (fuerza: naranja, tierra: naranja claro)
- Entrada de la fuente de alimentación externa (9-12V DC) X1 (rosa)
- Puerto USB (amarillo)

Hardware y cable USB

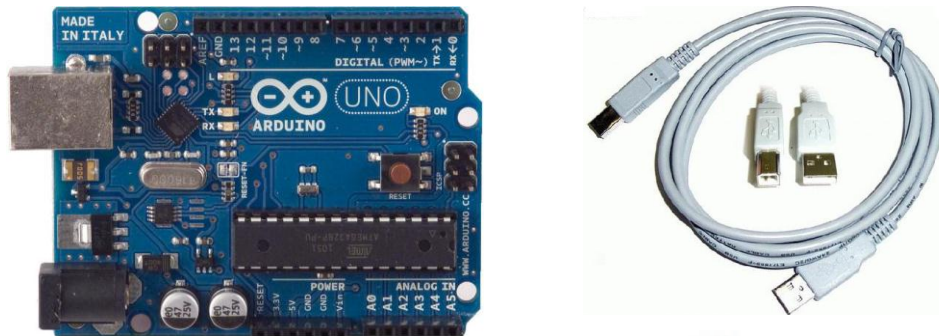


Figura 2. 28: Placa Arduino y Cable USB
Fuente: página web Arduino Uno

Especificaciones técnicas

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current for I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Pines de alimentación



Figura 2. 29: Placa Arduino UNO alimentación
Fuente: página web Arduino Uno

Bien alimentemos al arduino mediante la conexión USB o mediante una fuente externa (recomendada de 7-12V), vamos a tener unas salidas de tensión continua debido a unos reguladores de tensión y condensadores de estabilización.

Estos pines son:

- VIN: se trata de la fuente tensión de entrada que contendrá la tensión a la que estamos alimentando al Arduino mediante la fuente externa.
- 5V: fuente de tensión regulada de 5V, esta tensión puede venir ya sea de pin VIN a través de un regulador interno, o se suministra a través de USB o de otra fuente de 5V regulada.
- 3.3V: fuente de 3.3 voltios generados por el regulador interno con un consumo máximo de corriente 50mA.
- GND: pines de tierra.

Digital Inputs/Outputs

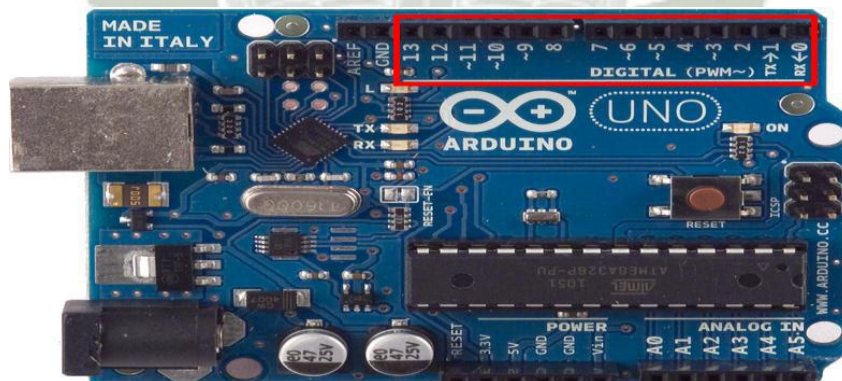


Figura 2. 30: Placa Arduino UNO (señales digitales)
Fuente: Página web Arduino Uno

Cada uno de los 14 pines digitales se puede utilizar como una entrada o salida. Cada pin puede proporcionar o recibir un máximo de 40 mA y

tiene una resistencia de pull-up (desconectado por defecto) de 20 a 50 kOhm. Además, algunos pines tienen funciones especializadas como:

- Pin 0 (RX) y 1 (TX). Se utiliza para recibir (RX) y la transmisión (TX) de datos serie TTL.
- Pin 2 y 3. Interrupciones externas. Se trata de pines encargados de interrumpir el programa secuencial establecido por el usuario.
- Pin 3, 5, 6, 9, 10 y 11. PWM (modulación por ancho de pulso). Constituyen 8 bits de salida PWM con la función analogWrite ().
- Pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos pines son de apoyo a la comunicación SPI.
- Pin 13. LED. Hay un LED conectado al pin digital 13. Cuando el pin es de alto valor, el LED está encendido

2.4. CONTROL POR PC (LABVIEW)

2.4.1. Labview

Labview (acrónimo de Laboratory Virtual Instrumentation Engineering Workbench) es una plataforma y entorno de desarrollo para diseñar sistemas, con un lenguaje de programación visual gráfico. Recomendado para sistemas hardware y software de pruebas, control y diseño, simulado o real y embebido, pues acelera la productividad. El lenguaje que usa se llama lenguaje G, donde la G simboliza que es lenguaje Gráfico.

Este programa fue creado por National Instruments (1976) para funcionar sobre máquinas MAC, salió al mercado por primera vez en 1986. Ahora está disponible para las plataformas Windows, UNIX, MAC y GNU/Linux. La última versión es la 2013, con la increíble demostración de poderse usar simultáneamente para el diseño del firmware de un

instrumento RF de última generación, a la programación de alto nivel del mismo instrumento, todo ello con código abierto.

Los programas desarrollados con LabVIEW se llaman Instrumentos Virtuales, o VIs, y su origen provenía del control de instrumentos, aunque hoy en día se ha expandido ampliamente no sólo al control de todo tipo de electrónica (Instrumentación electrónica) sino también a su programación embebida, comunicaciones, matemáticas, etc. Un lema tradicional de LabVIEW es: *"La potencia está en el Software"*, que con la aparición de los sistemas multinúcleo se ha hecho aún más potente. Entre sus objetivos están el reducir el tiempo de desarrollo de aplicaciones de todo tipo (no sólo en ámbitos de Pruebas, Control y Diseño) y el permitir la entrada a la informática a profesionales de cualquier otro campo. LabVIEW consigue combinarse con todo tipo de software y hardware, tanto del propio fabricante -tarjetas de adquisición de datos, PAC, Visión, instrumentos y otro Hardware- como de otros fabricantes.

Principales características

Su principal característica es la facilidad de uso, válido para programadores profesionales como para personas con pocos conocimientos en programación pueden hacer programas relativamente complejos, imposibles para ellos de hacer con lenguajes tradicionales. También es muy rápido hacer programas con LabVIEW y cualquier programador, por experimentado que sea, puede beneficiarse de él. Los programas en LabView son llamados instrumentos virtuales (VIs) Para los amantes de lo complejo, con LabVIEW pueden crearse programas de miles de VIs (equivalente a millones de páginas de código texto) para aplicaciones complejas, programas de automatizaciones de decenas de miles de puntos de entradas/salidas, proyectos para combinar nuevos VIs con VIs ya creados, etc. Incluso existen buenas prácticas de programación para optimizar el rendimiento y la calidad de la

programación. El labView 7.0 introduce un nuevo tipo de subVI llamado VIs Expreso (Express VIs). Estos son VIs interactivos que tienen una configuración de caja de diálogo que permite al usuario personalizar la funcionalidad del VI Expreso. El VIs estándar son VIs modulares y personalizables mediante cableado y funciones que son elementos fundamentales de operación de LabView.

Presenta facilidades para el manejo de:

- Interfaces de comunicaciones:
 - Puerto serie
 - Puerto paralelo
 - GPIB
 - PXI
 - VXI
 - TCP/IP, UDP, DataSocket
 - Irda
 - Bluetooth
 - USB
 - OPC...
- Capacidad de interactuar con otros lenguajes y aplicaciones:
 - DLL: librerías de funciones
 - .NET
 - ActiveX
 - Multisim
 - Matlab/Simulink
 - AutoCAD, SolidWorks, etc
- Herramientas gráficas y textuales para el procesado digital de señales.
- Visualización y manejo de gráficas con datos dinámicos.
- Adquisición y tratamiento de imágenes.

- Control de movimiento (combinado incluso con todo lo anterior).
- Tiempo Real estrictamente hablando.
- Programación de FPGAs para control o validación.
- Sincronización entre dispositivos.

Entorno de Imaq visión en Labview

Desarrollado por National Instruments (NI), Labview cuyas siglas que en inglés son el acrónimo de Laboratory Virtual Instrument Engineering Workbench y cuyo logo característico se muestra en la Figura 2.31, es una plataforma que permite crear programas usando notación gráfica que simulan elementos físicos unidos de manera virtual.

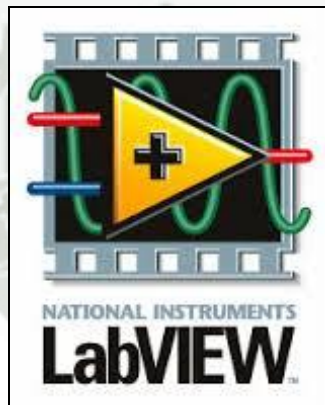


Figura 2. 31: Logo de LabVIEW
Fuente: Software Labview 2011

LabVIEW ofrece la opción de crear instrumentos virtuales, capaces de imitar tanto en apariencia como en operación a instrumentos físicos, se puede modificar dichos instrumentos virtuales y configurarlos a fin de que cumplan una tarea específica. Las diversas librerías de funciones y subrutinas ayudan a programar numerosas aplicaciones, además de las librerías específicas que disminuyen considerablemente la programación en el momento de usar la Adquisición de datos (DAQ), la interface de

bus de Propósito General (GPIB), comunicación Serial (VISA) o los módulos de Adquisición de Imágenes (IMAQ).

Entorno LabVIEW

El usuario interactúa con LabVIEW a través de tres partes principales: el Panel Frontal, acompañado del Diagrama de Bloques que no es más que el código fuente de los VIs, y el ícono del VI.

El **Panel Frontal** es la interface interactiva con el usuario, llamada así debido a que simula a un Panel de un instrumento físico. Como se aprecia en la Figura 2.32, contiene entradas y salidas. Los datos de entrada pueden ser ingresados a través del mouse, teclado gracias al uso de botones, controles, interruptores, etc. y los resultados mostrados en la pantalla a través de gráficos, leds, tanques, indicadores, etc.

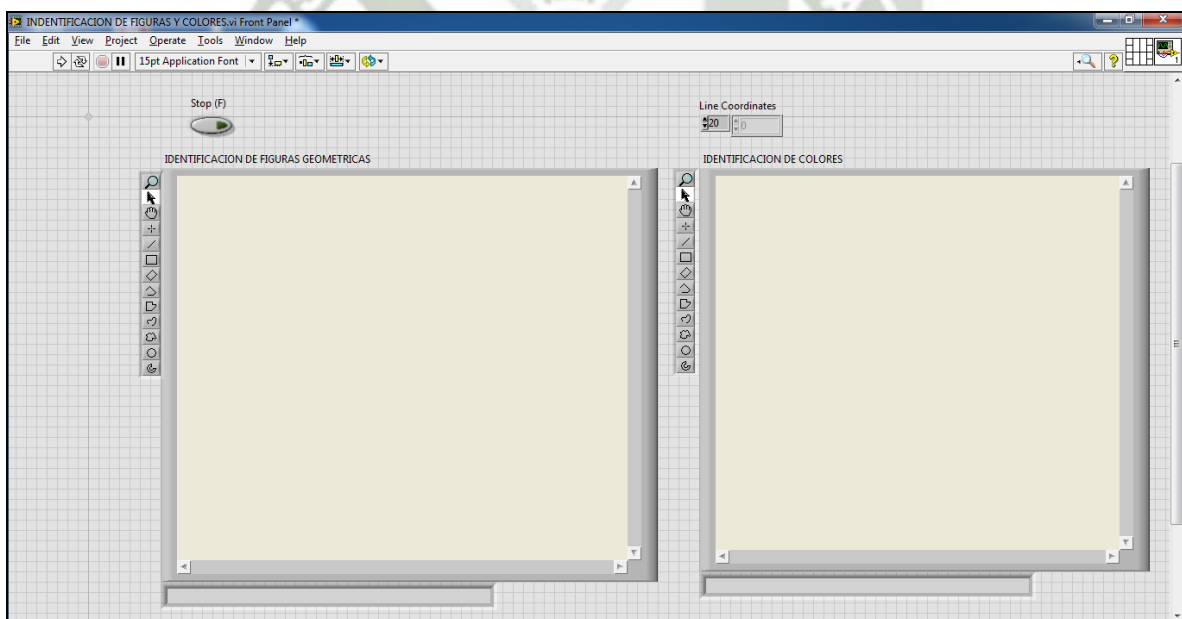


Figura 2. 32: Panel frontal
Fuente: Software Labview 2011

La ventana del **Diagrama de Bloques** es donde se ubican los objetos que en conjunto componen la programación gráfica que emplea Labview, de ésta manera se deja atrás a la programación tradicional que

utiliza líneas de texto de código. Como se aprecia en la Figura 2.23, gracias a este entorno gráfico se crean rápidamente interfaces de usuarios en donde se puede especificar la funcionalidad del sistema armando diagramas de bloques para implementar aplicaciones de control, automatización y medición.

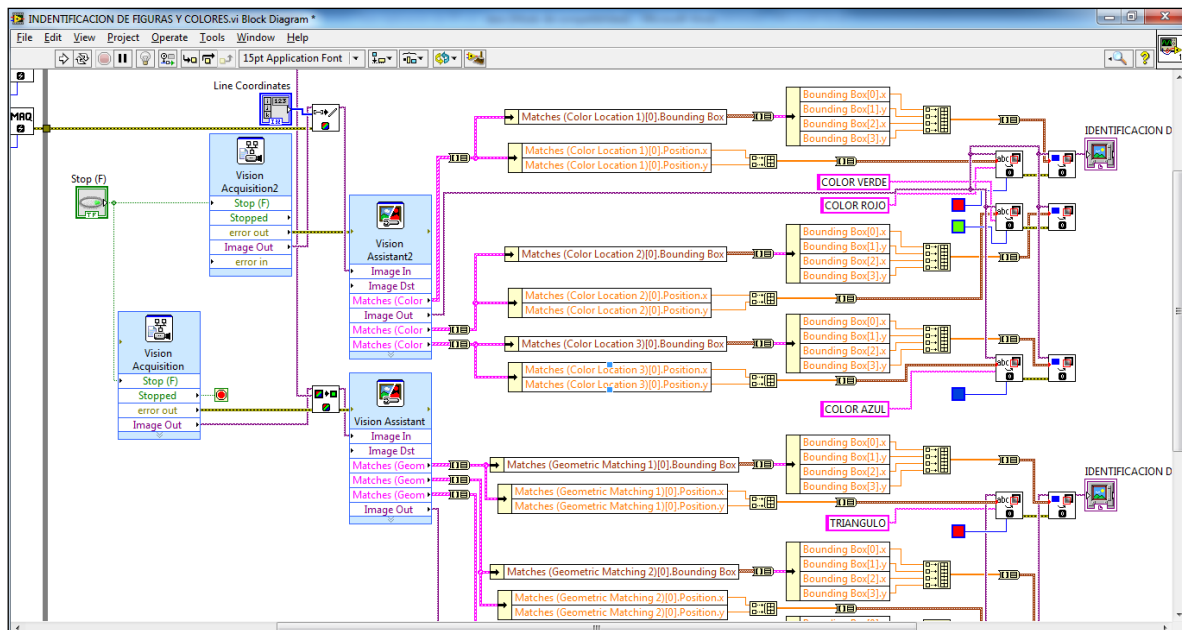


Figura 2. 33: Diagrama de bloques
Fuente: Software Labview 2011

Paletas de la ventana del panel frontal y diagrama de bloques.

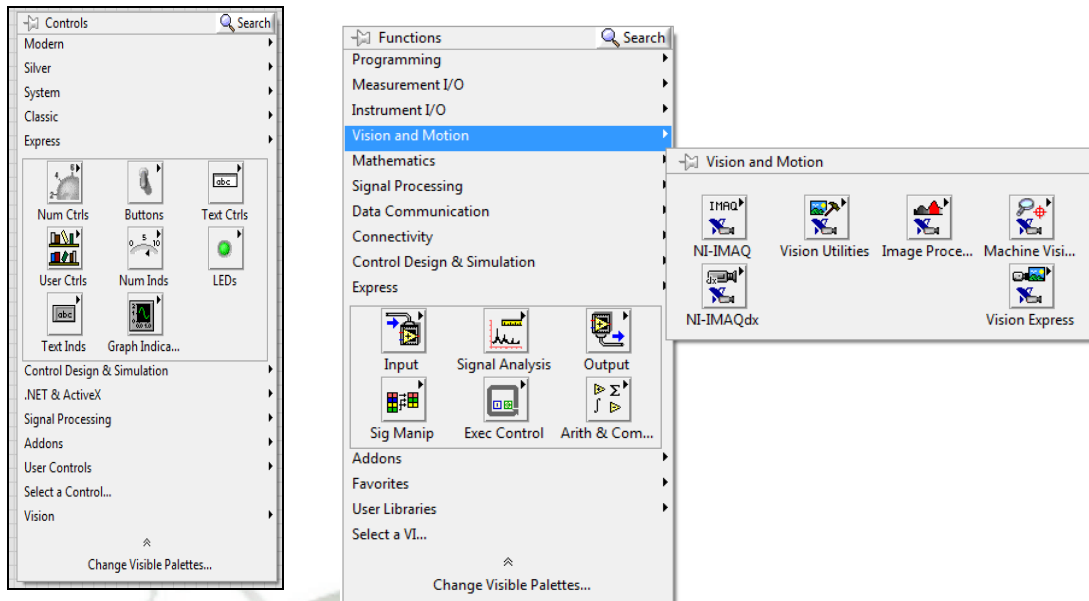


Figura 2. 34: Paletas
Fuente: Software Labview 2011

Ni visión

Bastante útil para el análisis y el procesamiento de imágenes, contiene una librería con más de 400 Vis. Posee una biblioteca de poderosas funciones para el procesamiento de visión y el visión assistant, un entorno interactivo para desarrolladores que necesitan generar rápidamente prototipos para aplicaciones o que necesitan inspecciones fuera de línea.

NI Vision es parte del Módulo de Desarrollo de Visión NI. La mayoría de los campos de la industria moderna, requieren para el análisis de sus tareas una herramienta que permita la combinación de adquisición de imágenes en movimiento, uso de visión robótica y la capacidad para el análisis de las mediciones tomadas de cada uno de los procesos, es así que Labview provee toda una gama de diversas funciones a fin de suplir todos estos requerimientos.

Labview soporta una gran cantidad de cámaras ya sean estas webcams, microscópicas, escáneres y muchas otras, todas estas divididas en 2 grandes grupos como lo son las cámaras análogas y digitales.

A continuación una breve descripción de las funciones que se incluyen en el módulo de Visión desarrollado por National Instruments para crear soluciones tecnológicas:

a) Vision and Motion: Provee una serie de paquetes que permiten la combinación de la visión artificial y la tecnología de control de movimiento, a continuación en la Figura 2.35 se muestra el menú de Visión and Movimiento

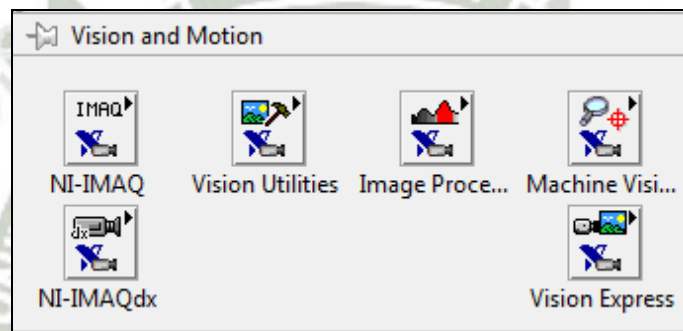


Figura 2. 35: Menú de funciones Vision and Motion
Fuente: Software Labview 2011

NI-IMAQ: Necesario para establecer el sistema de adquisición de imagen y adquirir imágenes. Los VIs incluidos en este paquete permiten abrir y cerrar una interface, a continuación se muestra el menú que ofrece en la Figura 2.36.

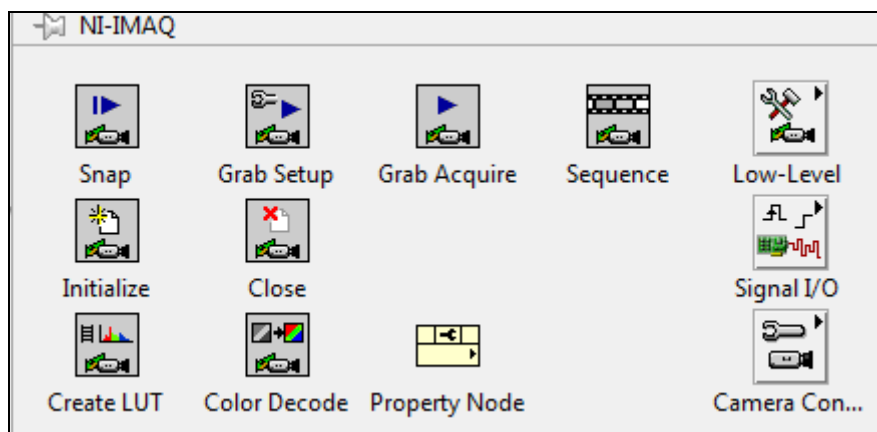


Figura 2. 36: Menú NI-IMAQ
Fuente: Software Labview 2011

Vision Utilities: Permite crear y manipular imágenes extraídas desde archivos, establecer las regiones de interés, calibración de imágenes, etc; en la Figura 2.37 se muestra el menú que proporciona este paquete.

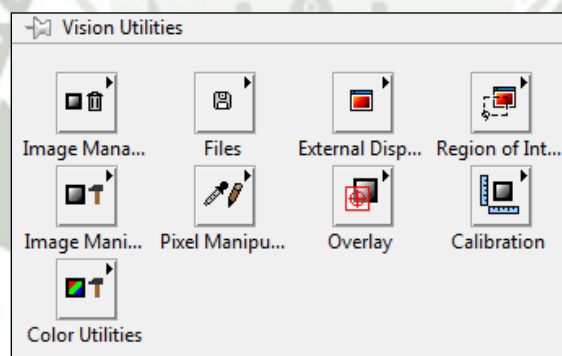


Figura 2. 37: Menú Vision Utilities
Fuente: Software Labview 2011

Image Processing: Usa los VIs de procesamiento de imágenes para analizar, filtrar, y tratar imágenes, la figura 2.38 muestra el menú de procesamiento de imágenes.

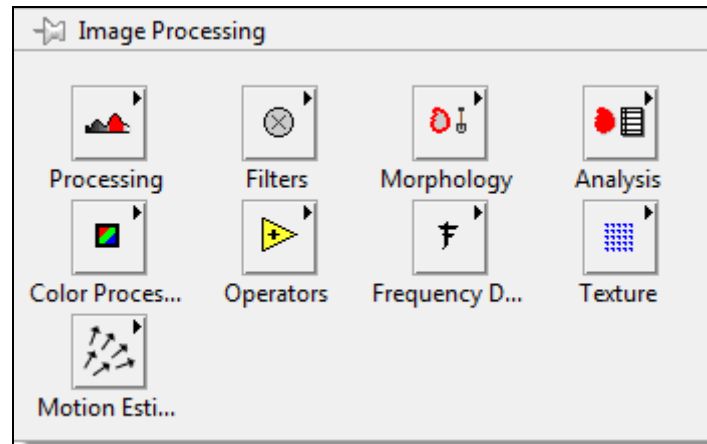


Figura 2. 38: Menú Image Processing
Fuente: Software Labview 2011

Machine Vision: Permite realizar tareas de inspección de visión de máquina comunes, incluyendo la presencia o la ausencia de partes en una imagen, medir dimensiones, localización de bordes, etc.

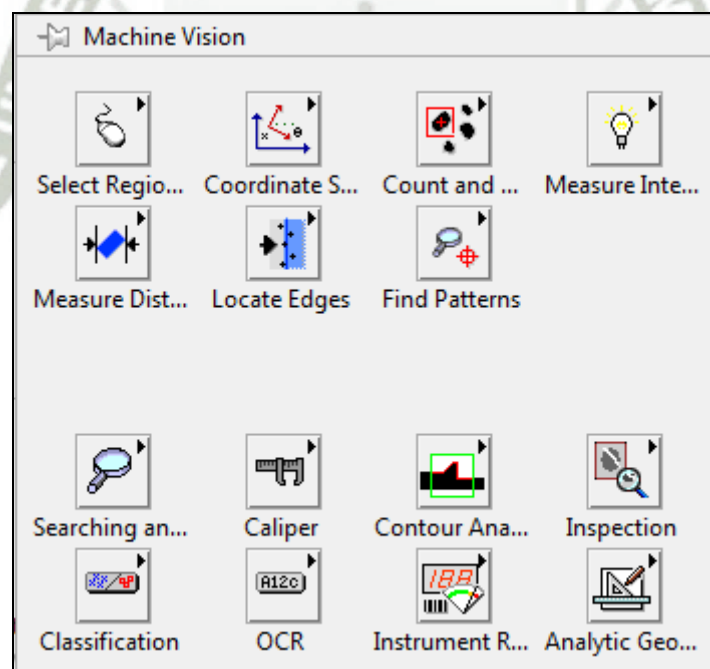


Figura 2. 39: Menú Machine Vision
Fuente: Software Labview 2011

NI-IMAQdx: Soporta cámaras GigE, IEEE 1394 y especificaciones de Instrumentación para Cámaras Digitales (IIDC), controla todos los modos disponibles y las características de las cámaras digitales.

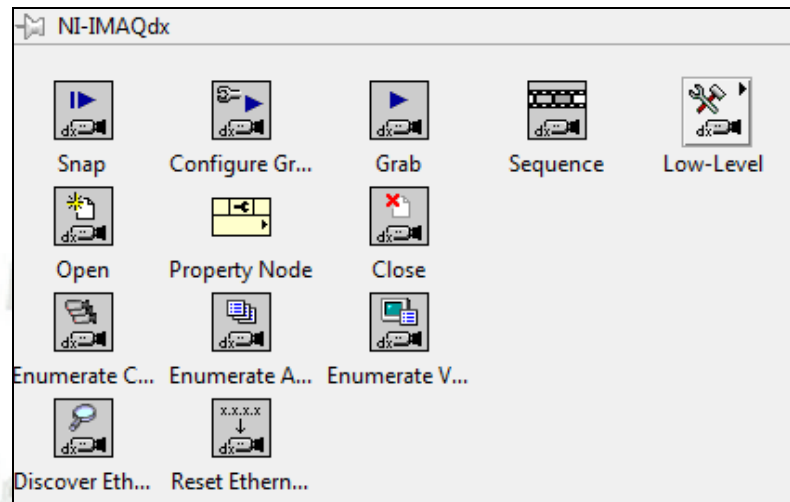


Figura 2. 40: Menú NI-IMAQdx
Fuente: Software Labview 2011

Vision Express: Permite desarrollar rápidamente la adquisición de imagen común y procesamiento de las mismas a través del uso de dos aplicaciones:

- Vision Acquisition: Ayuda a configurar fácilmente adquisiciones de cámaras análogas, digitales, Cámara link, IEEE 1394, y de Visión GigE.
- Vision Assistant: Permite realizar tareas de procesamiento de imágenes.

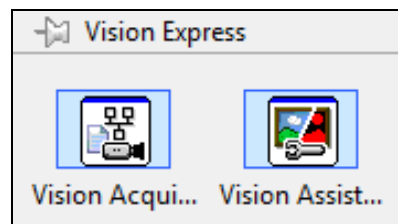


Figura 2. 41: Menú Visión Express
Fuente: Software Labview 2011

En la librería **Vision Utilities**, se encuentra la paleta de funciones de Image Management para crear, eliminar, definir parámetros, permitir la lectura y crear copias de una imagen.

La Figura 2.42 muestra las diferentes funciones necesarias para la administración de la imagen.

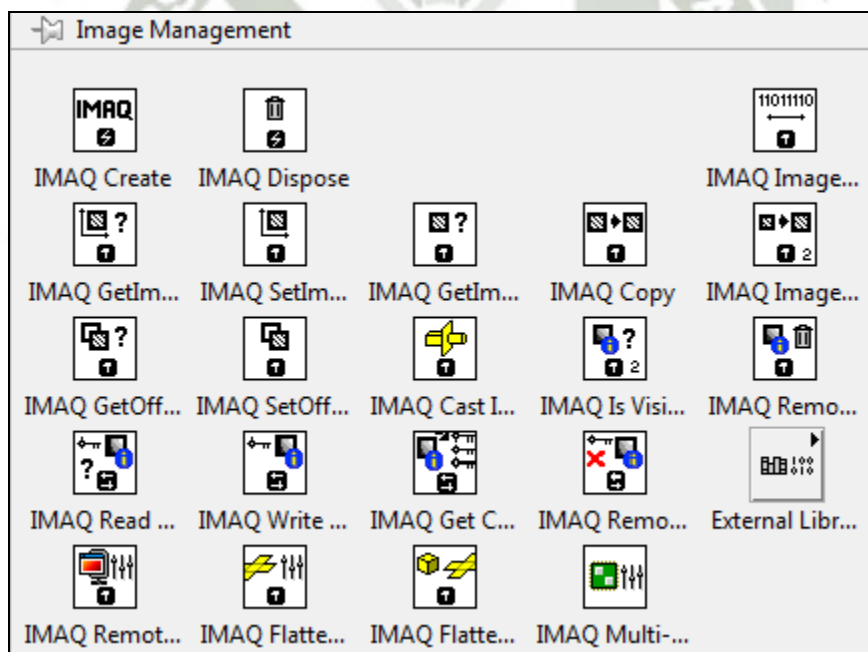


Figura 2. 42: Menú Image Management
Fuente: Software Labview 2011

Image Create, para crear una posición de memoria temporal para una imagen.

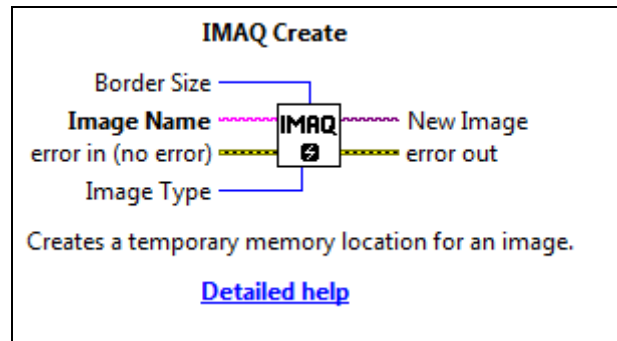


Figura 2. 43: Imaq Create
Fuente: Software Labview 2011

Dónde:

El tamaño del borde (Border Size): define el tamaño en pixeles, del borde a crear alrededor de una imagen. Se determina desde el inicio si una imagen va a ser procesada posteriormente usando las funciones que requieren uso de bordes, los bordes de una imagen nunca son mostrados o almacenados en un archivo.

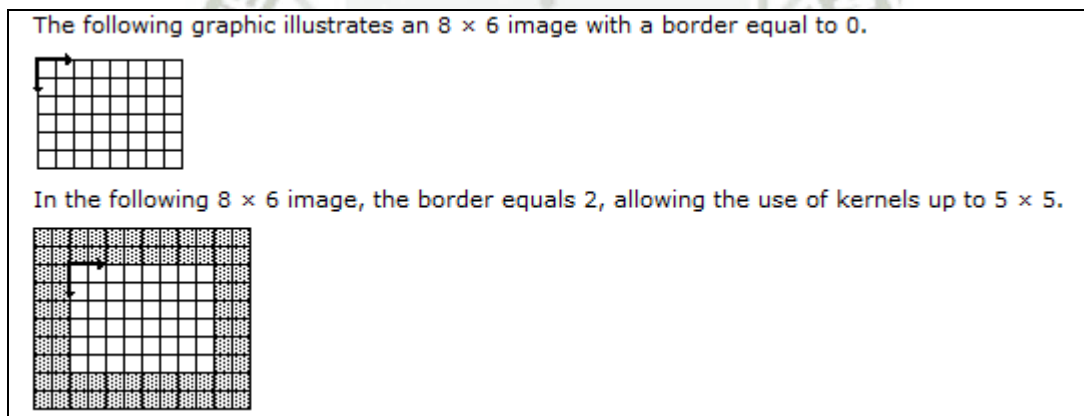


Figura 2. 44: Bordes de imagen
Fuente: Página web de National Instruments

El Nombre de la Imagen (Image Name): Es el nombre asociado a la imagen creada. Cada imagen creada tiene un nombre único. Error in describe el estado de error antes de ejecutar el VI o función. Por default no se tiene error.

El Tipo de Imagen (Image type): Especifica el tipo de imagen, puede ser: escala de grises, compleja, de color, etc. Imagen Nueva (New Image) es la salida de la Imagen que es utilizada como fuente de

muchas funciones usadas por NI Visión. Múltiples imágenes pueden ser creadas en una aplicación de LabVIEW.

Grayscale (U8) (0)	8 bits per pixel (unsigned, standard monochrome)
Grayscale (16) (1)	16 bits per pixel (signed)
Grayscale (SGL) (2)	32 bits per pixel (floating point)
Complex (CSG) (3)	2 × 32 bits per pixel (floating point)
RGB (U32) (4)	32 bits per pixel (red, green, blue, alpha)
HSL (U32) (5)	32 bits per pixel (hue, saturation, luminance, alpha)
RGB (U64) (6)	64 bits per pixel (red, green, blue, alpha)
Grayscale (U16) (7)	16 bits per pixel (unsigned, standard monochrome)

Figura 2. 45: Image type
Fuente: Página web de National Instruments

Tipos de datos

a) Tipo de Datos de Cadenas de Caracteres

Una cadena de caracteres es una secuencia de caracteres ASCII visibles y no visibles. Las cadenas de caracteres proporcionan un formato independiente de la plataforma para información y datos. Algunas aplicaciones más comunes de cadenas de caracteres incluyen las siguientes:

- Crear mensajes de texto simples.
- Controlar instrumentos al enviar comandos de texto al instrumento y regresando valores de datos en la forma de cadenas de caracteres binarios o ASCII, los cuales usted después convierte en valores numéricos.
- Almacenar datos numéricos en disco. Para almacenar datos numéricos en un archivo ASCII, primero debe convertir datos

numéricos en cadenas de caracteres antes de escribir los datos en un archivo de disco.

- Indicar o advertir al usuario con ventanas de diálogos.

En el panel frontal, las cadenas de caracteres aparecen como tablas, cuadros de texto y etiquetas. Labview incluye VIs integrados y funciones que usted puede usar para manipular cadenas de caracteres, incluyendo formato, análisis sintáctico o otras de edición. Labview representa datos de cadenas de caracteres con el color rosa.

b) Tipo de Datos Numéricos



Labview representa datos numéricos como números de punto flotante, números de punto fijo, enteros, enteros sin signo y números complejos. La precisión Doble y Sencilla, así como los datos numéricos Complejos son representados en Labview con el color naranja. Todos los datos numéricos son representados con el color azul.

La diferencia entre los tipos de datos numéricos es el número de bits que utilizan para almacenar datos y los valores de datos que representan.

Ciertos tipos de datos también ofrecen opciones de configuración extendida. Por ejemplo, usted puede asociar unidades físicas de medida con datos de punto flotante, incluyendo números complejos y usted puede configurar la codificación y el rango para datos de punto fijo.

c) Tipo de Datos Booleanos



Labview almacena datos Booleanos como valores de 8 bits. Un Booleano puede usarse en Labview para representar 0 o 1, o un TRUE o FALSE. Si el valor de 8 bits es cero, el valor Booleano es FALSE. Cualquier valor que no sea cero representa a TRUE. Las aplicaciones comunes para datos Booleano incluyen representar datos digitales y funcionar como un control de panel frontal que actúa como un interruptor que tiene una acción mecánica, generalmente usada para controlar una estructura de ejecución como una estructura de Caso. Un control Booleano generalmente es usado como expresión condicional para salir de un Ciclo While. En LabVIEW, el color verde representa datos Booleano.

Arrays

Los arrays son conjuntos de datos o elementos del mismo tipo, accesibles mediante los índices del propio array. En LabVIEW existen arrays unidimensionales, y también los bidimensionales. El array es en realidad una tabla de dos dimensiones, donde los elementos guardados en sus casillas son accesibles mediante los índices de la tabla, conocidos como filas y columnas.

Hay dos formas de inicializar los arrays. La primera es utilizando las funciones específicas de inicialización de arrays, y la otra, mediante las sentencias repetitivas (for, while).

a) Arrays unidimensionales

Los arrays unidimensionales son aquellos que tienen una única fila y C columnas, es decir son arrays de tamaño 1xC. Para inicializar un array unidimensional utilizando las funciones de arrays, se ejecuta en el diagrama Functions/Programming/Array/Build Array.

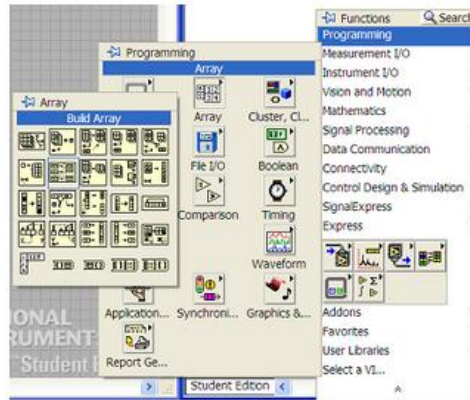


Figura 2. 46: Build Array bloques
Fuente: Software Labview 2011

A continuación se marca ese objeto con el ratón y se extiende hacia abajo dándole el tamaño de C columnas, p.e. 3 columnas. Seguidamente se generan las tres entradas (desde las 3 entradas del objeto Build Array), es decir, los controles que proporcionarán los valores de los 3 elementos del array, y también el indicador que muestra el contenido del array (desde la salida del objeto Build Array).



Figura 2. 47: Build array
Fuente: Software Labview 2011

A esto, le añadimos una sentencia while para que la aplicación se ejecute de forma continua. Al ejecutar la aplicación, se asignan los valores de los elementos del array en los controles correspondientes. El array puede ser indexado desde su índice (elemento de control del array), es decir, cada elemento del array se puede ver desde el su índice. Como se puede observar en figura, el índice comienza en 0, y no en 1.

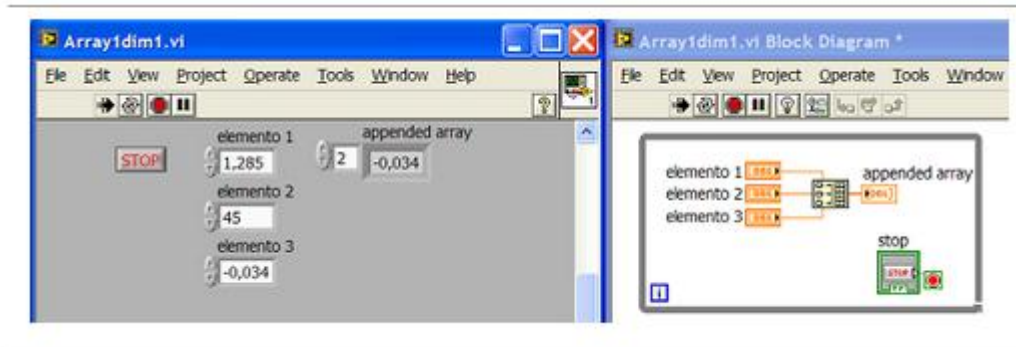


Figura 2. 48: Ejemplo de array
Fuente: Software Labview 2011

b) Arrays bidimensionales

Se trata de arrays de más de una fila, con lo que se consiguen arrays de F filas y C columnas, es decir arrays $F \times C$. Como se puede deducir, un array de dos dimensiones es ideal para guardar o representar tablas de datos, ya que la tabla tiene la misma organización bidimensional de filas y columnas que el array bidimensional.

En cuanto a la inicialización de estos arrays, tenemos las dos formas explicadas que para los arrays unidimensionales. Por una parte, se pueden inicializar utilizando las funciones Build Array, y por otra, mediante las sentencias de repetición for. Como hemos podido ver con los arrays de una dimensión, una función Build Array crea un array de una única fila. Pues bien, para crear un array de dos dimensiones, lo que se hace es unir varios arrays de una dimensión en un array mediante otra función Build Array. La única condición que han de cumplir estos arrays unidimensionales, es que todos ellos tengan el mismo número de elementos, es decir, que el mismo número de columnas.

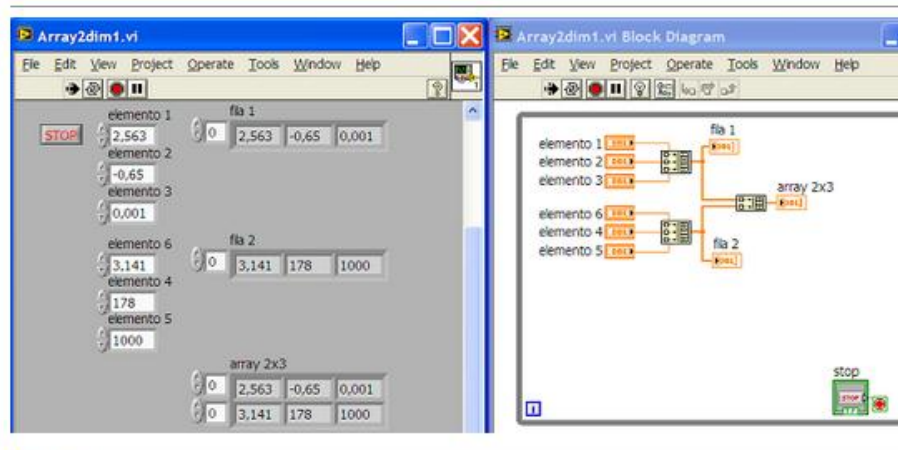


Figura 2. 49: Ejemplo Build array
Fuente: Software Labview 2011

Cluster

Los clusters son tipos de datos compuestos por varios elementos de distintos tipos. Son los equivalentes a las estructuras de datos del lenguaje de programación C y similares, o a las clases, en los lenguajes de Programación Orientados a Objetos, como el C++, Java, etc. En LabVIEW, cuando interesa unir varios tipos de datos bajo el mismo nombre, entonces se crea un cluster con ese nombre. Esto, suele ser muy práctico para que en el diagrama, aparezcan muchas menos líneas de conexiones y éstas estén agrupadas según un criterio dado. En el ejemplo que viene a continuación, se muestra cómo tres tipos de datos distintos (número real, número entero y cadena de caracteres) que se encuentran en los controles de la izquierda son llevados a la parte derecha del diagrama como una única unidad, figura 2.50 b) derecha. Para agruparlos, se utiliza el objeto Bundle, figura 2.50 a)., que se encuentra en Functions/Programming/Cluster, Class & Variants. En cambio, en el lado derecho, tanto del diagrama como del panel, se encuentran los indicadores que permiten visualizar los valores enviados desde los controles antes mencionados. Para ello, se utiliza el objeto

Unbundle, figura 2.50 derecha., cuya misión es obtener los elementos individuales del grupo que llega a su entrada.

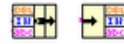


Figura 11

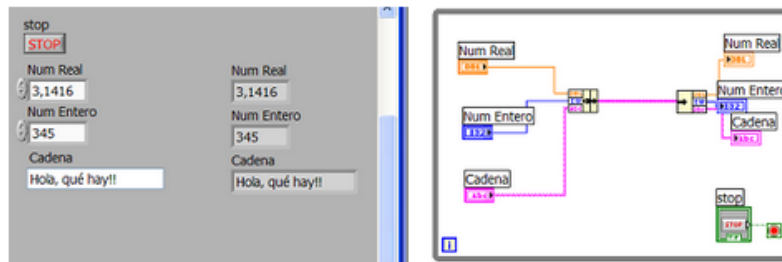


Figura 12

Figura 2. 50: Ejemplo Build array
Fuente: Software Labview 2011

2.4.2. Servomotor Tower Pro sg 5010

Un Servo es un dispositivo pequeño que tiene un eje de rendimiento controlado. Este puede ser llevado a posiciones angulares específicas al enviar una señal codificada. Con tal de que una señal codificada exista en la línea de entrada, el servo mantendrá la posición angular del engranaje. Cuando la señal codificada cambia, la posición angular de los piñones cambia. En la práctica, se usan servos para posicionar superficies de control como el movimiento de palancas, pequeños ascensores y timones. Ellos también se usan en radio control, títeres, y por supuesto, en robots.

Los Servos son sumamente útiles en robótica. Los motores son pequeños, tiene internamente una circuitería de control interna y es sumamente poderoso para su tamaño. Un servo normal o Standard como el HS-300 de Hitec tiene 42 onzas por pulgada o mejor 3kg por cm. De torque que es bastante fuerte para su tamaño. También potencia proporcional para cargas mecánicas. Un servo, por consiguiente, no consume mucha energía. Se muestra la composición interna de un servo motor en el cuadro de abajo. Podrá observar la circuitería de control, el

motor, un juego de piñones, y la caja. También puede ver los 3 alambres de conexión externa. Uno es para alimentación Vcc (+5volts), conexión a tierra GND y el alambre blanco es el alambre de control.



Figura 2. 51: Servomotor
Fuente: Servomotor Futaba

Funcionamiento

El motor del servo tiene algunos circuitos de control y un potenciómetro (una resistencia variable) esta es conectada al eje central del servo motor. En la figura se puede observar al lado derecho del circuito. Este potenciómetro permite a la circuiteria de control, supervisar el ángulo actual del servo motor. Si el eje está en el ángulo correcto, entonces el motor está apagado. Si el circuito chequea que el ángulo no es el correcto, el motor girará en la dirección adecuada hasta llegar al ángulo correcto. El eje del servo es capaz de llegar alrededor de los 180 grados. Normalmente, en algunos llega a los 210 grados, pero varía según el fabricante. Un servo normal se usa para controlar un movimiento angular de entre 0 y 180.

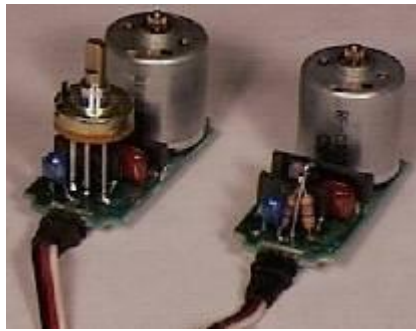


Figura 2. 52: Parte interna servomotores
Fuente: Servomotor Futaba

La cantidad de voltaje aplicado al motor es proporcional a la distancia que éste necesita viajar. Así, si el eje necesita regresar una distancia grande, el motor regresará a toda velocidad. Si este necesita regresar sólo una pequeña cantidad, el motor correrá a una velocidad más lenta. A esto se le llama control proporcional.

Características generales y funcionamiento:

Estos servos tienen un amplificador, servo motor, piñonaría de reducción y un potenciómetro de realimentación; todo incorporado en el mismo conjunto. Esto es un servo de posición (lo cual significa que uno le indica a qué posición debe ir), con un rango de aproximadamente 180 grados. Ellos tienen tres cables de conexión eléctrica; Vcc, GND, y entrada de control.

Para controlar un servo, usted le ordena un cierto ángulo, medido desde 0 grados. Usted le envía una serie de pulsos. En un tiempo ON de pulso indica el ángulo al que debe posicionarse; 1ms = 0 grados, 2.0ms = máx. Grado (cerca de 120) y algún valor entre ellos da un ángulo de salida proporcional. Generalmente se considera que en 1.5ms está el "centro." Entre límites de 1 ~ 2ms son las recomendaciones de los fabricantes, usted normalmente puede usar un rango mayor de 1.5ms para obtener un ángulo mayor e incluso de 2ms para un ángulo de rendimiento de 180 grados o más. El factor limitante es el tope del potenciómetro y los

límites mecánicos contruidos en el servo. Un sonido de zumbido normalmente indica que usted está forzando por encima al servo, entonces debe disminuir un poco.

Este es un ejemplo de la señal que debería tener el servo:

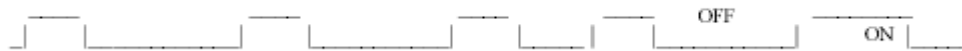


Figura 2. 53: Señales para el servomotor
Fuente: servomotor Futaba

El tiempo de OFF en el servo no es crítico; puede estar alrededor de los 20ms. Hemos usado entre 10ms y 30 ms. Esto No tiene que ser de ésta manera, puede variar de un pulso a otro. Los pulsos que ocurren frecuentemente en el tiempo de OFF pueden interferir con el sincronismo interno del servo y podría escucharse un sonido de zumbido o alguna vibración en el eje. Si el espacio del pulso es mayor de 50ms (depende del fabricante), entonces el servo podría estar en modo SLEEP entre los pulsos. Entraría a funcionar en pasos pequeños y el rendimiento no sería el óptimo.

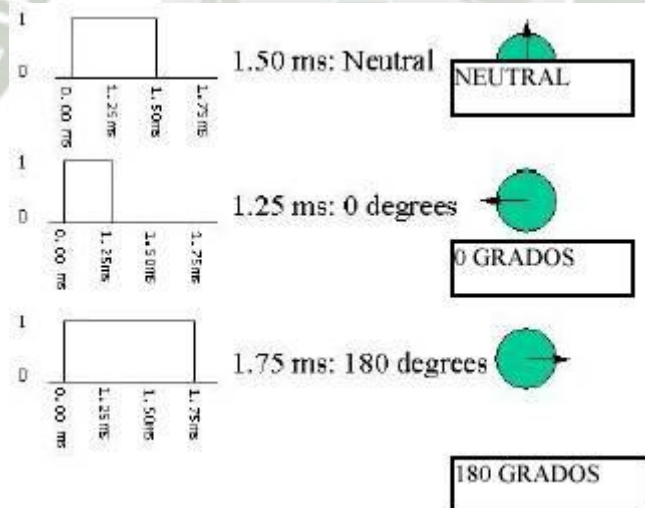


Figura 2. 54: Señales para el servomotor
Fuente: Servomotor Futaba

Como se observa en la figura, la duración del pulso indica o dictamina el ángulo del eje (mostrado como un círculo verde con flecha). Nótese que las ilustraciones y los tiempos reales dependen del fabricante de motor. El principio, sin embargo, es el mismo.

El cable de control se usa para comunicar el ángulo. El ángulo está determinado por la duración de un pulso que se aplica al alambre de control. A esto se le llama PCM Modulación codificada de Pulsos. El servo espera ver un pulso cada 20 milisegundos (.02 segundos). La longitud del pulso determinará los giros de motor. Un pulso de 1.5 ms., por ejemplo, hará que el motor se torne a la posición de 90 grados (llamado la posición neutra). Si el pulso es menor de 1.5 ms., entonces el motor se acercará a los 0 grados. Si el pulso es mayor de 1.5ms, el eje se acercará a los 180 grados.

La descripción realizada anteriormente como se ha podido observar son de servomotores de corriente continua usados en robótica doméstica y en aeromodelismo fundamentalmente.

Servomotores en modelismo:

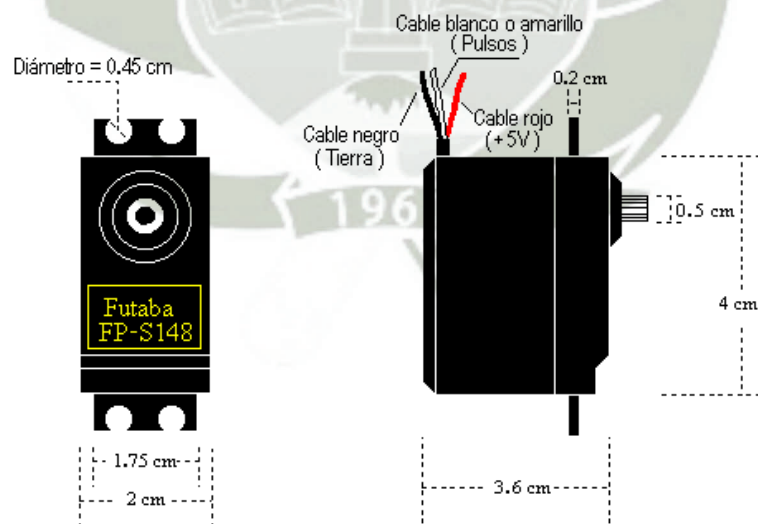


Figura 2. 55: Servomotor en modelismo
Fuente: Servomotor Futaba

Diagrama de un servomotor típico de modelismo.

Un servomotor de este tipo es básicamente un motor eléctrico que sólo se puede girar en un ángulo de aproximadamente 180 grados (no dan vueltas completas como los motores normales). De los tres cables que salen de su cubierta. El rojo es de **voltaje de** alimentación (+5V), el negro es de tierra (0V ó GND). El cable blanco (a veces amarillo) es el cable por el cuál se le instruye al servomotor en qué posición ubicarse (entre 0 grados y 180).

Dentro del servomotor, una tarjeta controladora le dice a un pequeño motor de corriente directa cuántas vueltas girar para acomodar la **flecha** (el eje de plástico que sale al exterior) en la posición que se le ha pedido.

En la siguiente figura se observa la ubicación de estas piezas dentro del servomotor:

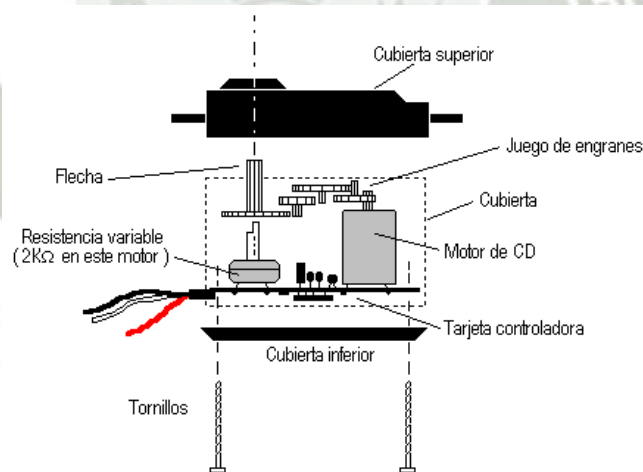


Figura 2. 56: Partes del servomotor
Fuente: Servomotor Futaba

Un potenciómetro que está sujeto a la flecha, mide hacia dónde está ubicado en todo momento. Es así como la tarjeta controladora sabe hacia dónde mover al motor.

La posición deseada se le da al servomotor por medio de pulsos. Todo el tiempo debe haber una señal de pulsos presente en ese cable.

La señal de pulsos controla al servo de la siguiente forma:

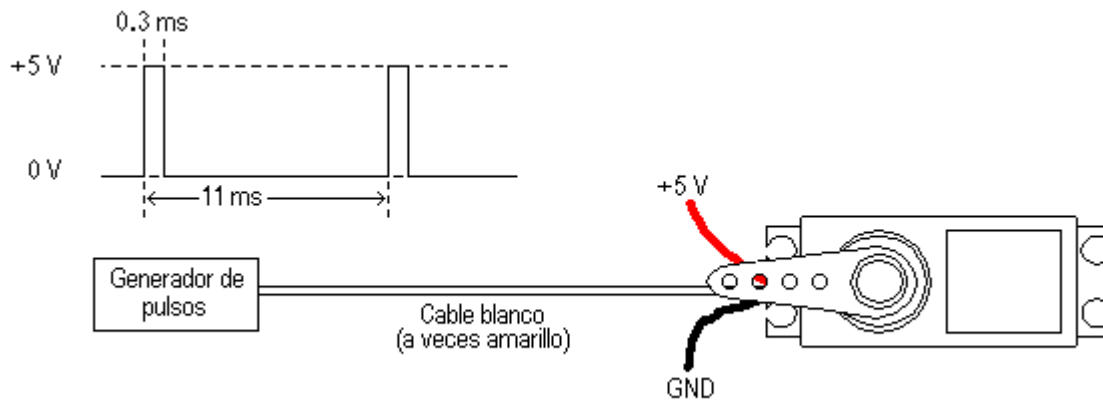


Figura 2. 57: Señal de pulsos
Fuente: Servomotor Futaba

Nótese que el intervalo de tiempo entre pulsos se mantiene constante, y la variación del ancho de los mismos es lo que le indica al servo la posición que se desea. Hemos encontrado también que son bastante tolerables en cuanto al período de los pulsos de control. Responden adecuadamente a pulsos desde 50 hz. hasta aproximadamente 100 hz., pero una vez escogida una frecuencia de operación debe procurarse mantener la misma frecuencia todo el tiempo.

Especificación del servomotor utilizado

TOWER PRO SERVO SPECIFICATION									
Type	Dimension	Weight	Stall torque	Operating speed	Operating voltage	Temperature rang	Dead band width	Gear type	apply for
SG50	21.5x11.7x25.1mm	5	0.6kg/cm(4.8V)	0.1sec/60degree(4.8v)	4.8V	0°C_ 55°C	10µs	nylon	Helicopter,3D-flyer,F3A
SG51R	21.5x11.7x25.1mm	5	0.7kg/cm(4.8V)	0.1sec/60degree(4.8v)	4.8V	0°C_ 55°C	10µs	nylon	
SG90	23x12.2x29mm	9g	1.8kg/cm(4.8V)	0.1sec/60degree(4.8v)	4.8V	0°C_ 55°C	10µs	nylon	
SG91R	23x12.2x29mm	9g	1.8kg/cm(4.8V)	0.1sec/60degree(4.8v)	4.8V	0°C_ 55°C	10µs	nylon	
SG5010	40.2x20.2x43.2mm	38g	5.5kg/cm(4.8V) 6.5kg/cm(6V)	0.2sec/60degree(4.8v) 0.16sec/60degree(6v)	4.8-6V	0°C_ 55°C	10µs	nylon	cars&airplane
MG945	40.7*19.7*42.9mm	55g	10kg/cm(4.8V), 12kg/cm(6V)	0.23sec/60degree(4.8v) 0.2sec/60degree(6.0v)	4.8-7.2V	0°C_ 55°C	5µs	metal	1/8 buggy, cars
MG945R	40.7*19.7*42.9mm	55g	10kg/cm(4.8V), 12kg/cm(6V)	0.23sec/60degree(4.8v) 0.2sec/60degree(6.0v)	4.8-7.2V	0°C_ 55°C	5µs	metal	1/10 buggy, cars
MG995	40.7*19.7*42.9mm	55g	8.5kg/cm(4.8V), 10kg/cm(6V)	0.20sec/60degree(4.8v) 0.16sec/60degree(6.0v)	4.8-7.2V	0°C_ 55°C	5µs	metal	
MG995R	40.7*19.7*42.9mm	55g	8.5kg/cm(4.8V), 10kg/cm(6V)	0.20sec/60degree(4.8v) 0.16sec/60degree(6.0v)	4.8-7.2V	0°C_ 55°C	5µs	metal	
9805BB	66x30.2x64.4mm	160g	20kg/cm(4.8V) 25kg/cm(6V)	0.20sec/60degree(4.8v) 0.16sec/60degree(6v)	4.8-7.2V	0°C_ 55°C	5µs	nylon	Cars
MG16R	29*11.2*29mm	18.8g	2.7kg.cm(4.8v) 2.9kg/cm(6.0)	0.10sec/60degree(4.8v) 0.08sec/60degree(6v)	4.8v-6V	0°C_ 55°C	5µs	metal	Helicopter
MG946R	40.7*19.7*42.9mm	55g	10.5kg/cm(4.8V), 13kg/cm(6V)	0.20sec/60degree(4.8v) 0.17sec/60degree(6.0v)	4.8-7.2V	0°C_ 55°C	5µs	metal	Gasoline engine plane, train-flyer
MG996R	40.7*19.7*42.9mm	55g	9.4kg/cm(4.8V), 11kg/cm(6V)	0.17sec/60degree(4.8v) 0.14sec/60degree(6v)	4.8-7.2V	0°C_ 55°C	5µs	metal	Helicopter
MG955H	40.7*19.7*42.9mm	55g	7.0kg/cm(4.8V), 8.5kg/cm(6V)	0.17sec/60degree(4.8v) 0.14sec/60degree(6v)	4.8-7.2V	0°C_ 55°C	5µs	metal	
MG956R	40.7*19.7*42.9mm	55g	7.5kg/cm(4.8V), 9kg/cm(6V)	0.15sec/60degree(4.8v) 0.12sec/60degree(6v)	4.8-7.2V	0°C_ 55°C	5µs	metal& 1 nylon gear	

Marks: "R" means RoHS, It is a material standard for European area ,means non-toxic,non-leaded,no harm to environment

Tabla 2.2: Especificación servo motor Tower pro [2]
Fuente: Data sheet servo motor Tower Pro sg5010



CAPITULO III

MARCO EXPERIMENTAL

3.1. ADQUISICIÓN DE IMÁGENES

Labview con su Módulo de Desarrollo de Visión fue el entorno elegido para realizar la aplicación, debido a que provee un entorno de arquitectura abierta y una interface amigable.

El Módulo de Desarrollo de Visión (Visión Development Module), contiene una serie de funciones con las que se puede detectar objetos, mejorar las características de las imágenes, localizar determinadas particularidades, piezas o realizar mediciones en una determinada imagen. Además de las bibliotecas, este módulo incluye el Asistente de Visión y el Software de Adquisición de National Instruments (NI).

Entre las bondades de éste módulo están:

- Ofrecer controladores para miles de cámaras, incluyendo GigE Vision y cámaras IEEE 1394
- Subpixel con exactitud hasta 0.1 de un pixel y 0.1 de un grado

Para fines de pruebas se pretende capturar las imágenes por cámara USB, para que posteriormente la aplicación la procese y de esta manera se determine el color o figura geométrica en tiempo real. Se controlará la iluminación y el fondo a fin de obtener el mayor contraste posible para precisamente resaltar la característica de interés (color o forma).

Vision Acquisition Software (VAS) es un conjunto de controladores y utilidades con funciones para adquirir, mostrar y guardar imágenes desde cualquier captador NI marco, cámara GigE Vision o IEEE 1394 (Firewire) de la cámara. VAS se incluye tanto en VDM y VBAI y es un conjunto de tres conductores.

Puede llevar a cabo la adquisición de imágenes de base, la pantalla y guardar las aplicaciones con sólo los drivers incluidos en las VAS, pero para cualquier tratamiento o manipulación usted necesitará VBAI o VDM, además de VAS.

Ni imaqdx

Imaqdx Open Camera, se requiere una cámara para esta capacidad, un archivo de cámara configurable, y crea una única referencia de la cámara. Use imaqdx close camera.vi cuando se haya terminado con el proceso.

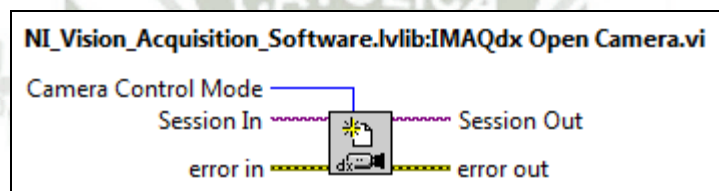


Figura 3. 1: IMAQdx Open Camera.vi

Fuente: Software Labview 2011

Imaqdx Configure Grab, configura la cámara y la prepara para la adquisición de imágenes

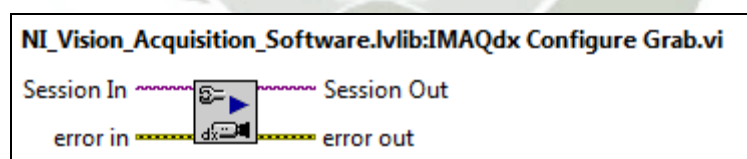


Figura 3. 2: IMAQdx Configure Grab .vi

Fuente: Software Labview 2011

Imaqdx Grab, adquiere continuamente una imagen de la cámara seleccionada. Se utiliza este modo para adquirir imágenes a altas velocidades.

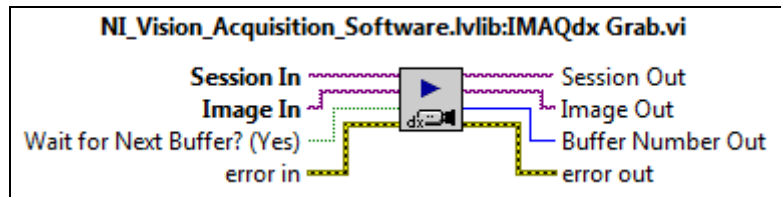


Figura 3. 3: IMAQdx Grab.vi
Fuente: Software Labview 2011

Imaqdx Close Camera, permite terminar la adquisición de imágenes, y cerrar la sesión con la cámara especificada.

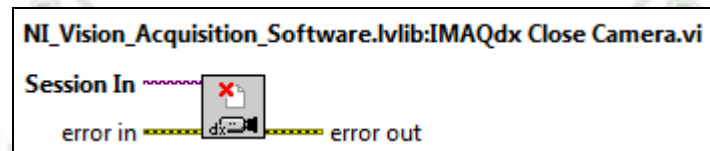


Figura 3. 4: IMAQdx Close Camera.vi
Fuente: Software Labview 2011

Ejemplo de captación con cámara

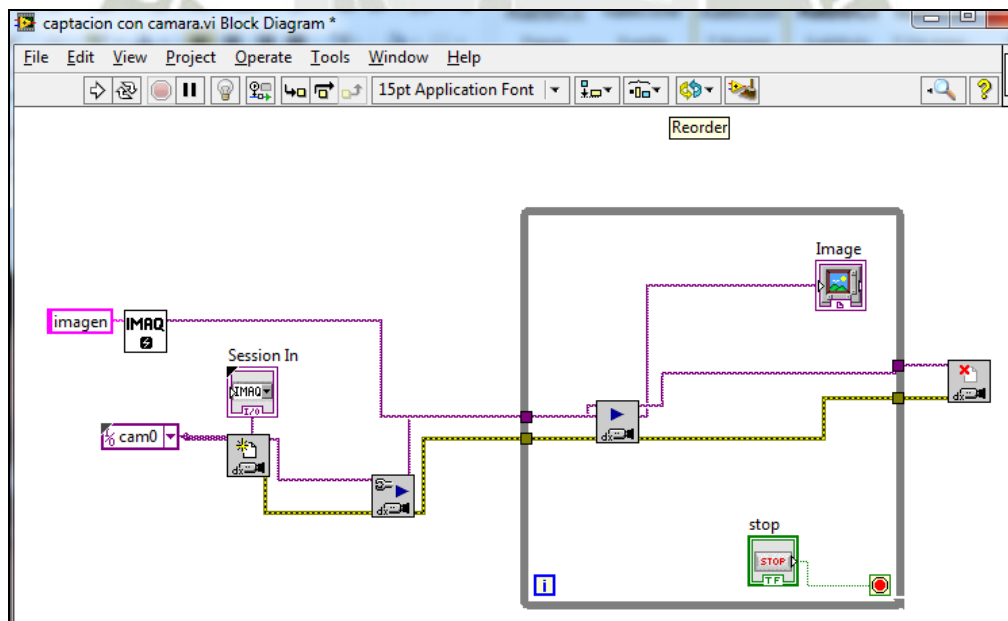


Figura 3. 5: Diagrama de bloques
Fuente: Elaboración propia

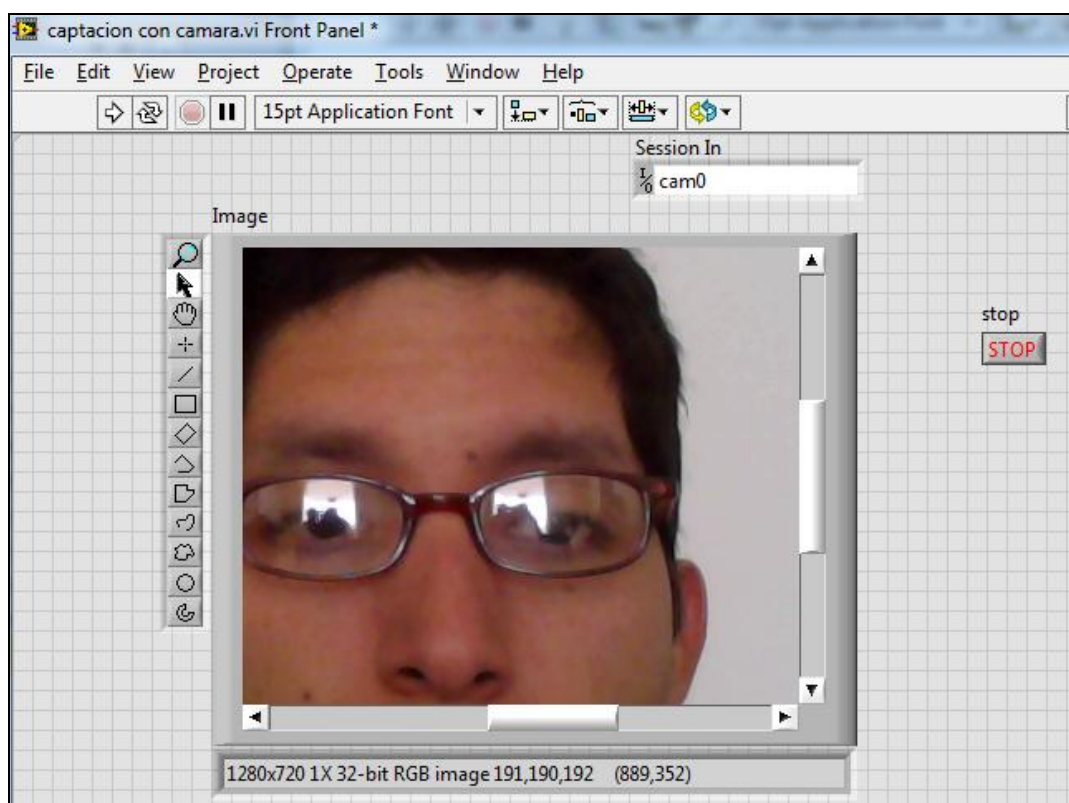


Figura 3. 6: Panel frontal
Fuente: Elaboración propia

PRIMER PASO

Visión acquisition

En la ventana de diagrama de bloques hacer click derecho y seleccionar del menú de visión express, el icono visión acquisition, el cual va a permitir captar imágenes en tiempo real y de forma continua.

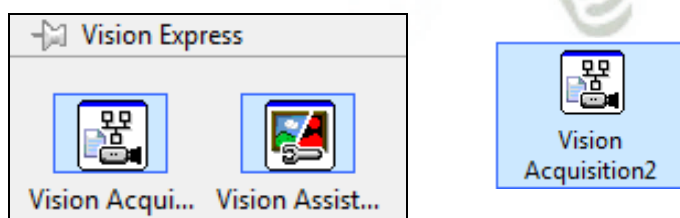


Figura 3. 7: Vision Express
Fuente: Software Labview 2011

En este primer paso se identifica que cámara se va utilizar, como podemos observar puede ser la cam0: que es la cámara integrada de la lapto y si utilizamos una cámara web con usb, se apreciara un icono debajo de la cam0, que vendría ser la cam1 (por USB). Se escoge la cámara a utilizar y se presiona play para certificar que la cámara funciona, luego next.

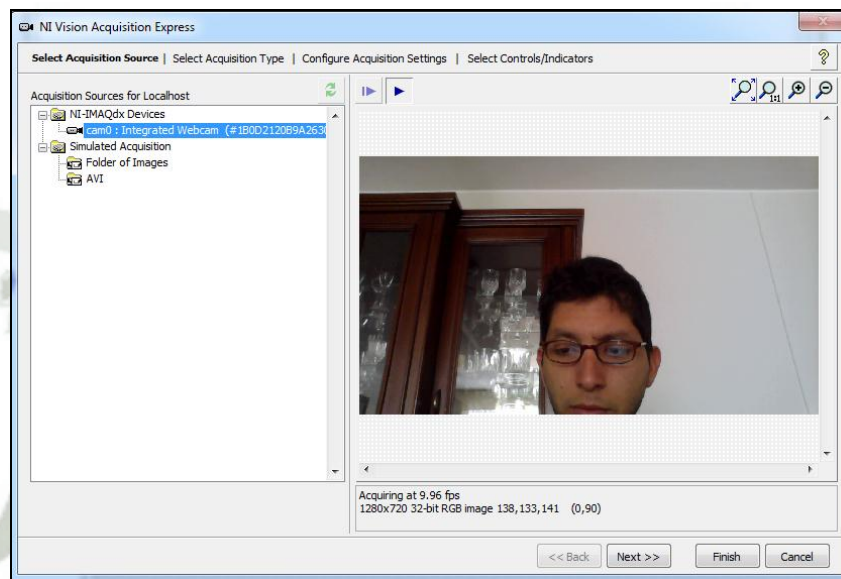


Figura 3. 8: Vision Acquisition
Fuente: Elaboración propia

Seleccionar Continuous Acquisition with inline processing (Adquisición continua en proceso), luego next.

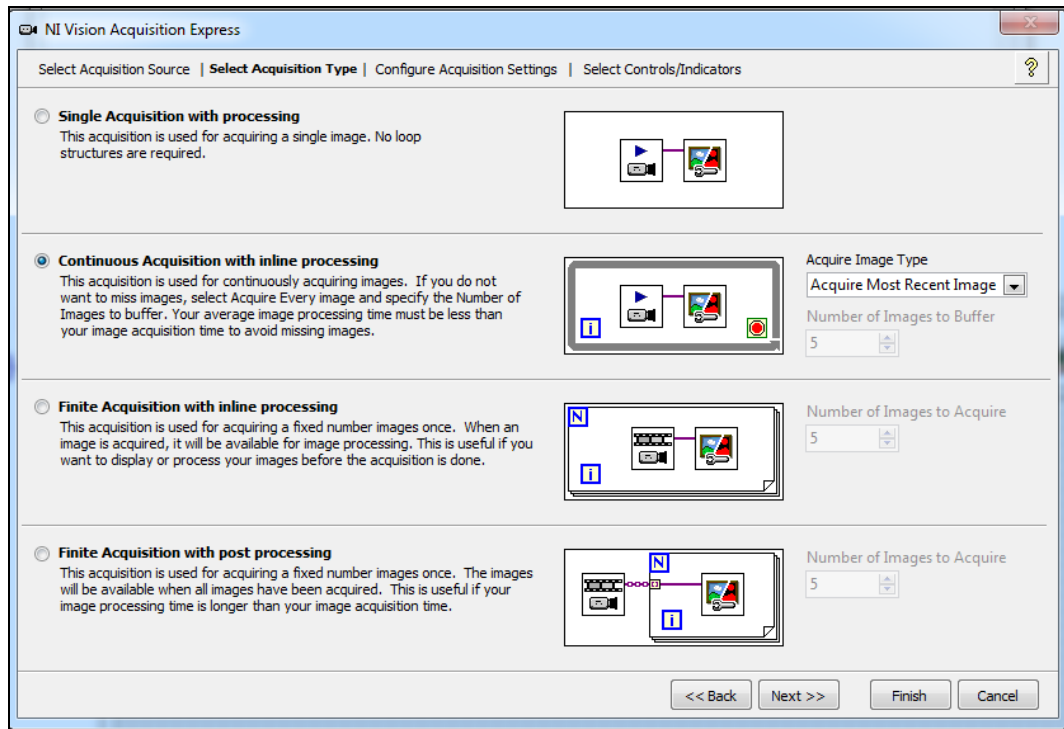


Figura 3. 9: Select Acquisition Type
Fuente: Elaboración propia

Se observa las opciones de la cámara, si se desea realizar alguna modificación, presionar next.

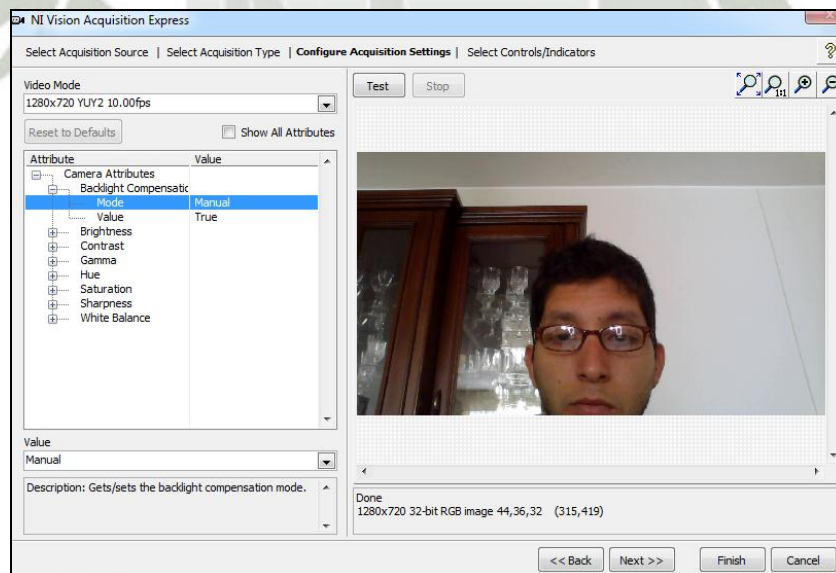


Figura 3. 10: Configure Acquisition Settings
Fuente: Elaboración propia

Seleccionar controles e indicadores a aparecer. Por ultimo presionar finish.

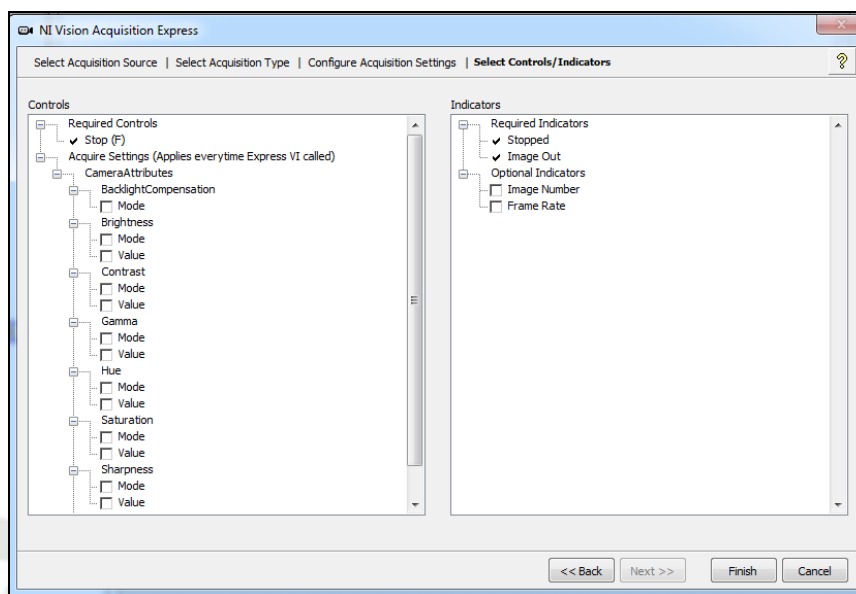


Figura 3. 11: Select Controls/Indicators
Fuente: Elaboración propia

Se creara un diagrama de bloques para la adquisición de imágenes, con unos iconos por defecto.

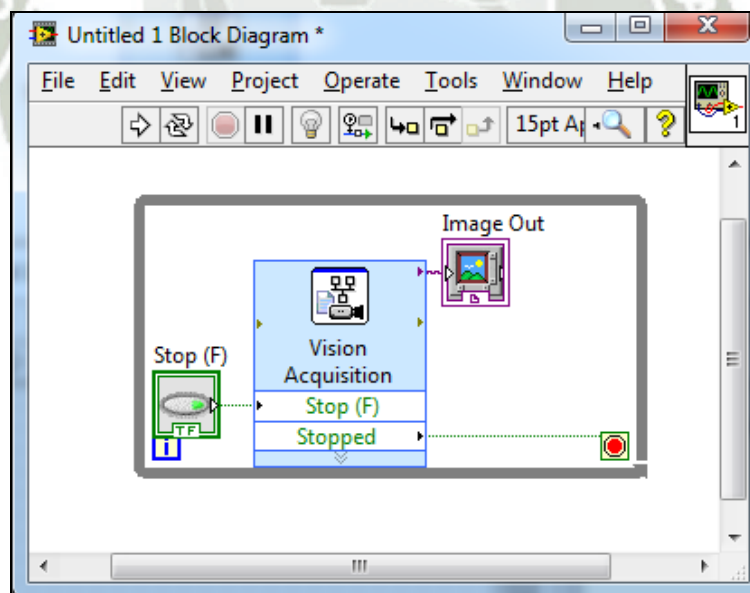


Figura 3. 12: Diagrama de bloques
Fuente: Elaboración propia

En este diagrama de bloques se observa un stop el cual nos permitirá detener la adquisición de imágenes, image out donde podremos visualizar las imágenes y un ciclo while, el cual realizara acciones repetitivas hasta que presionar stop.

En el panel frontal se crea lo siguiente:

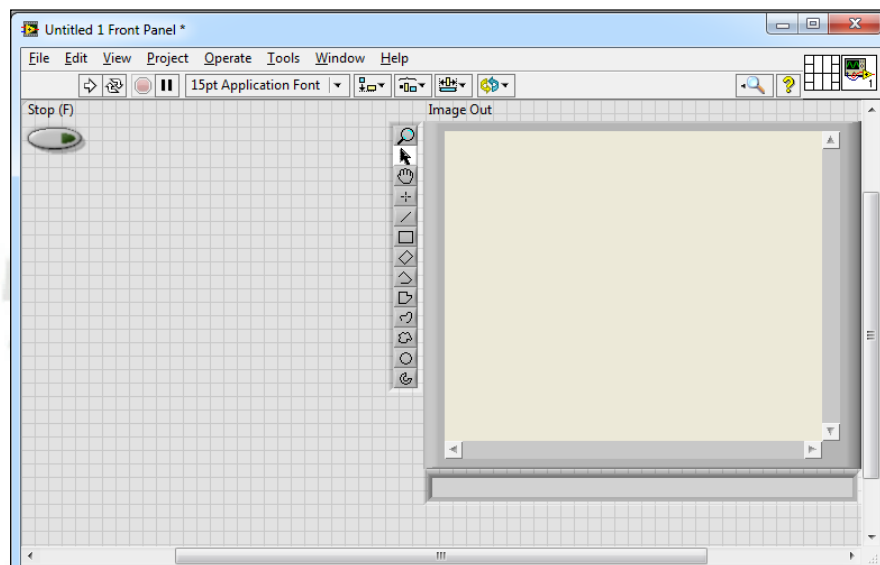


Figura 3. 13: Panel frontal
Fuente: Elaboración propia

Para que funcione la adquisición de imágenes continuas, se tiene que agregar el `imaq create` a la ventana de diagrama de bloques, en el cual se tiene que definir el nombre de la imagen y especificar si la entrada de imágenes van hacer en escala de gris, a colores, etc.

Seleccionamos Grayscale U8 (escala de grises de 8 bits)

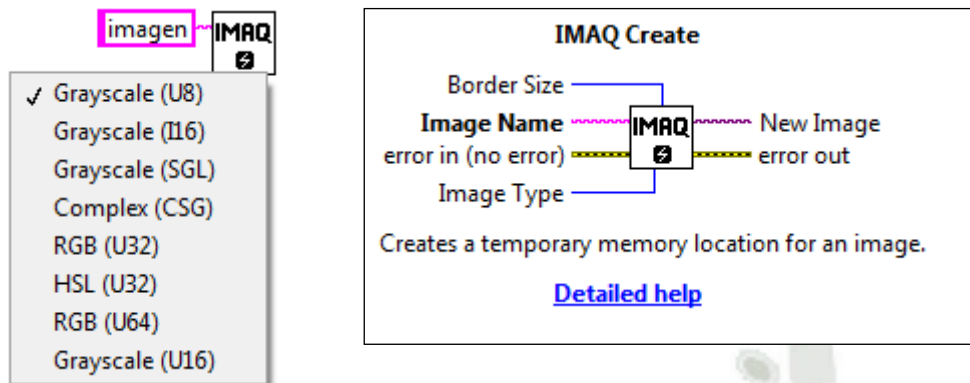


Figura 3. 14: Imaq Create
Fuente: Software Labview 2011

SEGUNDO PASO

Extract Single Color Plane: te permite extraer un plano simple de la imagen adquirida, estos planos pueden ser rojo, verde, azul, intensidad, luminancia, etc.

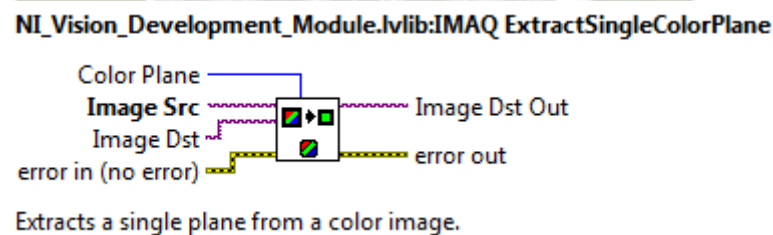


Figura 3. 15: Imaq Extract Single Color Plane
Fuente: Software Labview 2011

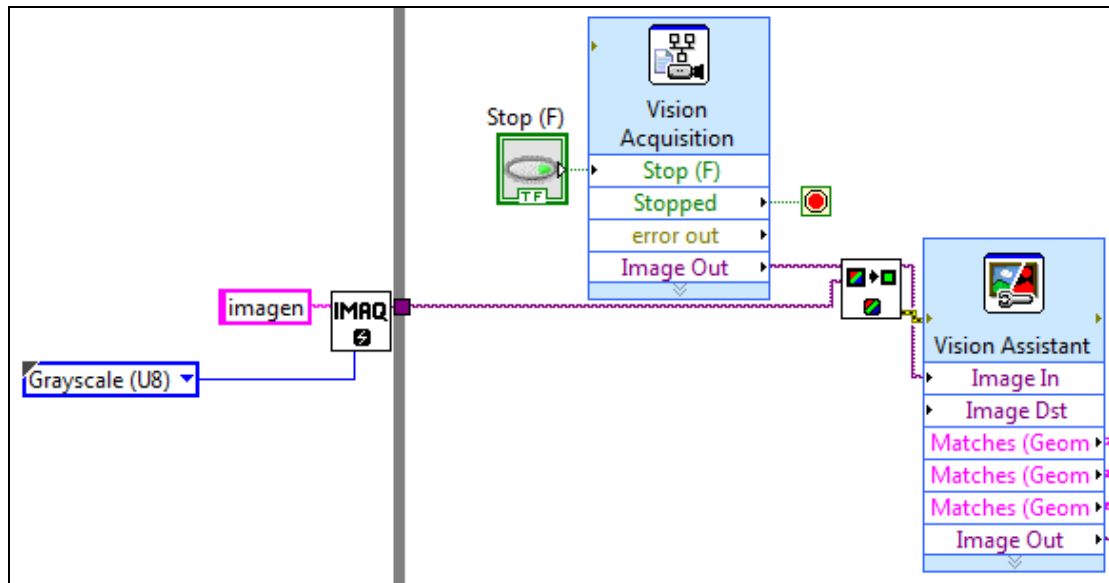


Figura 3. 16: Conexión de imaQ create, visión acquisition y Extract Single Color Plane.
Fuente: Elaboración propia

3.2. DISEÑO DEL PROGRAMA DE CLASIFICACION

3.2.1. Clasificación por Figuras Geométricas

TERCER PASO

Visión assistant

Para seleccionar el asistente de visión, hacemos click derecho en el diagrama de bloques=> visión and motion=> visión expres => visión assistant.

Para la detección de figuras geométricas

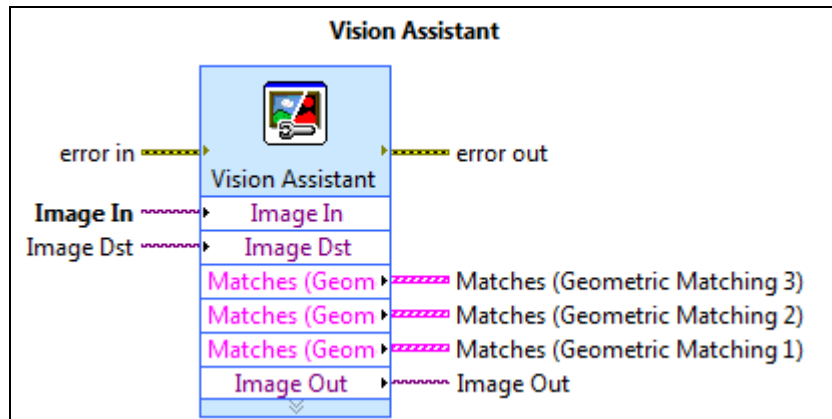


Figura 3. 17: Vision Assistant
Fuente: Elaboración propia

Una vez generado visión assistant, visualizaremos la ventana del asistente de visión, en la cual permite realizar tareas de procesamiento de imágenes, en este caso, seleccionaremos machine visión.

Machine vision: Permite realizar tareas de inspección de visión de máquina comunes, incluyendo la presencia o la ausencia de partes en una imagen, medir dimensiones, localización de bordes, etc.

Geometric Matching: Detecta la presencia de patrones al interior de una imagen o en la región de interés en base a su geometría.

Esta opción solo trabaja con imágenes de 8 Bits, se puede observar 3 tipos de geometric machine, una pertenece a la detección del cuadrado, otra para la detección del triángulo y por ultimo para la detección del círculo, cada una de estas opciones trabajan por separado pero con un mismo fin.

Esta función es utilizada para localizar diferentes patrones en imágenes de escala de grises, puede localizar patrones predeterminados, así exista ruido, rotaciones, etc.

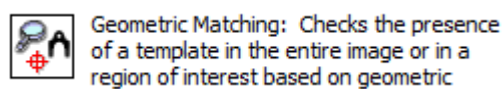


Figura: geometric matching
Fuente: software Labview 2011

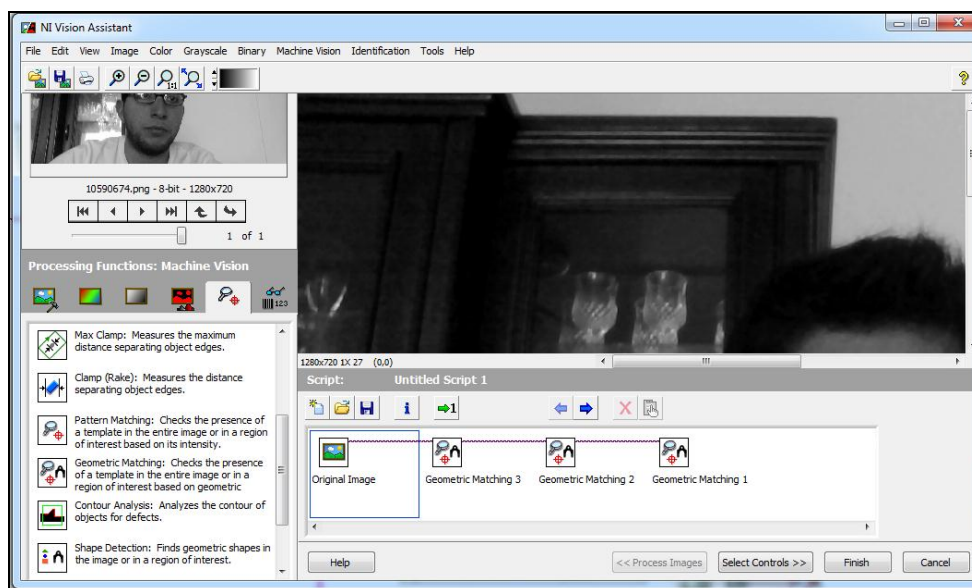


Figura 3. 18: Entorno de vision assistant
Fuente: Elaboración propia

3.2.1.1. Detección de la figura geométrica (cuadrado)

Para realizar este proceso se tiene el patrón que se va a localizar, que es el cuadrado. Hacer click new template, seleccionar el área que se va a utilizar como patrón.



Figura 3. 19: Template
Fuente: Elaboración propia

Definir curvas, se coloreara las áreas que no son de nuestro interés, todo lo que se encuentra alrededor del cuadrado, se colorea, puedes modificar el ancho del lápiz en pen width para ayudar al proceso, por defecto ya reconoce con color verde el contorno del cuadrado.

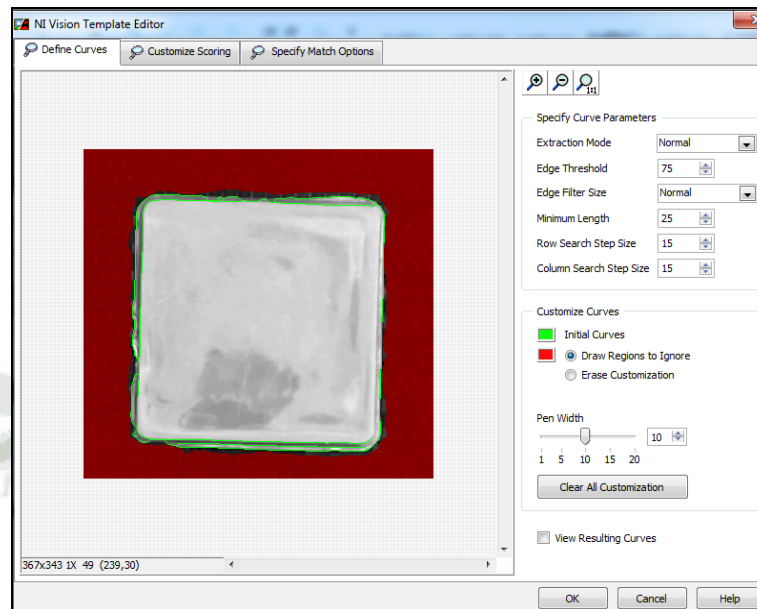


Figura 3. 20: Define Curves
Fuente: Elaboración propia

Personalizar puntuación, colorear el área que no son de interés una vez más para certificar lo anteriormente realizado, variar el ancho del lápiz para un mejor coloreo.

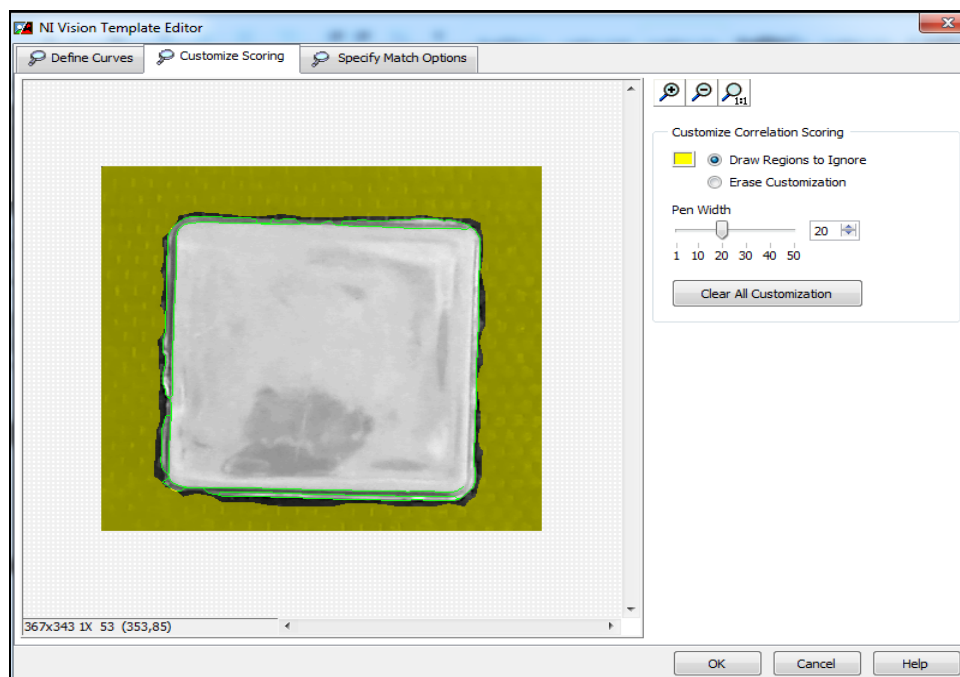


Figura 3. 21: Customize Scoring
Fuente: Elaboración propia

Especificar coordenadas, especifica las coordenadas, planos donde se va a realizar la captación del patrón.

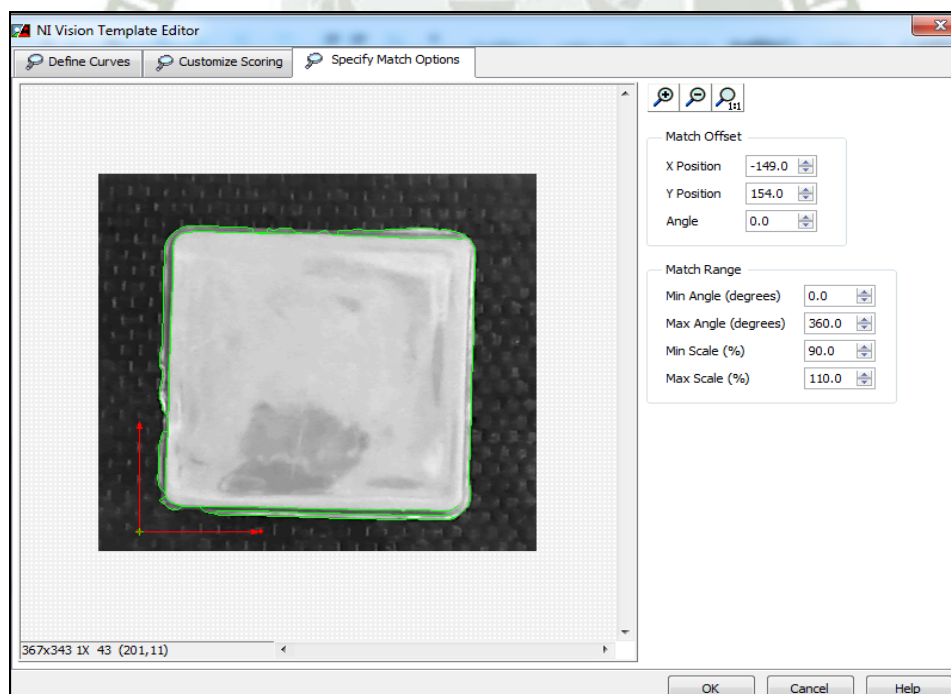


Figura 3. 22: Specify Match Options
Fuente: Elaboración propia

3.2.1.2. Detección de la figura geométrica (círculo)

Para realizar este proceso se tiene el patrón que se va a localizar, que es el círculo. Hacer click new template, seleccionar el área que se va a utilizar como patrón. Figura 3.23 template Circulo

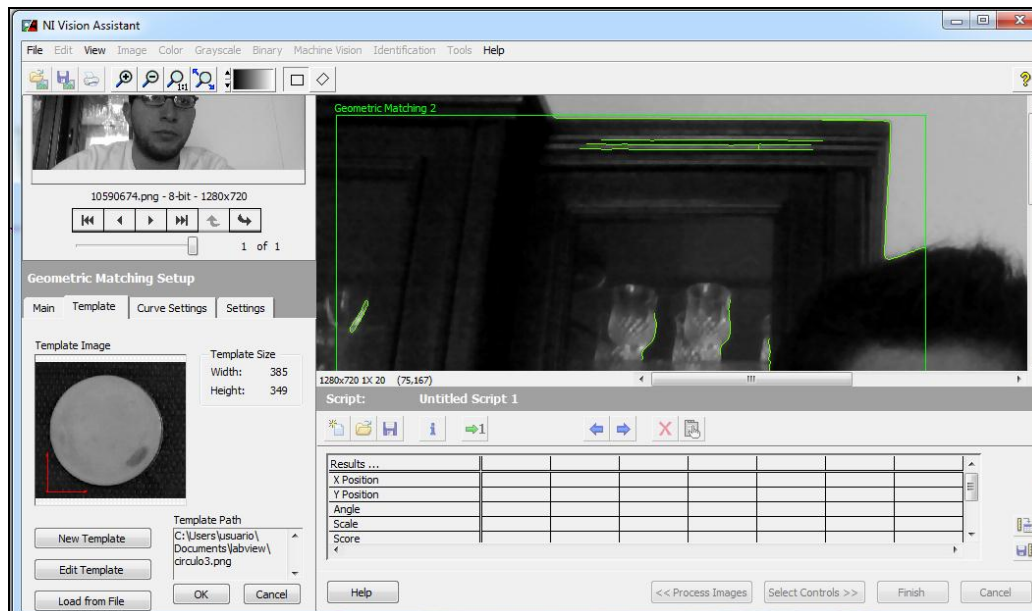


Figura 3. 23: Template Circulo
Fuente: Elaboración propia

Definir curvas, se coloreara las áreas que no son de nuestro interés, todo lo que se encuentra alrededor del cuadrado, se colorea con el lápiz, puedes modificar el ancho del lápiz en pen width para ayudar al proceso, por defecto se reconoce con color verde el contorno del círculo.

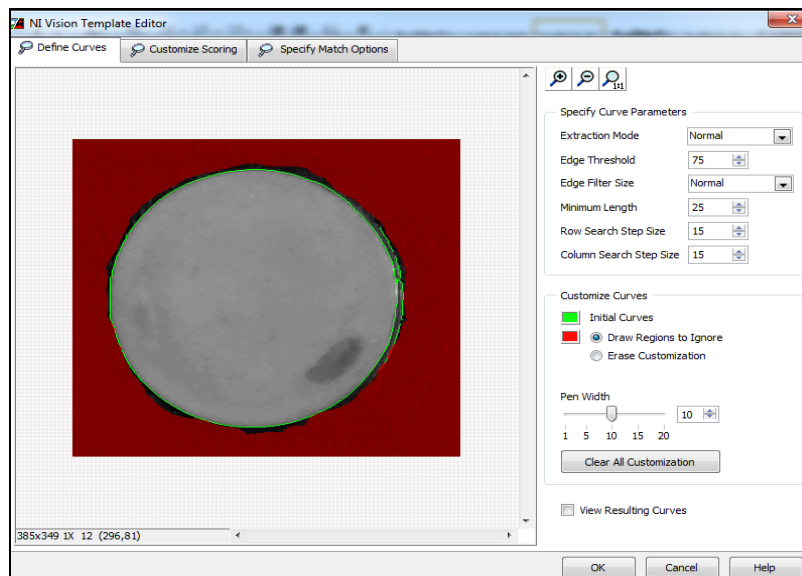


Figura 3. 24: Define Curves
Fuente: Elaboración propia

Personalizar puntuación, colorear el área que no son de interés una vez más para certificar lo anteriormente realizado, ayudarse con el ancho del lápiz al momento del colorear.

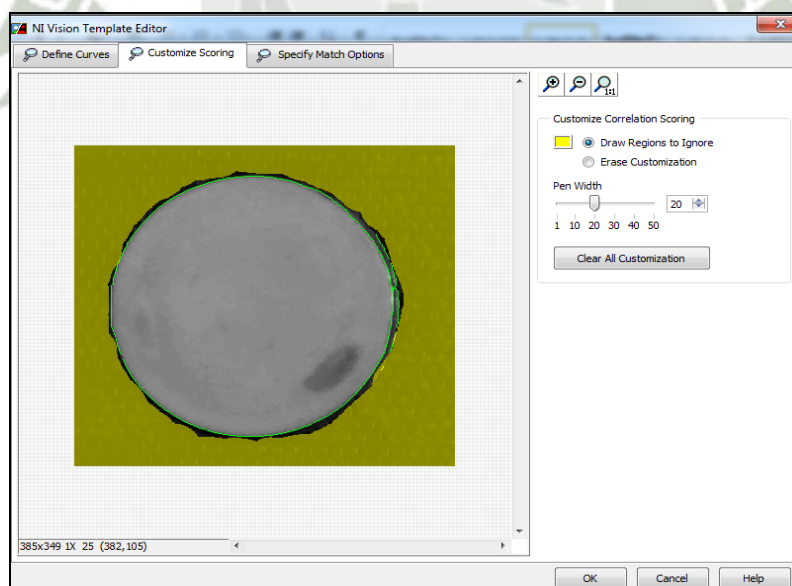


Figura 3. 25: Customize Scoring
Fuente: Elaboración propia

Especificar coordenadas, especifica las coordenadas, planos donde se va a realizar la captación del patrón. Figura 3.26 Specify Match options

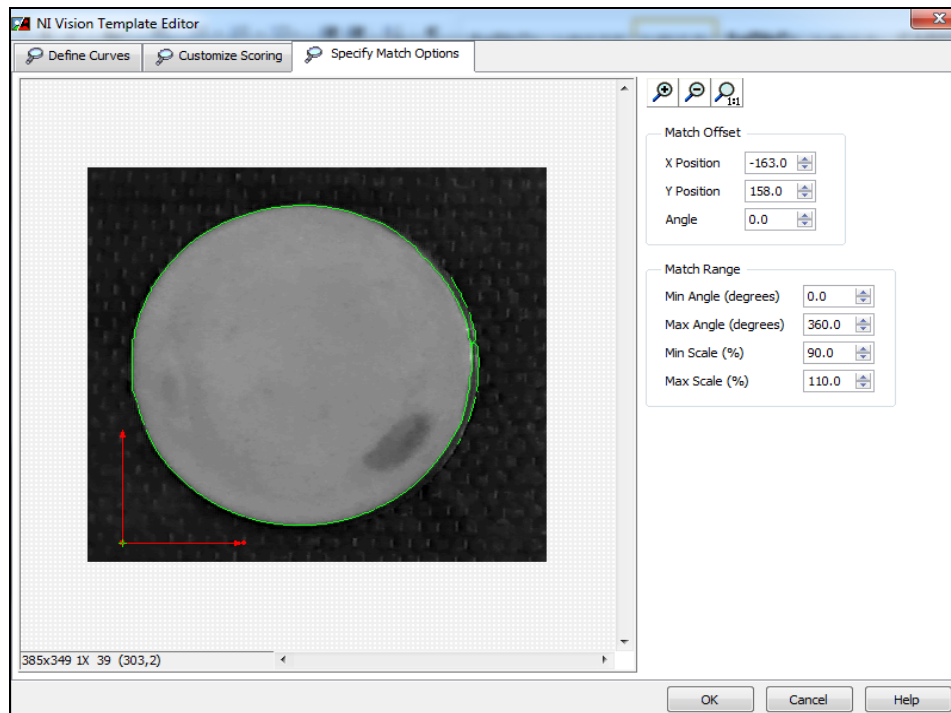


Figura 3. 26: Specify Match options
Fuente: Elaboración propia

3.2.1.3. Detección de la figura geométrica (Triángulo)

Para realizar este proceso se tiene el patrón que se va a localizar, que es el triángulo. Hacer click new template, seleccionar el área que se va a utilizar como patrón.

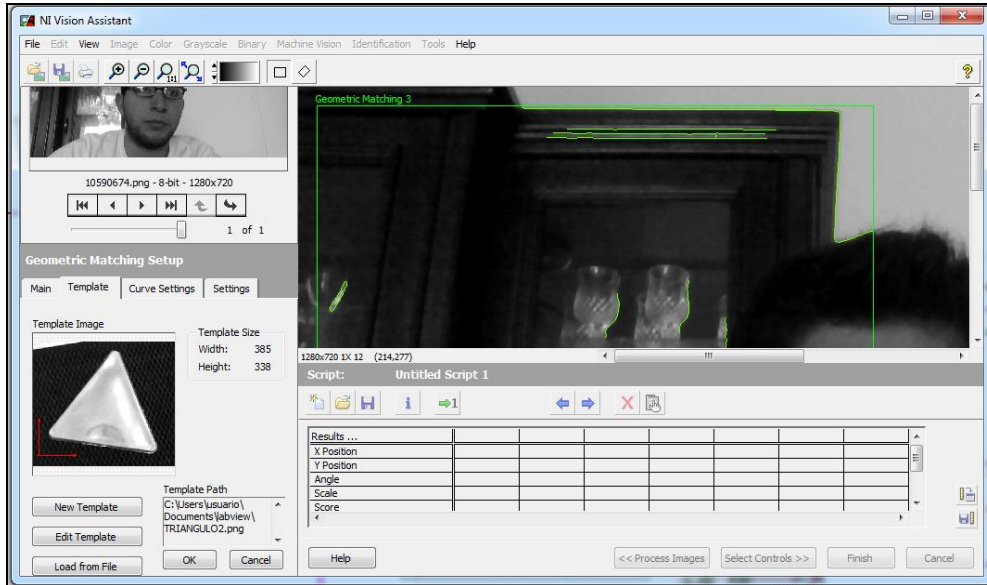


Figura 3. 27: Template tirngulo

Fuente: Elaboración propia

Definir curvas, se coloreara las áreas que no son de nuestro interés, todo lo que se encuentra alrededor del cuadrado, se colorea con el lápiz, puedes modificar el ancho del lápiz en pen width para ayudar al proceso, por defecto se reconoce con color verde el contorno del triángulo.

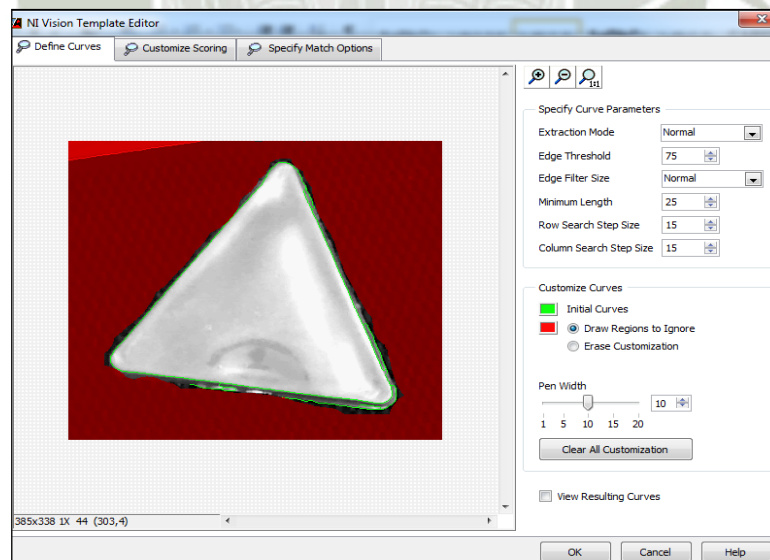


Figura 3. 28: Define Curves

Fuente: Elaboración propia

Personalizar puntuación, colorear el área que no son de interés una vez más para certificar lo anteriormente realizado, ayudarse con el ancho del lápiz al momento del colorear.

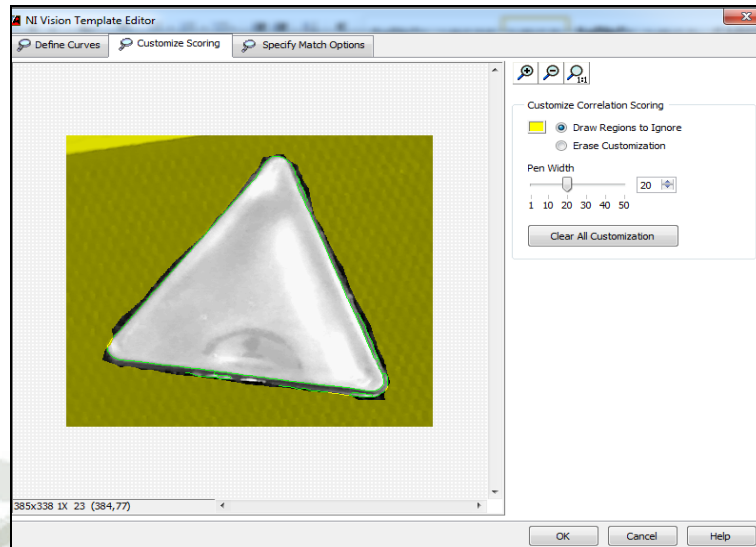


Figura 3. 29: Customize Scoring
Fuente: Elaboración propia

Especificar coordenadas, especifica las coordenadas, planos donde se va a realizar la captación del patrón.

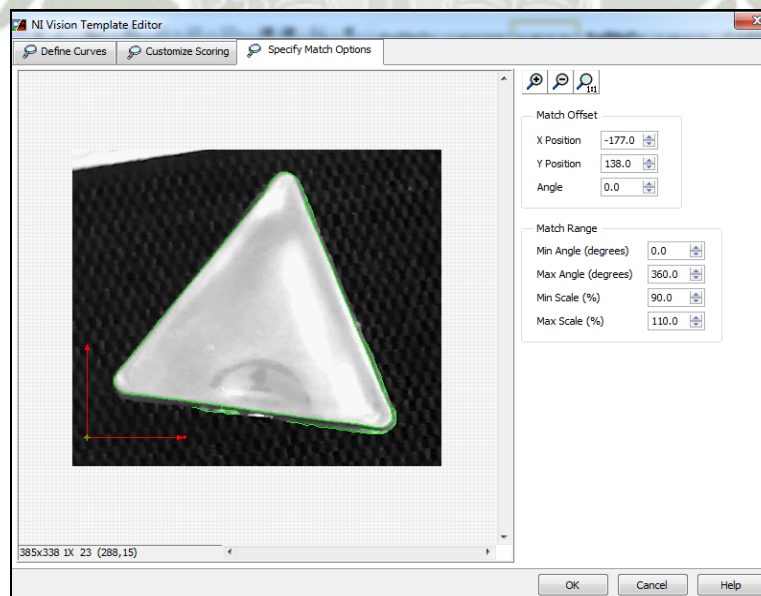


Figura 3. 30: Specify Match options
Fuente: Elaboración propia

Se visualizará estas 3 opciones ya creadas anteriormente, cada uno con las especificación perteneciente al triángulo, cuadrado y círculo, siendo así las figuras geométricas a localizar. Si es que se desea realizar la detección de más figuras geométricas, solo se creará un nuevo patrón y así sucesivamente aumentarán las figuras a detectar.

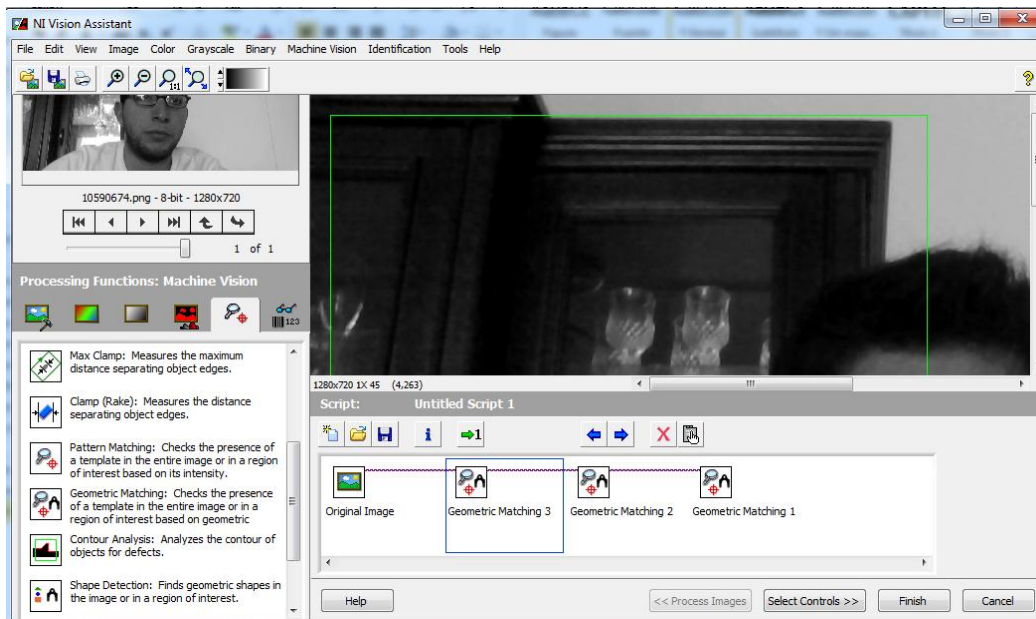


Figura 3. 31: Asistente de visión
Fuente: Elaboración propia

Hacer click en select controls, donde se hará check en Matches de cada figura geométrica, para así visualizar en el diagrama de bloques del asistente de visión la opción de geometric machine perteneciente a cada figura geométrica.

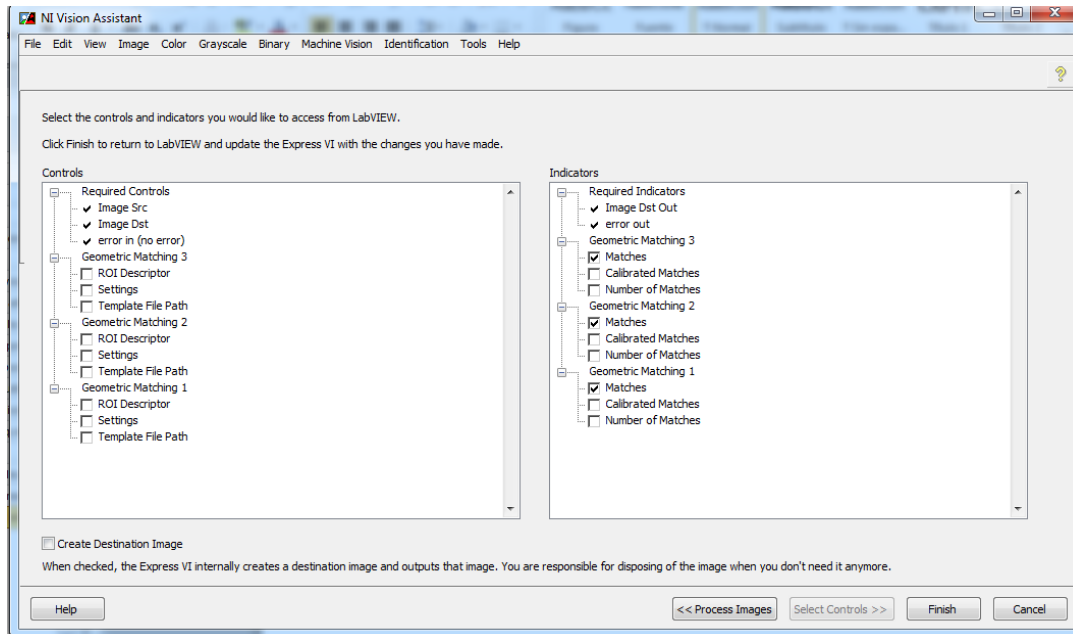


Figura 3. 32: Select Controls
Fuente: Elaboración propia

CUARTO PASO

Array to cluster: Convierte a 1D array a cluster de elementos del mismo tipo como elementos de array.

Conectar la salida del asistente visión (geometric maching) al array to cluster.

Array To Cluster

array  cluster

Figura 3. 33: Array to cluster
Fuente: Software Labview 2011

Unbundle by name: Retorna los elementos del cluster de acuerdo al nombre especificado.

Buscar el icono unbundle by name en la pestaña de funciones del panel de diagrama de bloques, conectar la salida del array to cluster con el unbundle by name, y por defecto va a tomar el nombre del asistente que es matches (geometric matching), hacer click y especificar la opción bounding box para la aparición del cuadrado.

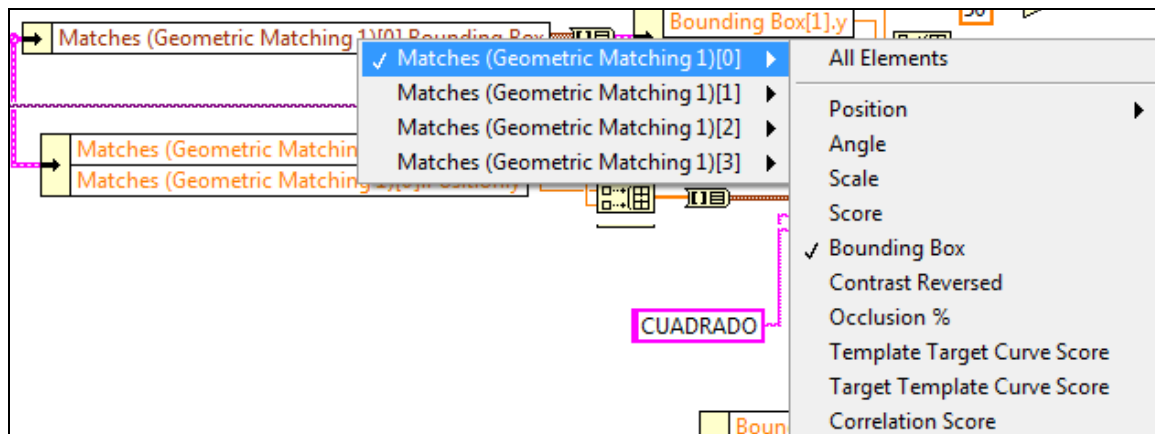


Figura 3. 34: Unbundle by name
Fuente: software Labview 2011

Conectamos la salida del unbudle by name con el array to cluster.

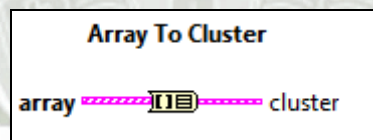


Figura 3. 35: Conexión
Fuente: Software Labview 2011

Seguidamente agregar 4 unbudle by name, esto es para una mayor captación de la cámara, como si fueran planos. Especificar que cada uno se encuentre en la coordenadas X y Y.

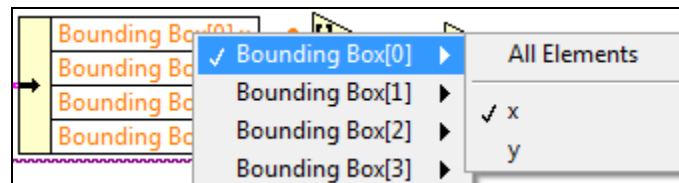
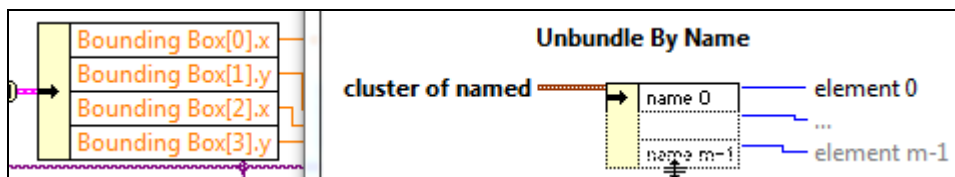


Figura 3. 36: Unbundle by name
Fuente: Software Labview 2011

Round to nearest: Acerca los datos de entrada al valor entero próximo.

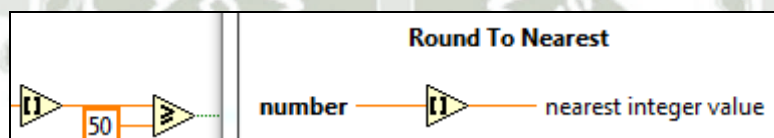


Figura 3. 37: Round to nearest
Fuente: Software Labview 2011

Equal: Realiza una comparación, entre X como valor de entrada y Y como numero por defecto, si esta comparación se cumple envía TRUE (1) como salida y sino no envía FALSE (0), convirtiendo a una salida booleana.

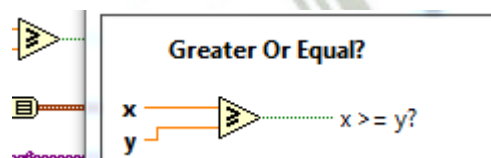


Figura 3. 38: Greater or Equal
Fuente: Software Labview 2011

Build array: Concatena multiples array o elementos a un solo array.

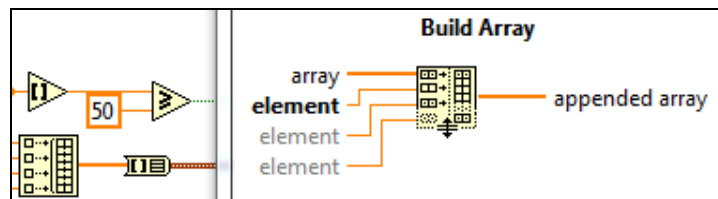


Figura 3. 39: Build Array
Fuente: Software Labview 2011

Imaq overlay rectangle: Superpone un rectángulo en la imagen localizada.

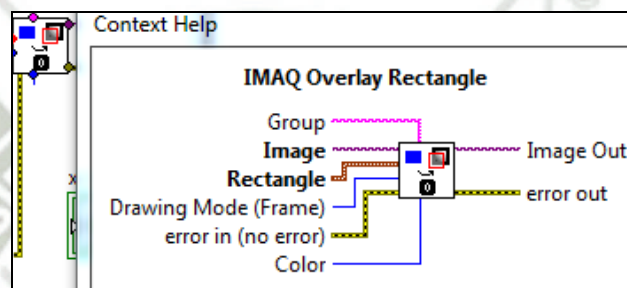


Figura 3. 40: Imaq overlay rectangle:
Fuente: Software Labview 2011

Imaq overlay text: Superpone texto en la imagen.

En este caso para la aparición de texto con el nombre de la figura geométrica captada, se crea unbundle by name pero se escoge position X y position Y, se conecta con build array, este asu vez con array to cluster y para termiar con la entrada Oringin del Imaq overlay text.

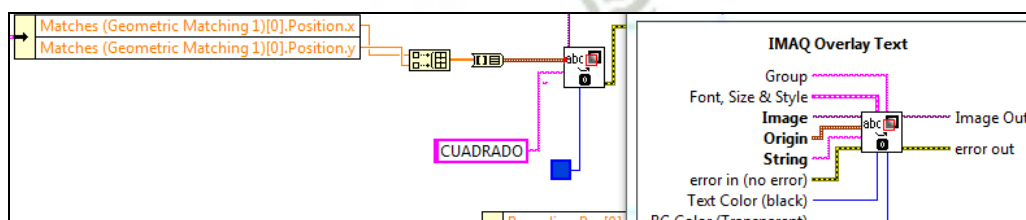


Figura 3. 41: Imaq overlay text:
Fuente: Software Labview 2011

Diagrama de bloques para la conexión del asistente de visión con ima overlay rectangle y el ima overlay text.

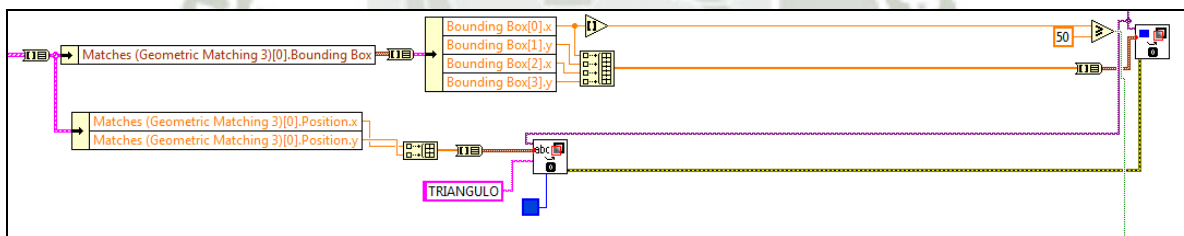
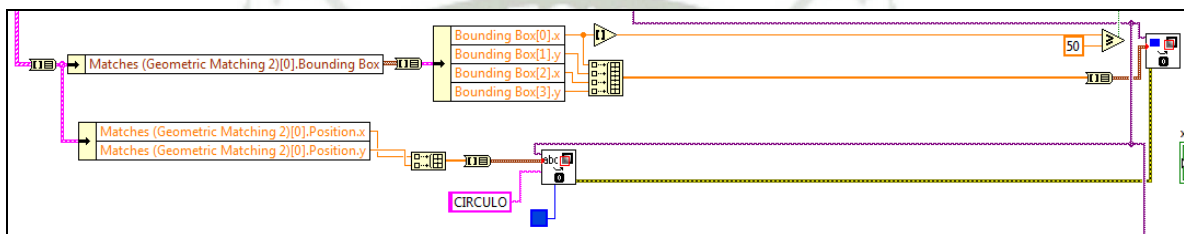
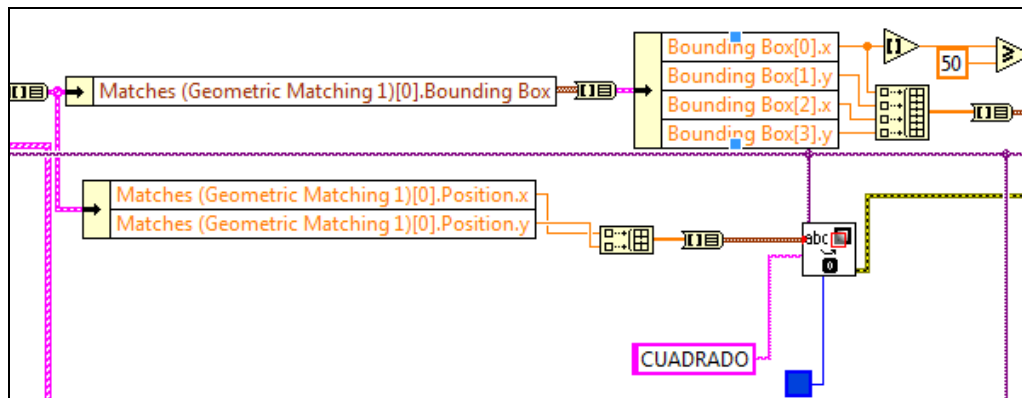


Figura 3. 42: Conexión
Fuente: Elaboración propia

QUINTO PASO

FLIP-FLOP RS TEMPORIZADO

El flip-flop básico, por si sólo es un circuito secuencial asíncrono. Agregando compuertas a las entradas, puede hacerse que el flip-flop responda a los niveles de entrada durante la ocurrencia del pulso de reloj. El flip-flop RS consiste en un flip flop básico NOR y dos

compuertas AND, las salidas de las dos compuertas AND permanecen en cero mientras el pulso de reloj (CP) sea =0, independientemente de los valores de entrada S y R. Cuando el pulso de reloj vaya a 1, la información se permite llegar al flip-flop básico, el estado puesta en uno se logra con S=1, R=0 y CP=1, con S y R en 1, la ocurrencia de los pulsos de reloj causará que ambas salidas vayan momentáneamente a 0. Cuando se quite el pulso, el estado del flip-flop será indeterminado, es decir, podría resultar cualquier estado, dependiendo de sí la entrada de puesta a uno o la de puesta a cero del flip-flop básico, permanezca el tiempo mayor, antes de la transición a 0 al final del pulso.

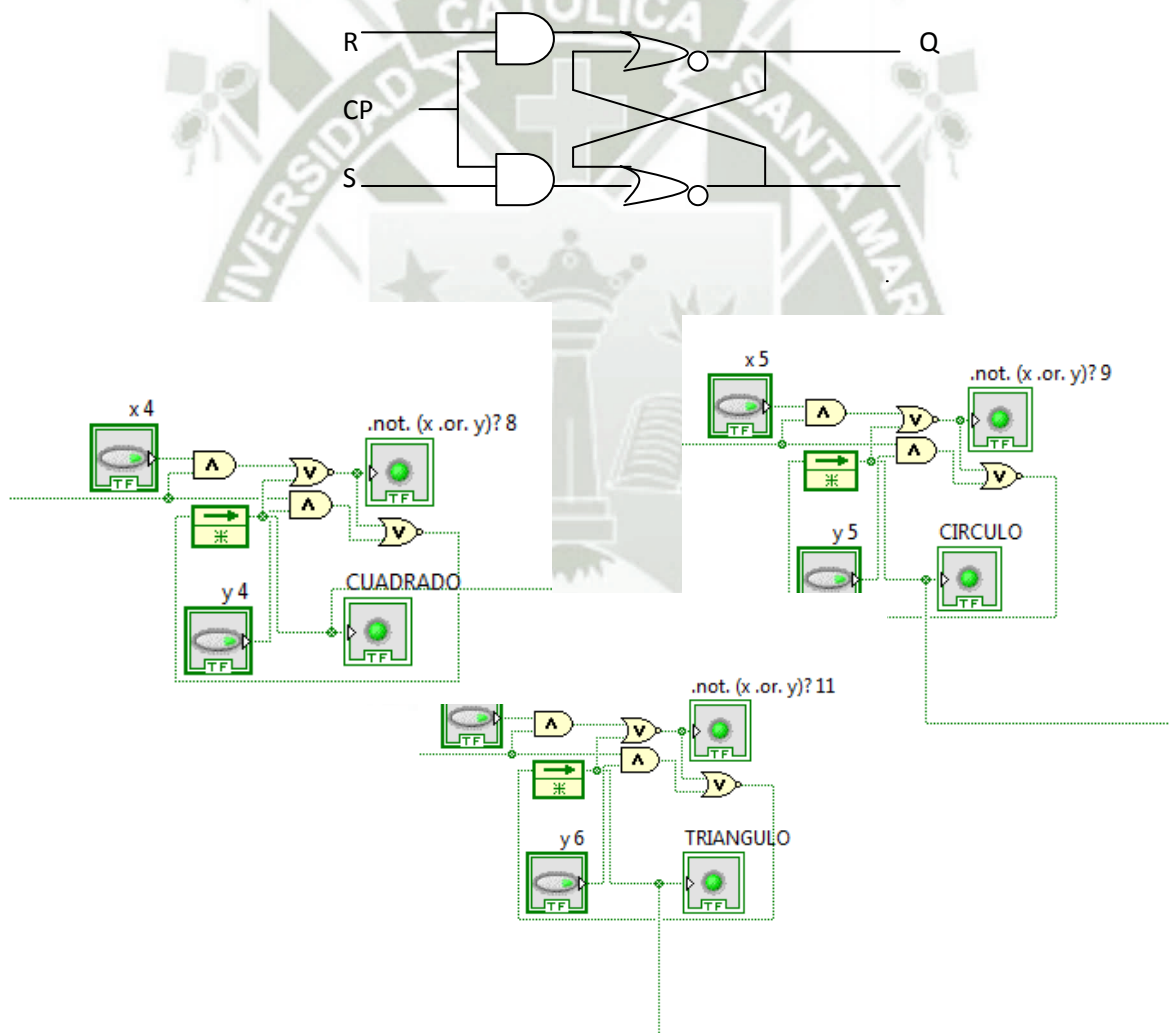


Figura 3. 43: Flip Flop
Fuente: Elaboracion propia

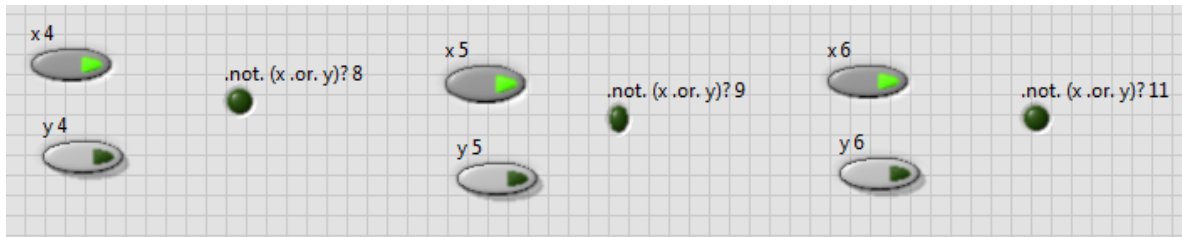


Figura 3. 44: Diagrama de los flip flop en el panel frontal.
Fuente: Elaboracion propia

Se prende los booleanos x4, x5, x6, para que se cumpla la condición requerida de acuerdo a la tabla de verdad del flip flop RS.

SEXTO PASO

3.2.1.4. Control de PWM

Utilizando las librerías de arduino para el control servomotores en labview

Visa resource: Especifica el puerto de conexión con el que se va trabajar.



Figura 3. 45: Visa resource
Fuente: Software Labview 2011

Init: Se utiliza para iniciar la conexión de arduino con la interface de labview.

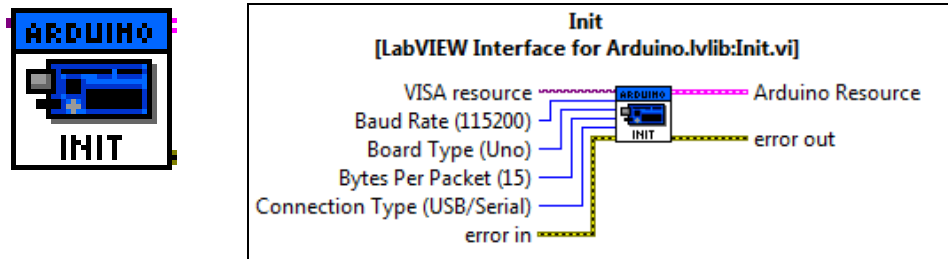


Figura 3. 46: Init
Fuente: Software Labview 2011

Set Number of Servos: Enumera el espacio memoria creada en el arduino para especificar el número de servomotores a utilizar. Se coloca la cantidad de servomotores en el VI.

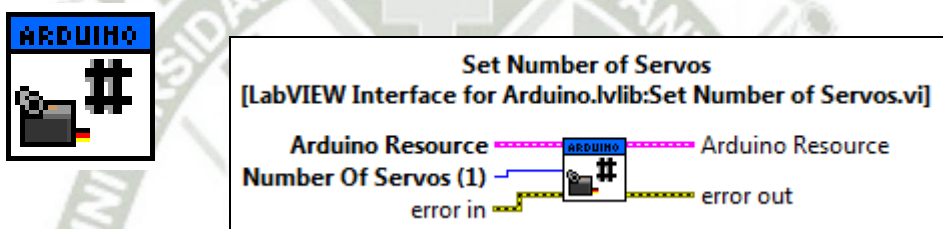


Figura 3. 47: Set Number of Servos
Fuente: Software Labview 2011

Configure Servo: Configura el servo asignando un pin de salida digital, antes de correr el programa se debe especificar el número de servos que se va a utilizar, indicar si es 0, 1, 2, 3, etc

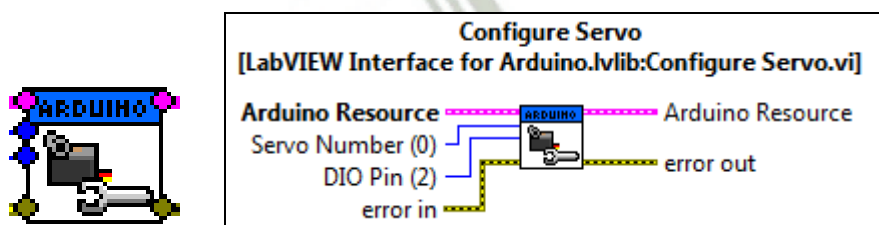


Figura 3. 48: Configure Servo
Fuente: Software Labview 2011

Servo Write Angle: Escribe ángulo que va a girar el servomotor e indicar el número de servo.

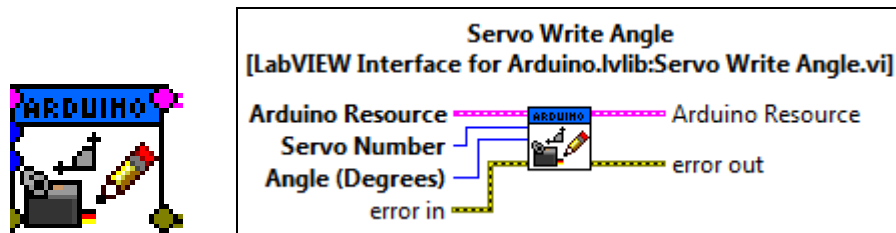


Figura 3. 49: Servo Write Angle
Fuente: Software Labview 2011

Servo read Angle: Lee el último ángulo en grados escrito en el servo correspondiente.

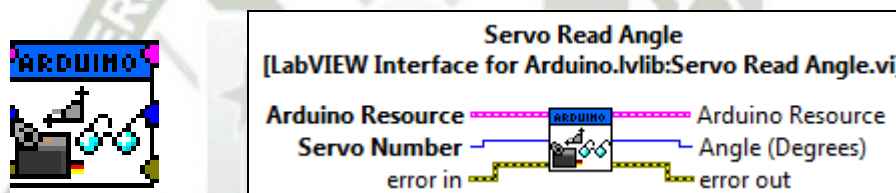


Figura 3. 50: Servo read Angle
Fuente: Software Labview 2011

Close: Cierra la conexión activa del arduino.

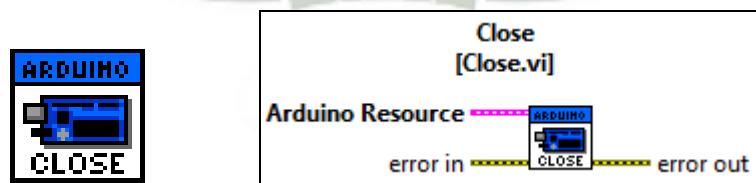


Figura 3. 51: Close
Fuente: Software Labview 2011

Escribir los grados de movimiento del servomotor:

Wait: Espera los segundos indicados en este caso 1000 ms.

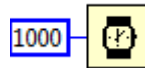


Figura 3. 52: Wait
Fuente: Software Labview 2011

Increment: Incrementa en uno, al valor de entrada.



Figura 3. 53: Increment
Fuente: Software Labview 2011

Equal: Realiza una comparación si el valor de entrada es igual al número fijo.



Figura 3. 54: Equal
Fuente: Software Labview 2011

Select: Si el valor de entrada es F, mandará los grados que se encuentran en la parte inferior y si el valor de entrada es V, mandará los grados de la parte superior.



Figura 3. 55: Select
Fuente: Software Labview 2011

Una vez detectado la figura geométrica, esta envía la señal al ciclo, y comienza a funcionar el bucle de diagrama de bloques. Las iteraciones comienzan a enviar valores y a su vez incrementados en uno, en un tiempo de demora de 1000 ms, la señal llega al selector que es una señal $F = 0$, la cual permite enviar los grados ubicados en

la parte inferior del selector, hasta que pase un tiempo determinado, en este caso es de 30 segundos.

Hace la comparación si $30 = 30$ entonces envía $V=1$ al selector, activando los grados que se encuentran en la parte superior, que es el último movimiento del servomotor 0° grados.

Una vez que tenemos 0° , existe otra condición de comparación, que si es igual 0, se termina el proceso del movimiento del servomotor y a su vez entra a un ciclo que permite apagar el led que corresponde a la figura geométrica detectada, como indicación de termino de proceso.

Diagrama de bloques para escribir los ángulos que se debe mover el servomotor de acuerdo a la figura geométrica captada.

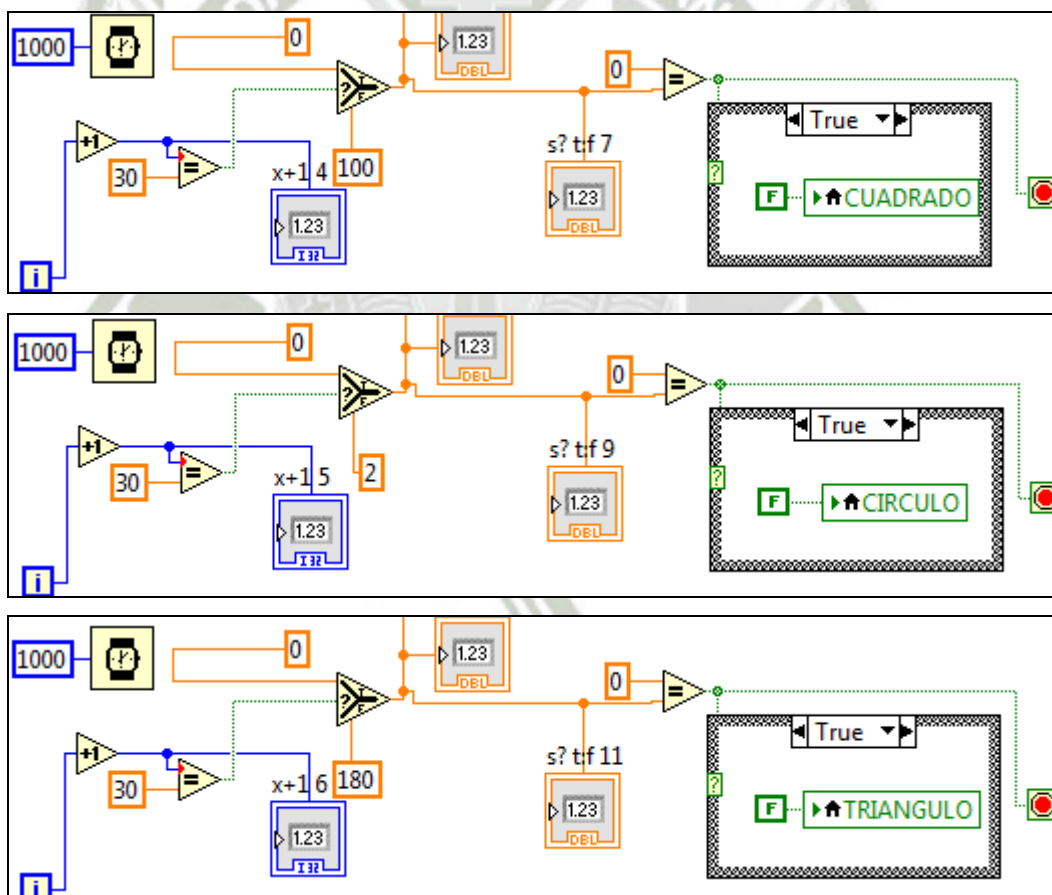


Figura 3. 56: Diagrama de bloques para escribir los ángulos para el servomotor
Fuente: Elaboración propia

Diagrama de bloques para el movimiento del servomotor.

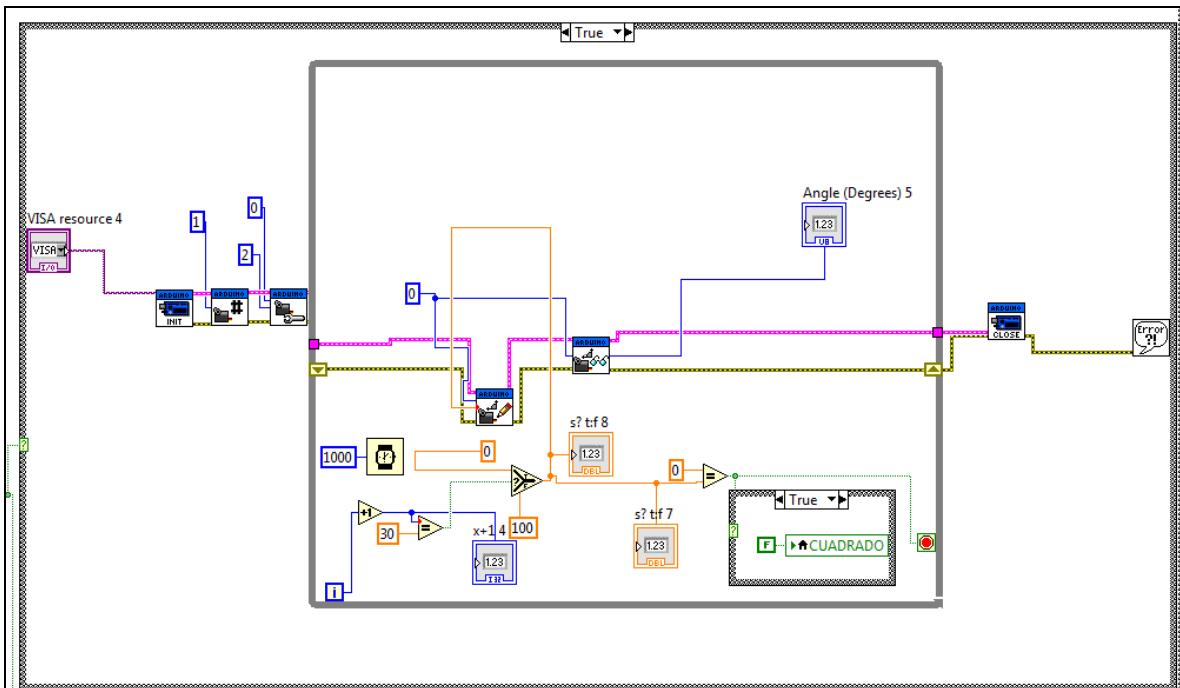


Figura 3. 57: Diagrama de bloques completo
Fuente: Elaboración propia

3.2.2. Clasificación por Colores

PRIMER PASO

En la ventana de diagrama de bloques hacer click derecho y seleccionar del menú de visión express, el icono visión acquisition, el cual nos va a permitir captar imágenes en tiempo real y de forma continua.

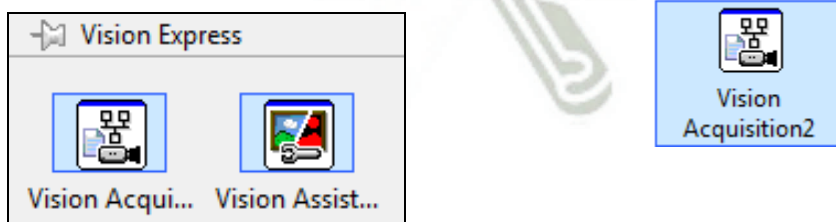


Figura 3. 58: Vision express
Fuente: Software Labview 2011

En este primer paso se identifica que cámara se va utilizar, como podemos observar puede ser la cam0: que es la cámara integrada de la laptop y si utilizamos una web cam con usb, se apreciara un icono debajo de la cam0, que vendría ser la cam1 (por USB).

Se escoge la cámara a utilizar y se presión play para certificar que la cámara funciona, luego next.

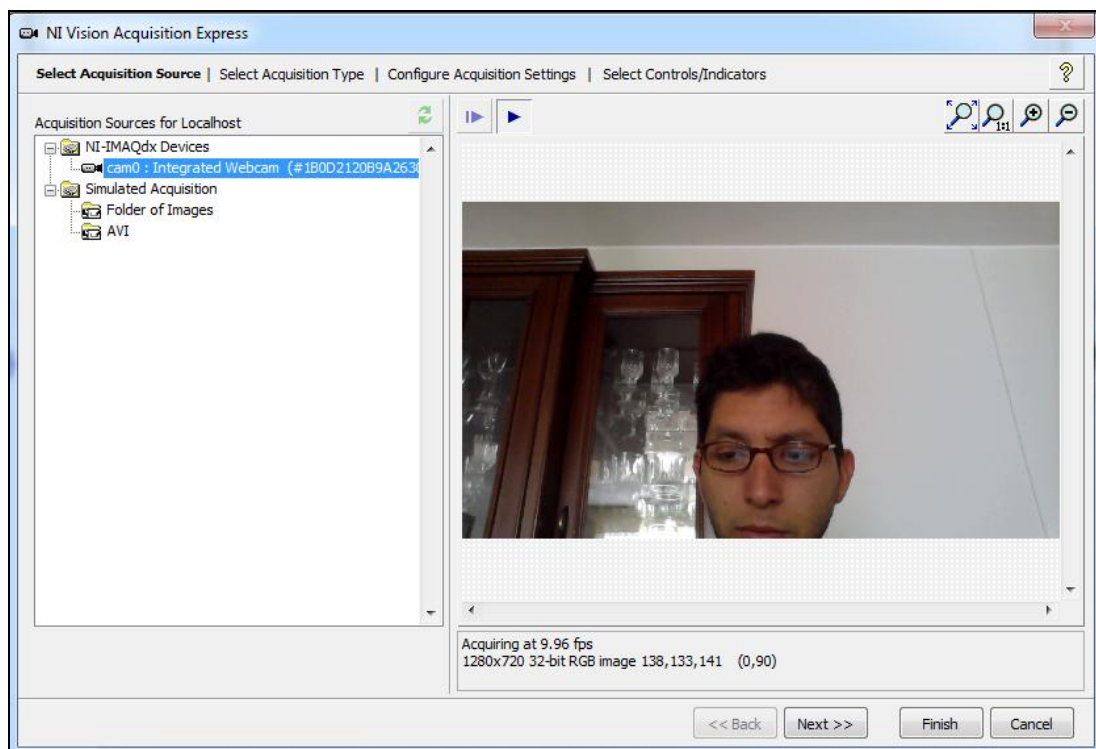


Figura 3. 59: Select acquisition source
Fuente: Elaboración propia

Seleccionar Continuous Acquisition with inline processing (Adquisición continua en proceso), luego next.

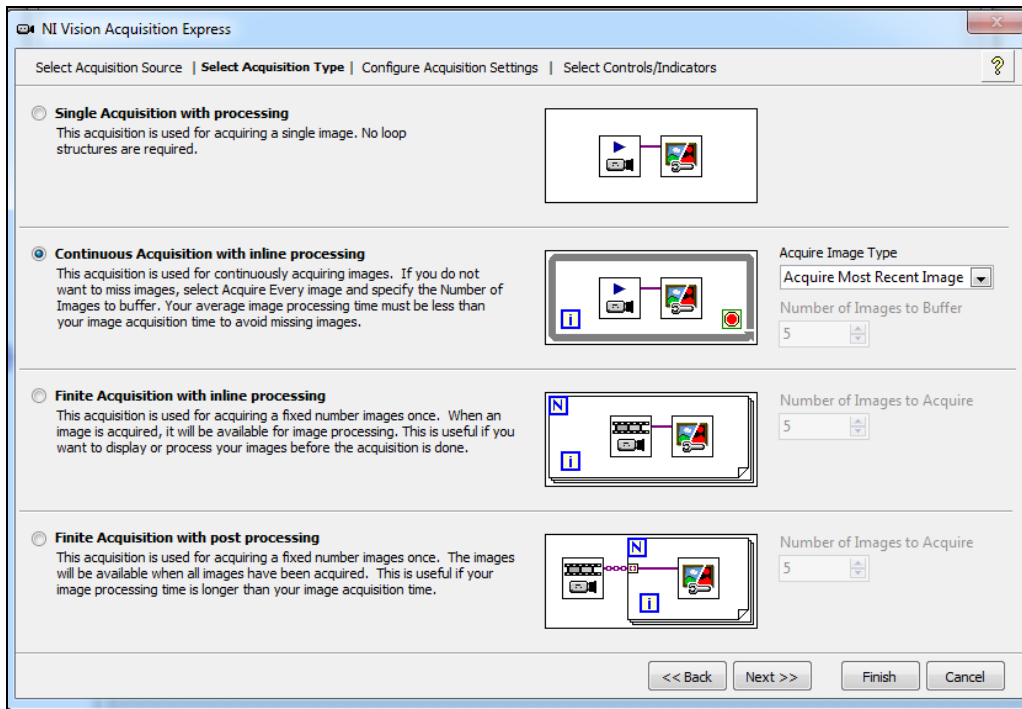


Figura 3. 60: Select acquisition type
Fuente: Elaboracion propia

Se visualiza las opciones que nos brinda la cámara web, presionar next.

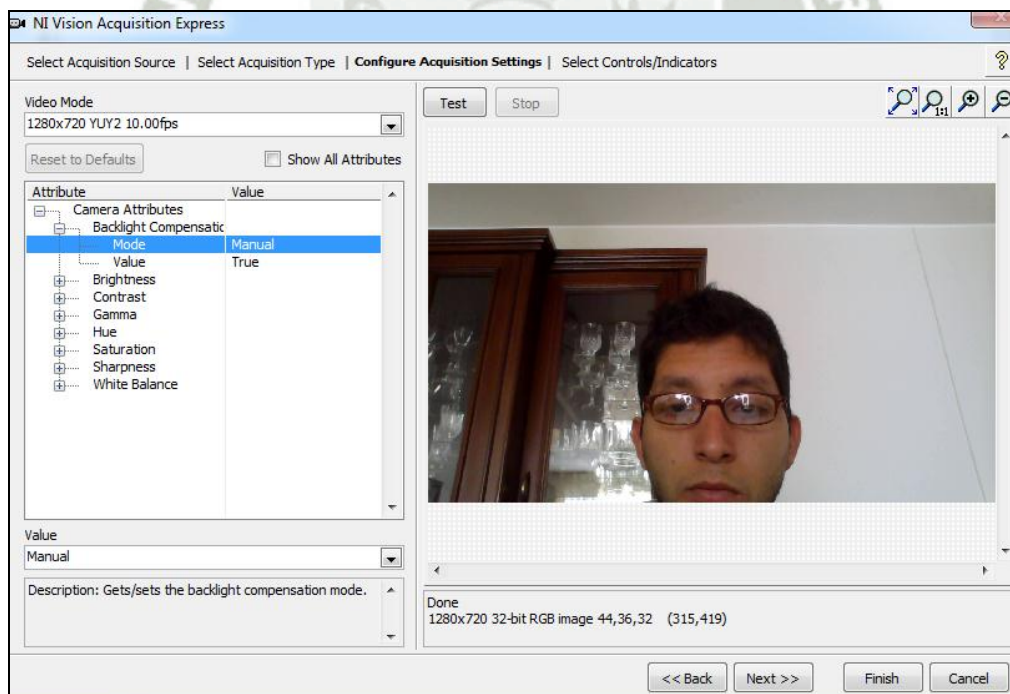


Figura 3. 61: Configure acquisition settings
Fuente: Elaboracion propia

Seleccionar controles e indicadores a aparecer. Por ultimo presionar finish.

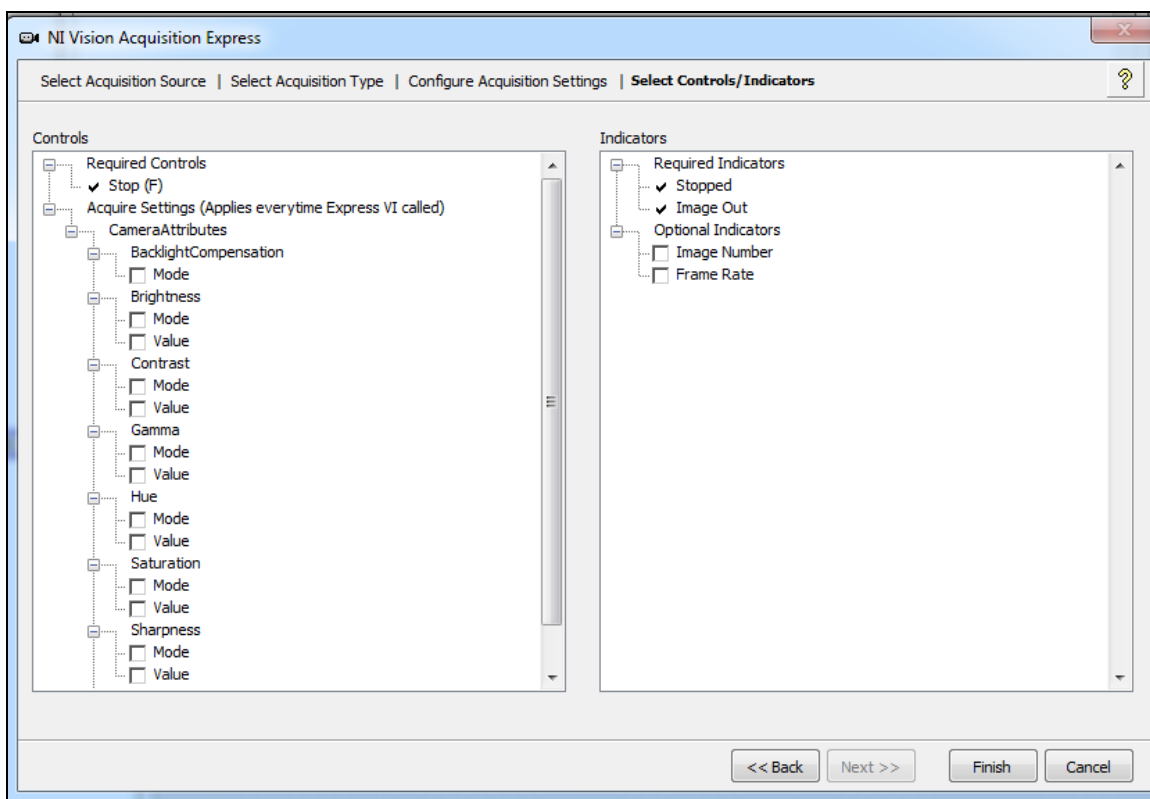


Figura 3. 62: Select controls/indicator
Fuente: Elaboracion propia

Se crea un diagrama de bloques para la adquisición de imágenes, con unos iconos por defecto.

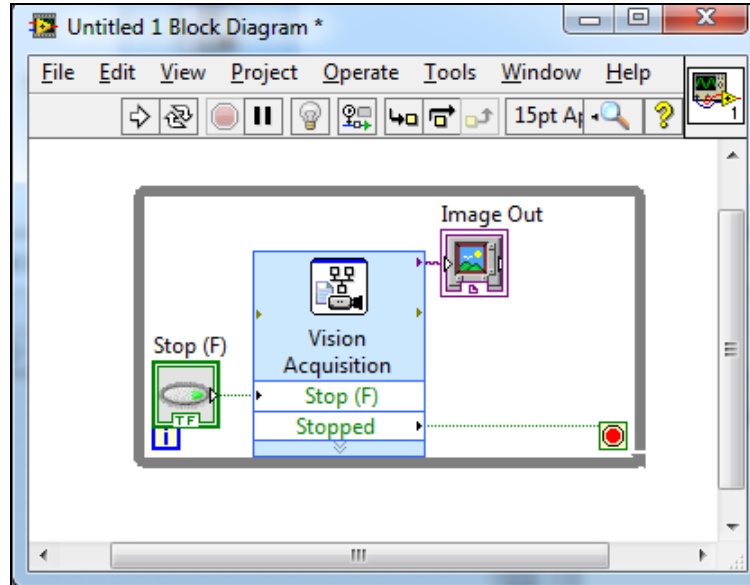


Figura 3. 63: Diagrama de bloques de adquisición de imágenes
Fuente: Elaboración propia

En este diagrama de bloques se observa un stop el cual nos permitirá detener la adquisición de imágenes, image out donde podremos visualizar las imágenes y un ciclo while, el cual realizara acciones repetitivas hasta que presionar stop.

En el panel frontal se crea lo siguiente:

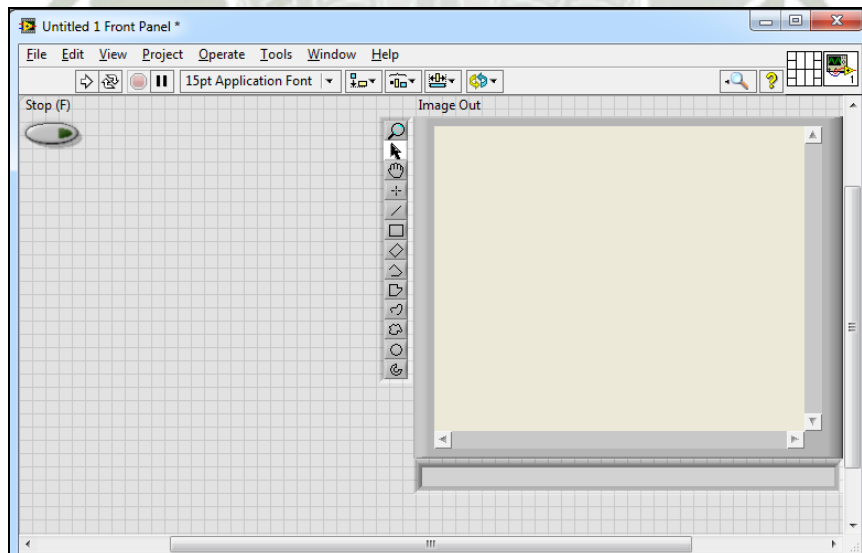


Figura 3. 64: Panel frontal adquisición de imágenes
Fuente: Elaboración propia

Para que funcione la adquisición de imágenes continuas, se tiene que agregar el imaq create a la ventana de diagrama de bloques, en el cual tenemos que definir el nombre de la imagen y especificar si la entrada de imágenes van hacer en escala de gris, a colores, etc.

Seleccionamos RGB (imágenes a colores de 32 bits)

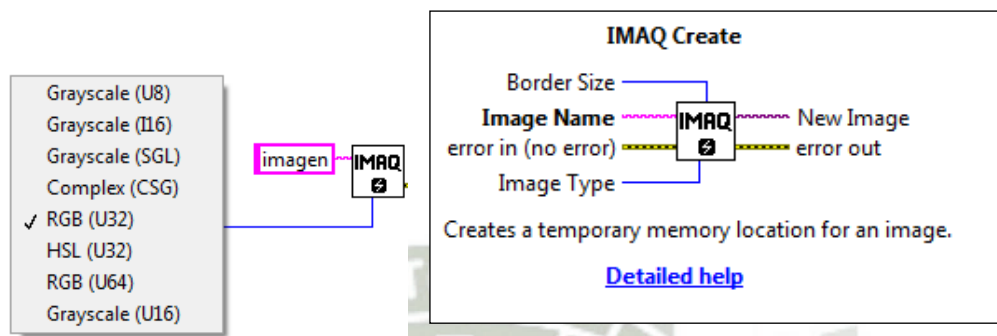


Figura 3. 65: Imaq create
Fuente: Software labview 2011

SEGUNDO PASO

Imaq set color Pixe line 2: Cambia una línea de pixeles de una imagen a color.

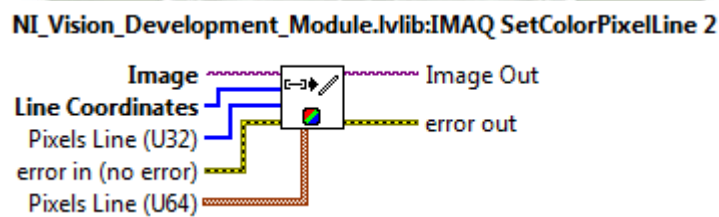


Figura 3. 66: Imaq set color Pixe line 2
Fuente: Software labview 2011

TERCER PASO

Visión assistant

Para seleccionar el asistente de visión, hacemos click derecho en el diagrama de bloques=> visión and motion=> visión expres => visión assistant.

Para la localización de colores.

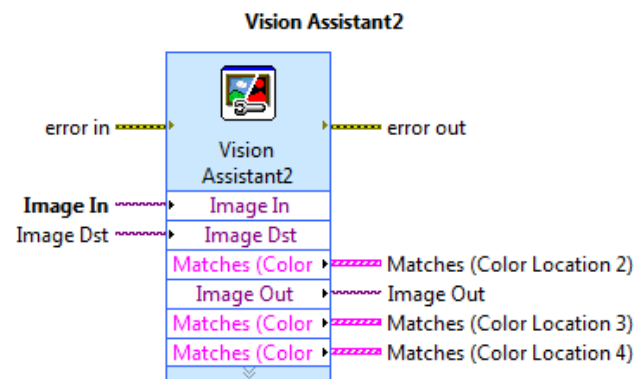


Figura 3. 67: Visión assistant de localización de colores
Fuente: Elaboración propia

Seleccionamos:

Processing functions (color): Esta función nos permite analizar los colores de las diferentes imágenes del proceso, es un procesador de funciones de color.

Color location: Usar esta función para localizar regiones que contenga un específico color, esta función no modifica la imagen, solo la localiza.



Figura 3. 68: Color location
Fuente: Software labview 2011

Se observan tres iconos de color location, estos tres iconos pertenecen a la toma de patrones de los colores, rojo, verde y azul.

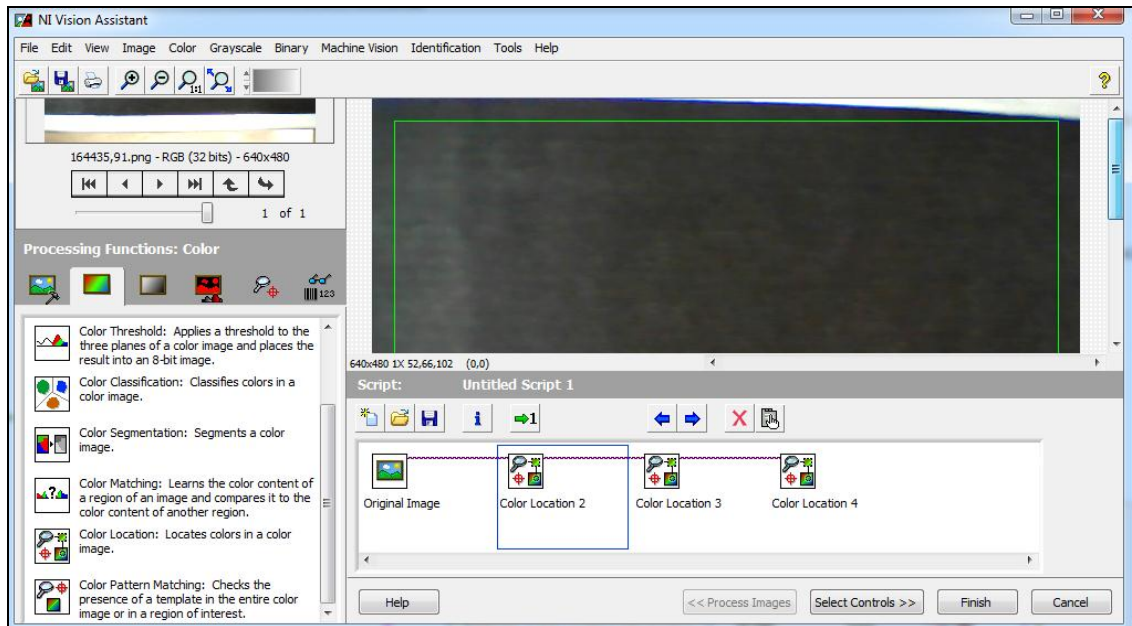


Figura 3. 69: Ni vision assistant para colores
Fuente: Elaboración propia

3.2.2.1. Localización del color rojo

Seleccionar create template (crear patrón), permite seleccionar el patrón que vamos a localizar en una imagen antes ya grabada.

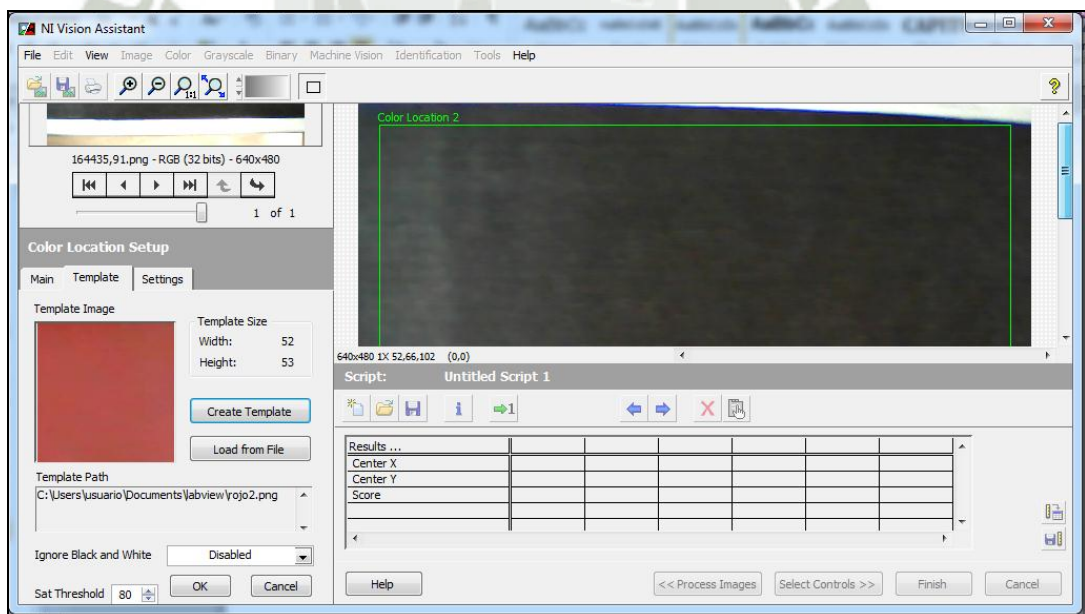


Figura 3. 70: Create template (color rojo)
Fuente: Elaboración propia

Select template in the image: Seleccionar el patrón en la imagen, con el mouse seleccionamos el patrón a localizar, en este caso es el color rojo, localizamos el área, y la encerramos en un cuadro como se ve en la figura, presionar ok.

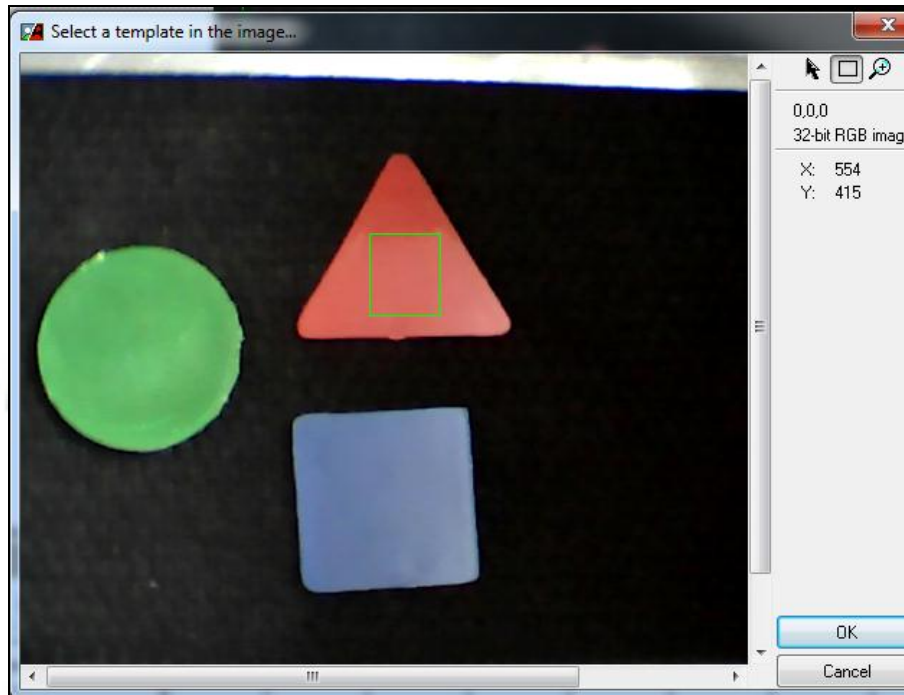


Figura 3. 71: Select template in the image (color rojo)
Fuente: Elaboración propia

Se visualiza el patrón de color rojo y a su vez el patrón es localizado en la imagen, indicando las coordenadas donde se encuentra, presionamos ok.

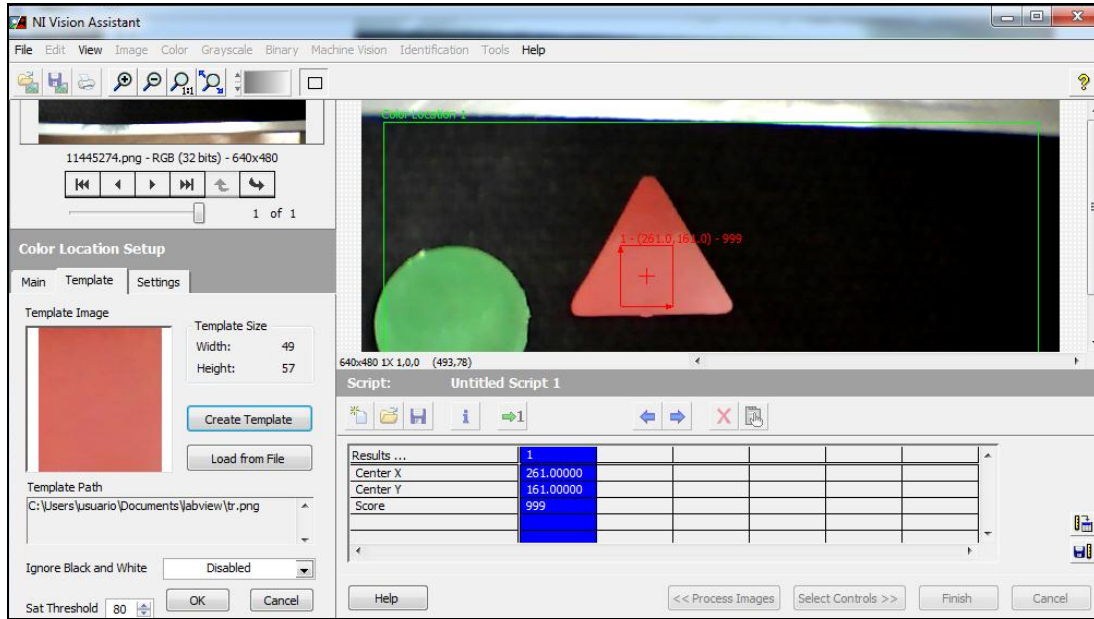


Figura 3. 72: Color detectado (color rojo)
Fuente: Elaboración propia

3.2.2.2. Localización del color verde

Seleccionar créate template (crear patrón), permite seleccionar el patrón que vamos a localizar en una imagen antes ya grabada.

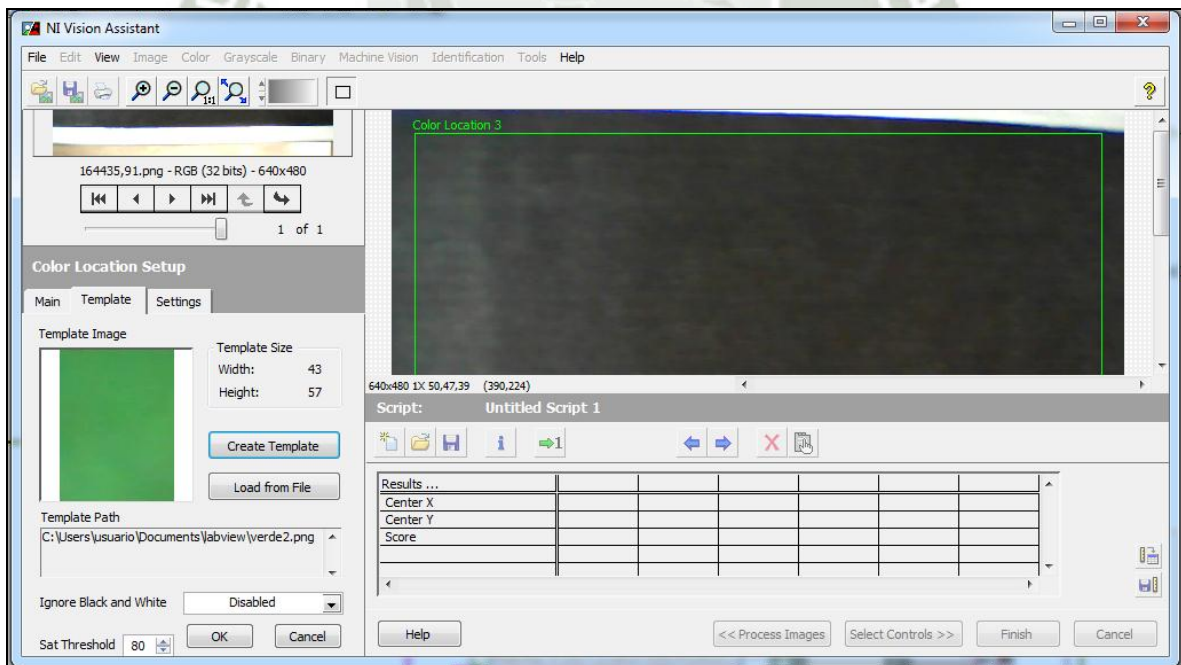


Figura 3. 73: Créate template (color verde)
Fuente: Elaboración propia

Select template in the image: Seleccionar el patrón en la imagen, con el mouse seleccionamos el patrón a localizar, en este caso es el color verde, localizamos el área, y la encerramos en un cuadro como se ve en la figura, presionar ok.

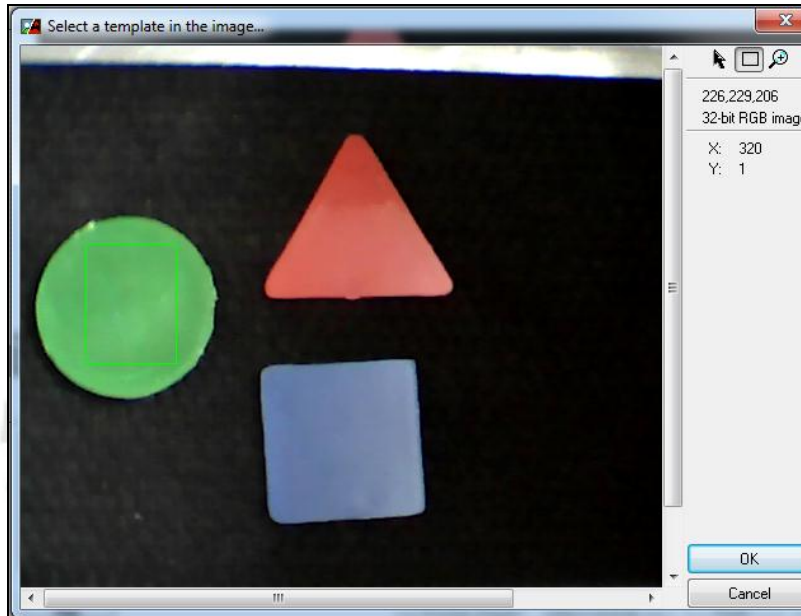


Figura 3. 74: Select template in the image (color verde)
Fuente: Elaboración propia

Se visualiza el patrón de color rojo y a su vez el patrón es localizado en la imagen, indicando las coordenadas donde se encuentra, presionamos ok

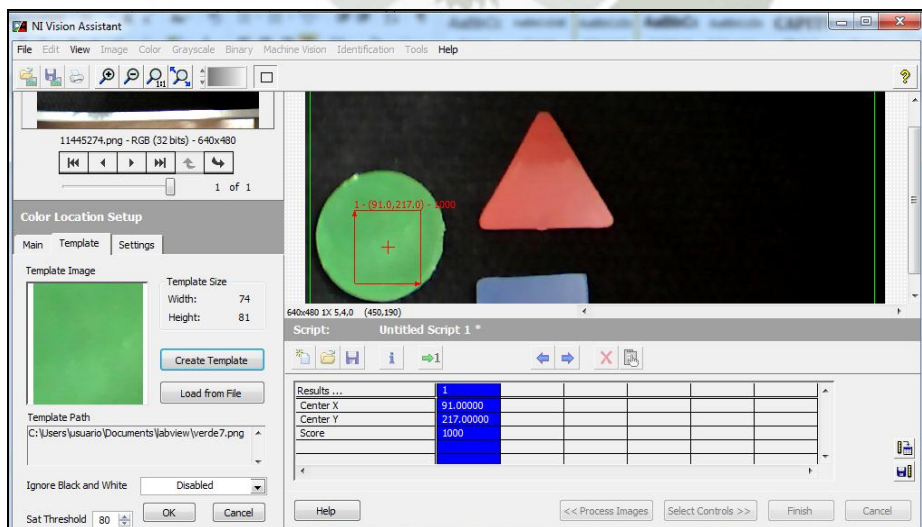


Figura 3. 75: Color detectado (verde)
Fuente: Elaboración propia

3.2.2.3. Localización del color azul

Seleccionar créate template (crear patrón), permite seleccionar el patrón que vamos a localizar en una imagen antes ya grabada.

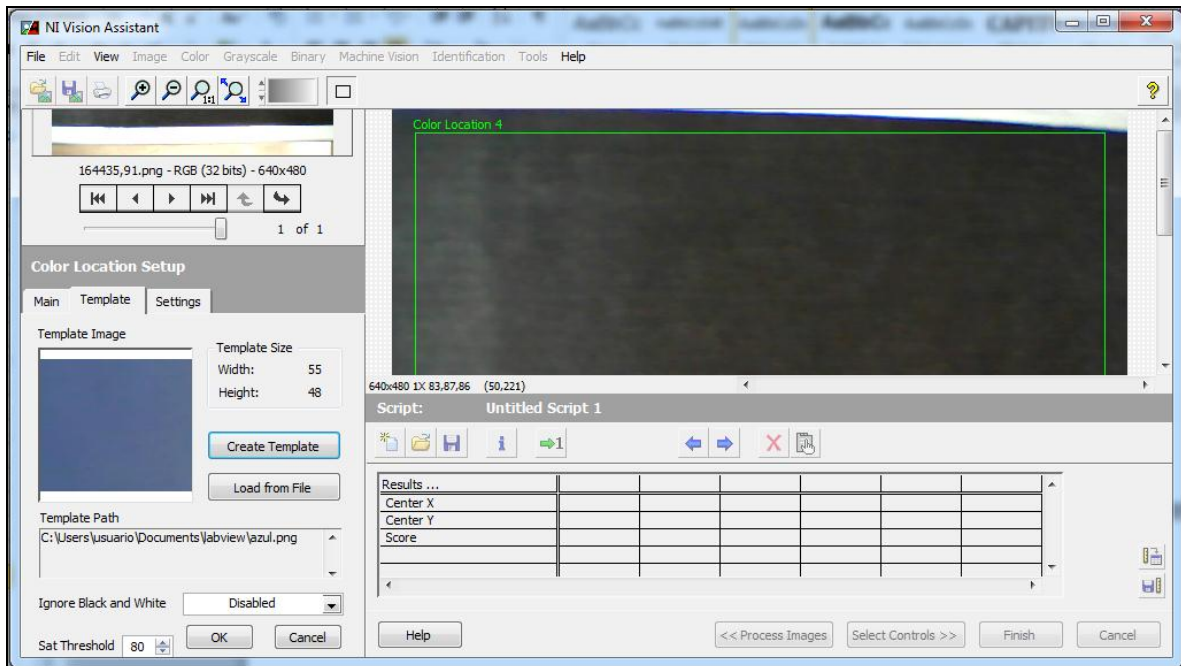


Figura 3. 76: Create template (color azul)
Fuente: Elaboración propia

Select template in the image: Seleccionar el patrón en la imagen, con el mouse seleccionamos el patrón a localizar, en este caso es el color azul, localizamos el área, y la encerramos en un cuadro como se ve en la figura, presionar ok.

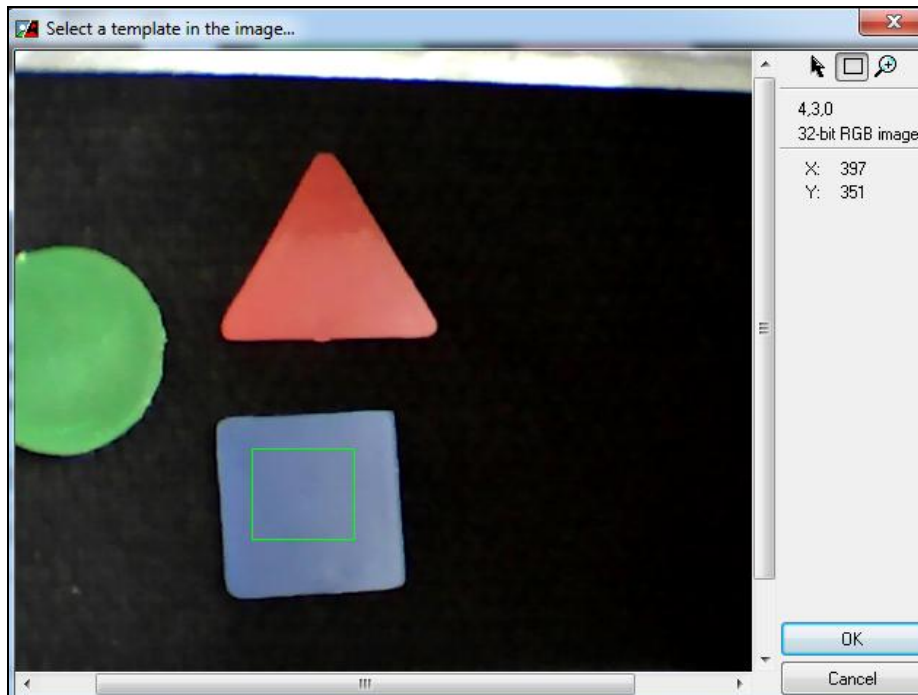


Figura 3. 77: Select template in the image (color azul)
Fuente: Elaboración propia

Se visualiza el patrón de color azul y a su vez el patrón es localizado en la imagen, indicando las coordenadas donde se encuentra, presionamos ok

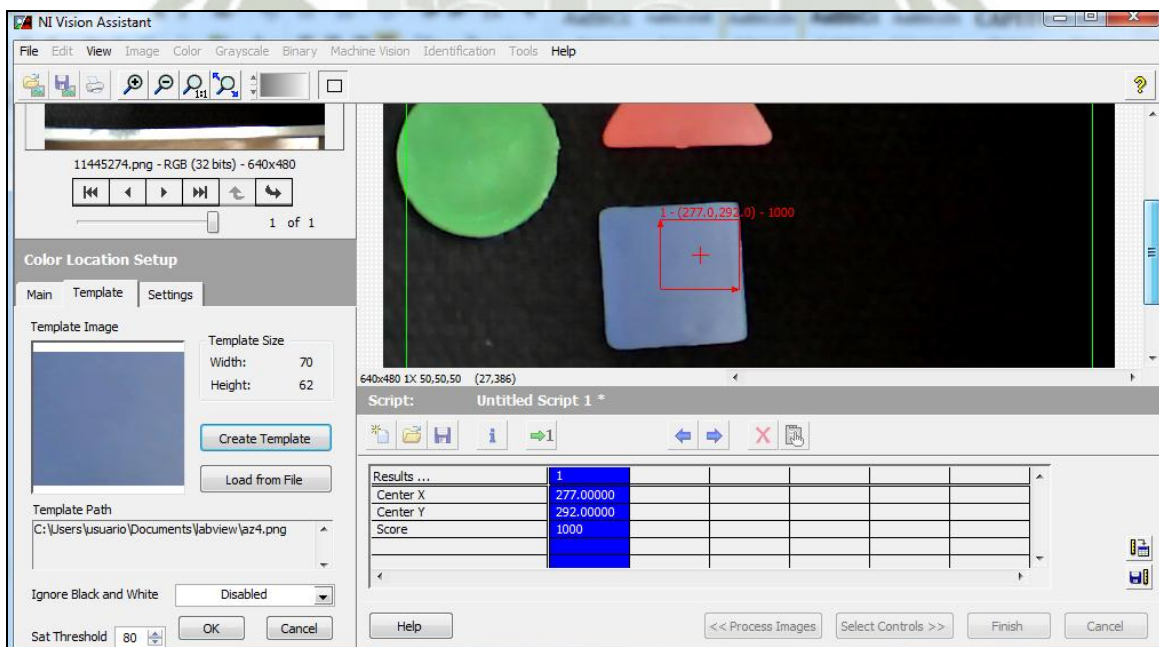


Figura 3. 78: Color detectado (azul)
Fuente: Elaboración propia

CUARTO PASO

Imaq overlay rectangle: Superpone un rectángulo en una imagen.

Conectamos la salida del asistente (matches color location) con el array to cluster, este a su vez es conectado con el unbundle by name y por defecto aparecerá el nombre de la tarea a realizar, se especifica bounding box para la aparición del rectángulo en la figura localizada.

Seguidamente conectamos las salida del unbundle by name con otro array to cluster y a la salida de este es conectado con 4 unbundle by name.

Para juntar las 4 salidas de datos en una, se utiliza el build array, que este es conectado a otro array to cluster.

Para terminar esta salida es conectada con el imaq overlay rectangle para crear el rectángulo en la imagen a localizar.

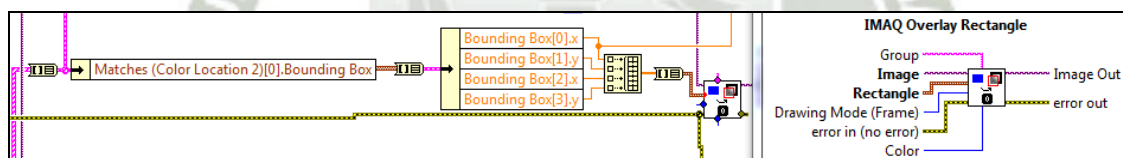


Figura 3. 79: Imaq overlay rectangle
Fuente: Elaboración propia

Imaq overlay text: Superpone texto en la imagen.

En este caso para la aparición de texto con el nombre de la figura geométrica captada, se crea unbundle by name pero se escoge position X y position Y, se conecta con build array este a su vez con array to cluster y para terminar con la entrada Oringin del Imaq overlay text.

Diagrama de bloques para la conexión del asistente de visión con el imaq overlay rectangle y el imaq overlay text en la identificación de colores.

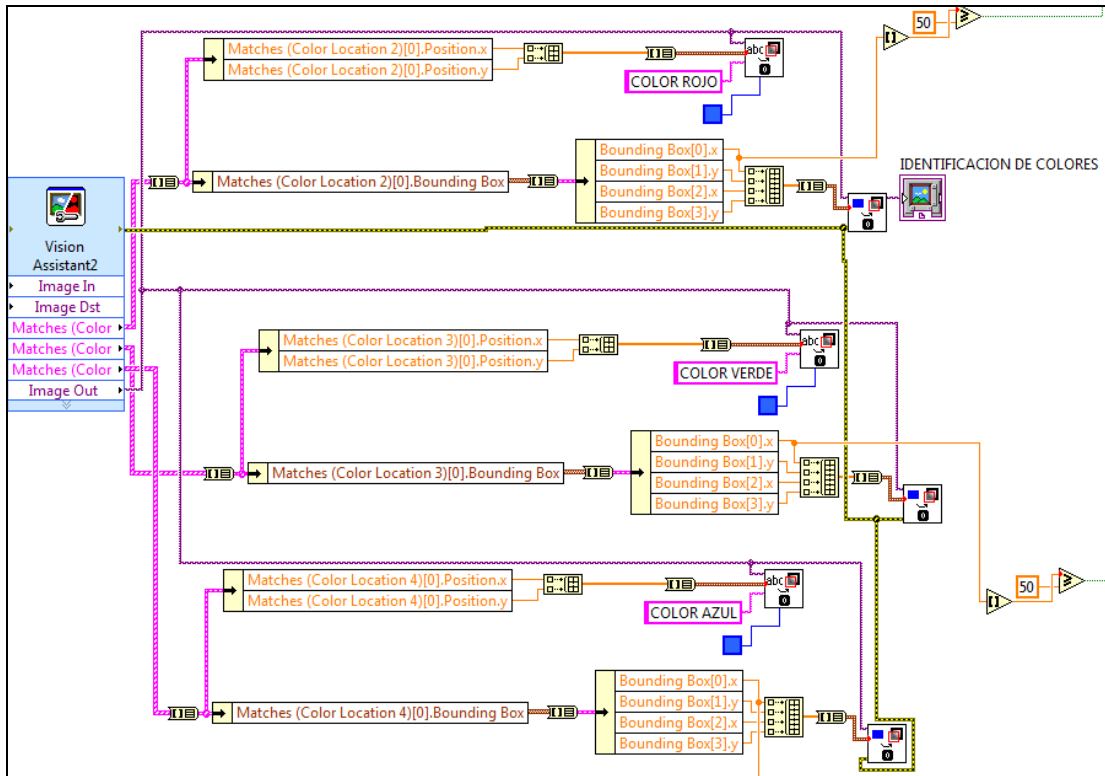


Figura 3. 80: Conexión del asistente de visión con el imaq overlay rectangle y el imaq overlay text

Fuente: Elaboración propia

QUINTO PASO

Flip flop RS temporizado

Se utiliza un flip flop rs temporizado, en cual nos ayuda a detectar solo los flancos positivos, y con los negativos el flip flop queda inactivo.

Diagrama de bloques

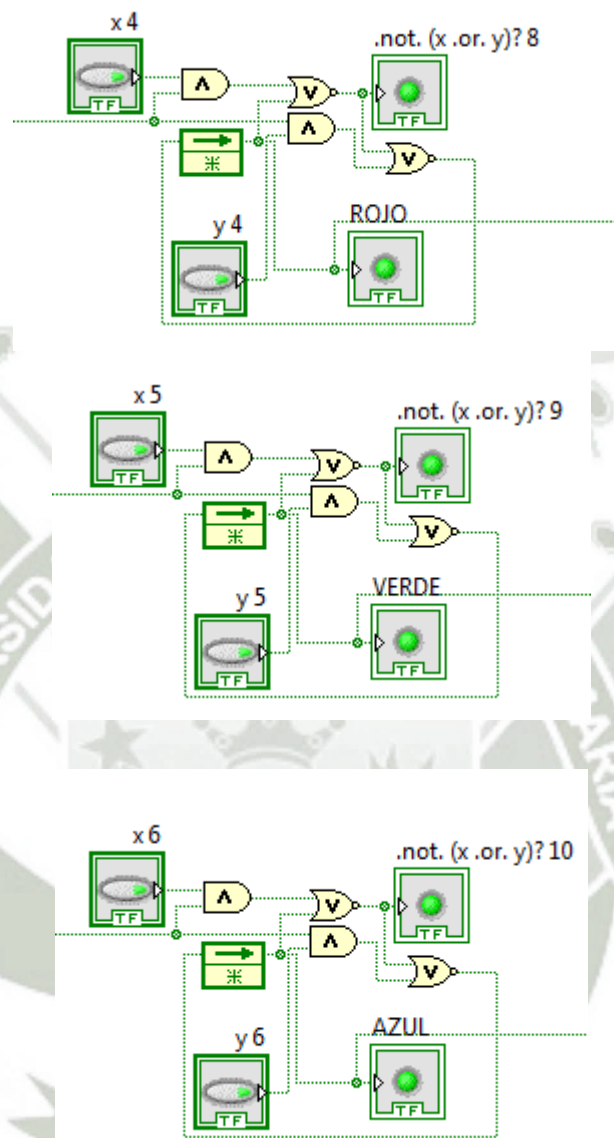


Figura 3. 81: Flip flip rs
Fuente: Elaboración propia

Diagramas en el panel frontal

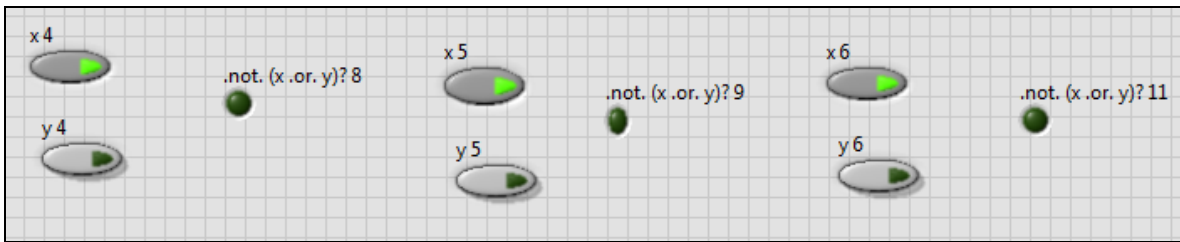


Figura 3. 82: Booleanos del flip flop
Fuente: Elaboración propia

Se prenden los botones x4, x5, x6, para que cumpla la condición deseada de la tabla de verdad del flip flop RS.

SEXTO PASO

3.2.2.4. Control de PWM

Utilizando las librerías de Arduino para servomotores en labview

Visa resource: Especifica el puerto de conexión con el que se va trabajar.



Figura 3. 83: Visa resource
Fuente: Software Labview 2011

Init: Se utiliza para iniciar la conexión de arduino con la interface de labview.

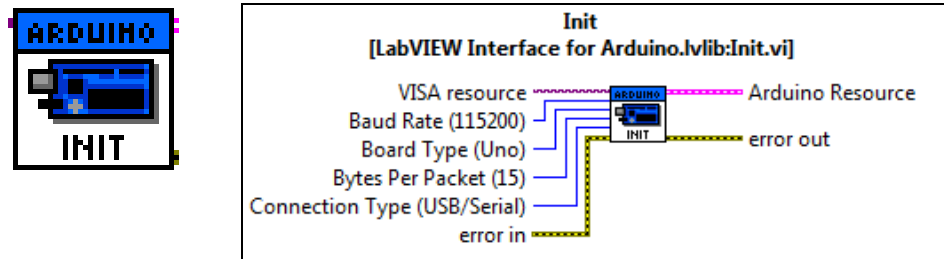


Figura 3. 84: Init
Fuente: Software Labview 2011

Set Number of Servos: Enumera el espacio memoria creada en el arduino para especificar el número de servomotores a utilizar. Se coloca la cantidad de servomotores en el VI.

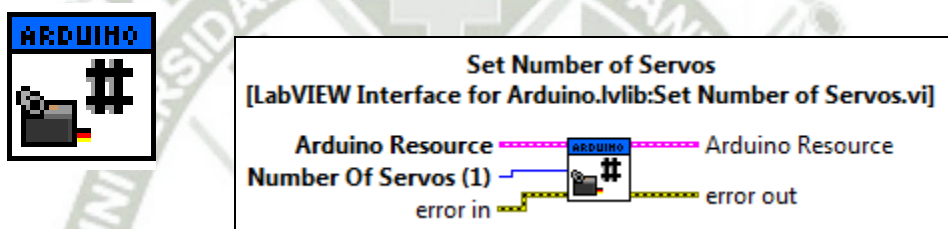


Figura 3. 85: Set Number of Servos
Fuente: Software Labview 2011

Configure Servo: Configura el servo asignando un pin de salida digital, antes de correr el programa se debe especificar el número de servos que se va a utilizar, indicar si es 0, 1, 2, 3, etc

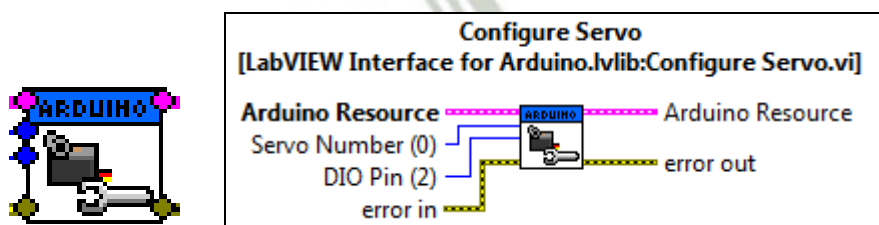


Figura 3. 86: Configure Servo
Fuente: Software Labview 2011

Servo Write Angle: Escribe ángulo que se va a girar el servomotor e indicar el número de servo.

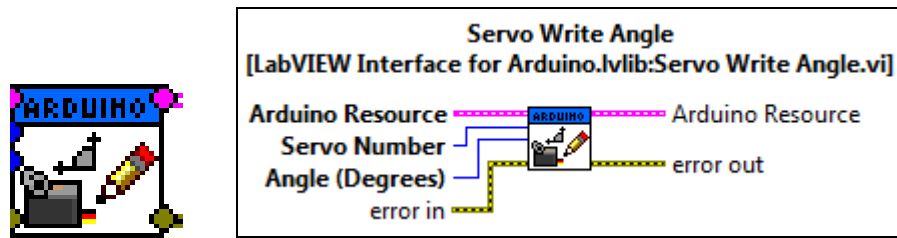


Figura 3. 87: Servo Write Angle
Fuente: Software Labview 2011

Servo read Angle: Lee el último ángulo en grados escrito en el servo correspondiente.

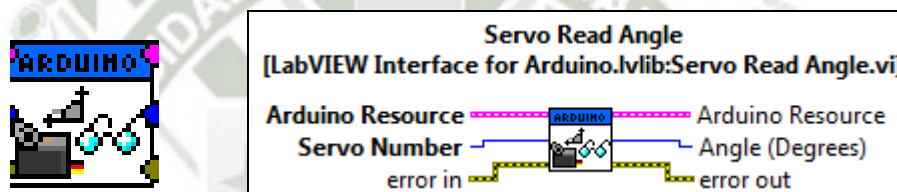


Figura 3. 88: Servo read Angle:
Fuente: Software Labview 2011

Close: Cierra la conexión activa del arduino.

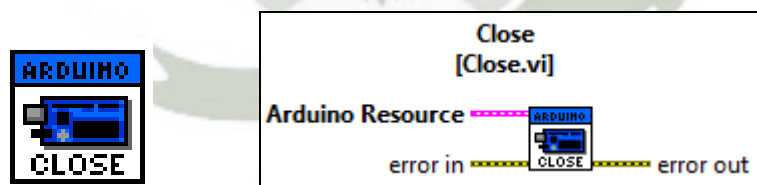


Figura 3. 89: Close
Fuente: Software Labview 2011

Escribir los grados de movimiento del servomotor:

Wait: Espera los segundos indicados en este caso 1000 ms.

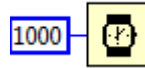


Figura 3. 90: Wait
Fuente: Software Labview 2011

Increment: Incrementa en uno, al valor de entrada.



Figura 3. 91: Increment
Fuente: Software Labview 2011

Equal: Realiza una comparación si el valor de entrada es igual al número fijo.



Figura 3. 92: Equal
Fuente: Software Labview 2011

Select: Si el valor de entrada es F, mandará los grados que se encuentran en la parte inferior y si el valor de entrada es V, mandará los grados de la parte superior.



Figura 3. 93: Select
Fuente: Software Labview 2011

Una vez detectado la figura geométrica, esta envía la señal al ciclo, y comienza a funcionar el bucle de diagrama de bloques. Las iteraciones comienzan a enviar valores y a su vez incrementados en uno, en un tiempo de demora de 1000 ms, la señal llega al selector

que es una señal $F = 0$, la cual permite enviar los grados ubicados en la parte inferior del selector, hasta que pase un tiempo determinado, en este caso es de 30 segundos.

Hace la comparación si $30 = 30$ entonces envía $V=1$ al selector, activando los grados que se encuentran en la parte superior, que es el último movimiento del servomotor 0° grados.

Una vez que tenemos 0° , existe otra condición de comparación, que si es igual 0, se termina el proceso del movimiento del servomotor y a su vez entra a un ciclo que permite apagar el led que corresponde a la figura geométrica detectada, como indicación de término de proceso.



Diagrama de bloques para escribir los ángulos que debe moverse el servomotor de acuerdo al color localizado.

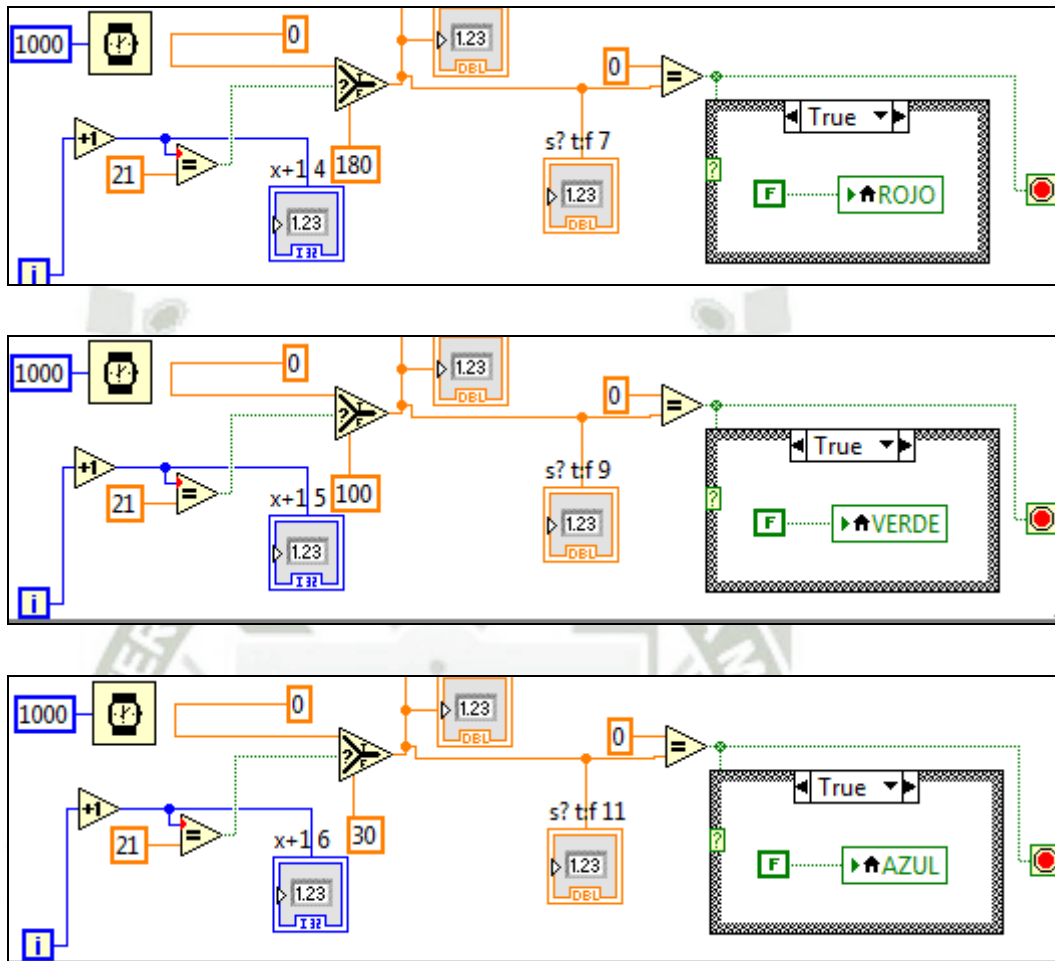


Figura 3. 94: Diagrama de bloques para escribir los ángulos que debe moverse el servomotor de acuerdo al color localizado.

Fuente: Elaboración propia

Diagrama de bloques para el movimiento del servomotor.

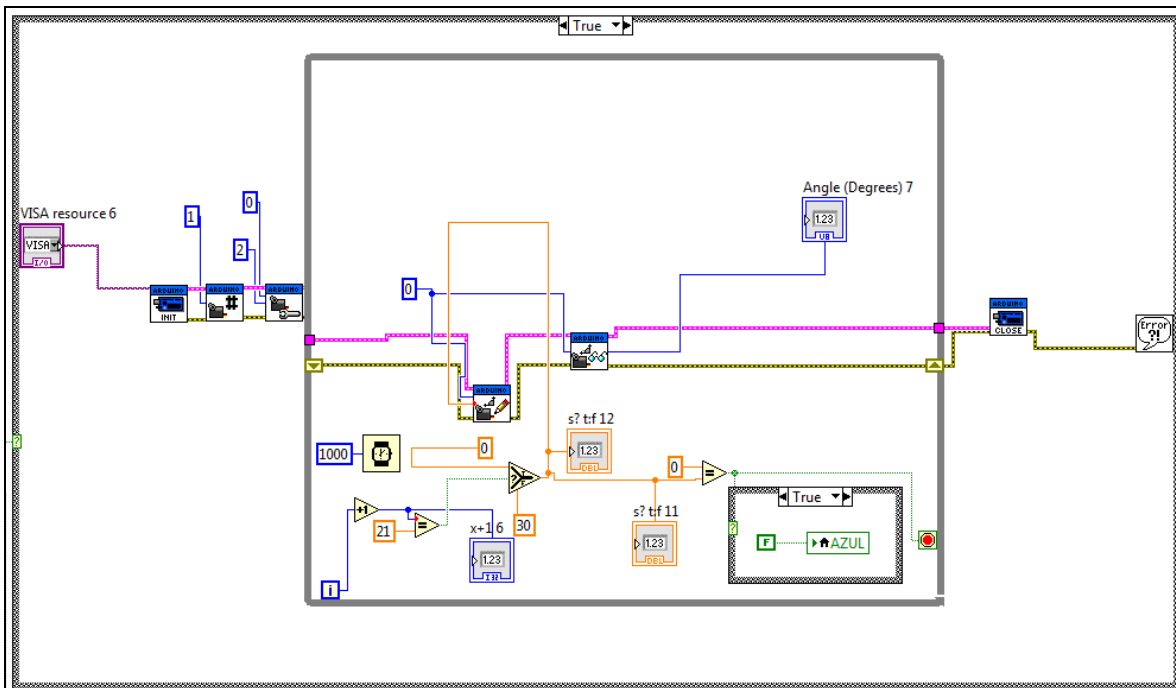


Figura 3. 95: Diagrama de bloques para el movimiento del servomotor.
Fuente: Elaboración propia

3.3. ILUMINACIÓN

Se debe proporcionar condiciones de iluminación uniforme e independiente del entorno para facilitar la extracción de la característica de interés. Existen varios tipos de iluminación cada uno de ellos utilizados para diversos casos:

Iluminación posterior (backlight): Comúnmente se utiliza para la detección de perfiles, impurezas de objetos traslúcidos ya que permite obtener el máximo contraste posible del objeto, en la Figura 3.96 se observa que no es de gran ayuda al momento de resaltar los colores.

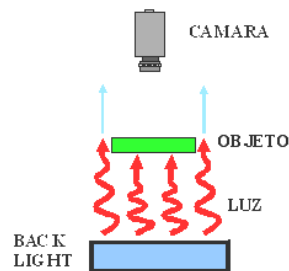


Figura 3. 96: Iluminación posterior
Fuente: Tipos de emisión de luz DCMsistemenes

Iluminación Frontal Oblicua y Direccional: reduce las sombras, se usa para facilitar la detección de diferentes colores, caracteres, etc. En la Figura 2.97 se muestra imágenes de 2 frascos de color amarillo y rojo bajo iluminación frontal oblicua, en la misma se puede apreciar de mejor manera el tono de los colores a analizar.

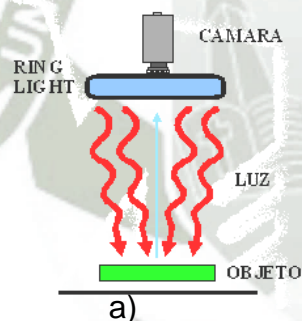


Figura 3. 97: Iluminación Frontal Oblicua
Fuente: Tipos de emisión de luz DCMsistemenes

Existen otros tipos de iluminación tales como: Frontal Axial (difusa), Proyector radial de luz directa cuyo resultado se muestra en la Figura 2.98 literal a) recomendado para la detección de todo lo que suponga un cambio de color, el recuadro b) muestra un esquema de la iluminación directa; la iluminación con proyector de luz difusa se muestra en el recuadro c) es indicado para poder detectar fácilmente variaciones del

color como las que se producen al imprimir texto en un objeto; el esquema de iluminación de campo brillante que se muestra en el recuadro d) facilita la detección de rayas, fisuras y deformaciones en objetos con superficies planas y brillantes, iluminación de Día Nublado y Campo Oscuro.

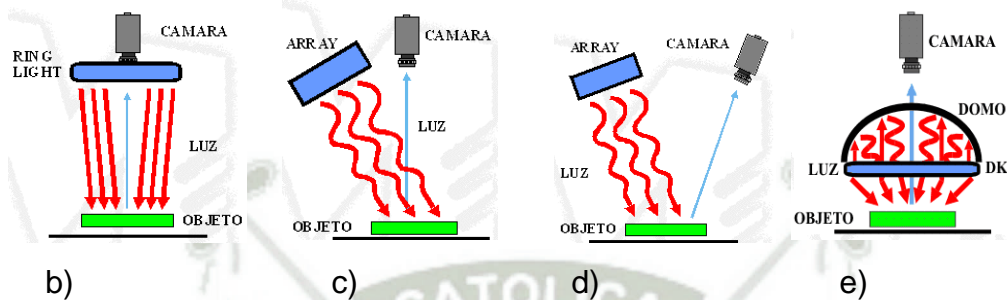


Figura 3. 98: Otros tipos de iluminación
Fuente: Tipos de emisión de luz DCMsistemenes

Fuentes de Luz: Algunas aplicaciones pueden utilizar la luz ambiente. Existen una variedad de fuentes de iluminación para visión artificial que utilizan diversas formas estándares tales como fuentes de luz en forma de anillo, spot, barra y backlight, otras como lámparas de filamento incandescente, halógenas, tubos fluorescentes, láser, etc.



Figura 3. 99: Fuentes de luz
Fuente: Página web spotsline

3.4. MECANISMOS

Partes:

- Banda transportadora.
- Transformador de 24 v y 12 v dc (regulables).
- Placa Arduino Uno.
- Cable usb de conexión placa Arduino Uno.
- Servo motor tower pro sg5010.
- Plataforma circular de 20 cm. (clasificación)
- Protoboard.
- Cables de conexión.
- Laptop.

- Web cam gigaware 2500564.
- Figuras geométricas (triángulo, cuadrado y círculo).
- Fuente de luz (Foco ahorrador de 8 w)
- Estructura de melamine (para crear un ambiente para el paso de la figuras).

Banda transportadora

Una cinta transportadora es un sistema de transporte continuo formado básicamente por una banda continua que se mueve entre dos tambores o poleas.

Existen bandas transportadoras para uso ligero y uso pesado.

La banda es arrastrada por fricción por uno de los tambores, que a su vez es accionado por un motor. El otro tambor suele girar libre, sin ningún tipo de accionamiento, y su función es servir de retorno a la banda. La banda es soportada por rodillos entre los dos tambores.

Debido al movimiento de la banda el material depositado sobre la banda es transportado hacia el tambor de accionamiento donde la banda gira y da la vuelta en sentido contrario. En esta zona el material depositado sobre la banda es vertido fuera de la misma debido a la acción de la gravedad.

Las cintas transportadoras se usan principalmente para transportar materiales granulados, agrícolas e industriales, tales como cereales, carbón, minerales, etcétera, aunque también se pueden usar para transportar personas en recintos cerrados (por ejemplo, en grandes hospitales y ciudades sanitarias). A menudo para cargar o descargar buques cargueros o camiones. Para transportar material por terreno inclinado se usan unas secciones llamadas cintas elevadoras. Existe una amplia variedad de cintas transportadoras, que difieren en su modo de

funcionamiento, medio y dirección de transporte, incluyendo transportadores de tornillo, los sistemas de suelo móvil, que usan planchas oscilantes para mover la carga, y transportadores de rodillos, que usan una serie de rodillos móviles para transportar cajas o palés.

Las cintas transportadoras se usan como componentes en la distribución y almacenaje automatizados. Combinados con equipos informatizados de manejo de palés, permiten una distribución minorista, mayorista y manufacturera más eficiente, permitiendo ahorrar mano de obra y transportar rápidamente grandes volúmenes en los procesos, lo que ahorra costes a las empresas que envía o reciben grandes cantidades, reduciendo además el espacio de almacenaje necesario.

Partes de la faja transportadora

- 1 polea de cabeza (8 x 15 cm)
- 1 polea de cola (8 x 15 cm)
- 6 rodillos
- 5 soportes
- 1 faja (320 cm.)
- Estructura plana para el paso de la faja
- Pernos, tuercas, volandas.
- 1 plataforma de Melamine.
- Motor dc de 24v (regulación de 3v,6v,9v,12v,24v)
- Una correa.

Figuras geométricas.

En la geometría, como disciplina, se distinguen componentes tales como el plano, el punto, la línea -recta, curva, quebrada-, la superficie, el segmento y otros de cuya combinación nacen todas las figuras geométricas.

Entonces, una figura geométrica (también se la puede denominar lugar geométrico) corresponde a un espacio cerrado por líneas o por superficies.

Las figuras geométricas de lados rectos se denominan polígonos y las figuras de lados curvos se denominan círculo y circunferencia y corresponden también a polígonos.

Es importante recordar que las formas sólidas o tridimensionales corresponden a los cuerpos geométricos y se denominan poliedros, como el cubo y la pirámide, y a los cuerpos redondos, como la esfera y el cilindro.

Según las características de las figuras geométricas (polígonos) se pueden establecer varias clasificaciones.

Según la medida de sus lados y ángulos, los polígonos pueden ser regulares e irregulares.

Un polígono es regular si todos sus lados poseen la misma longitud y si todos sus ángulos son iguales.

Figuras geométricas utilizadas: Cuadrado, triángulo y círculo.

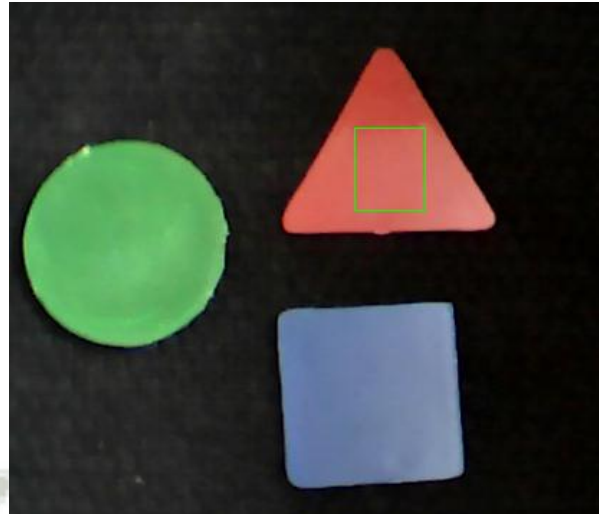


Figura 3. 100: Figuras geometricas
Fuente: Laboración propia

Planos de conexión: Placa Arduino, protoboard, fuente de poder y servomotor

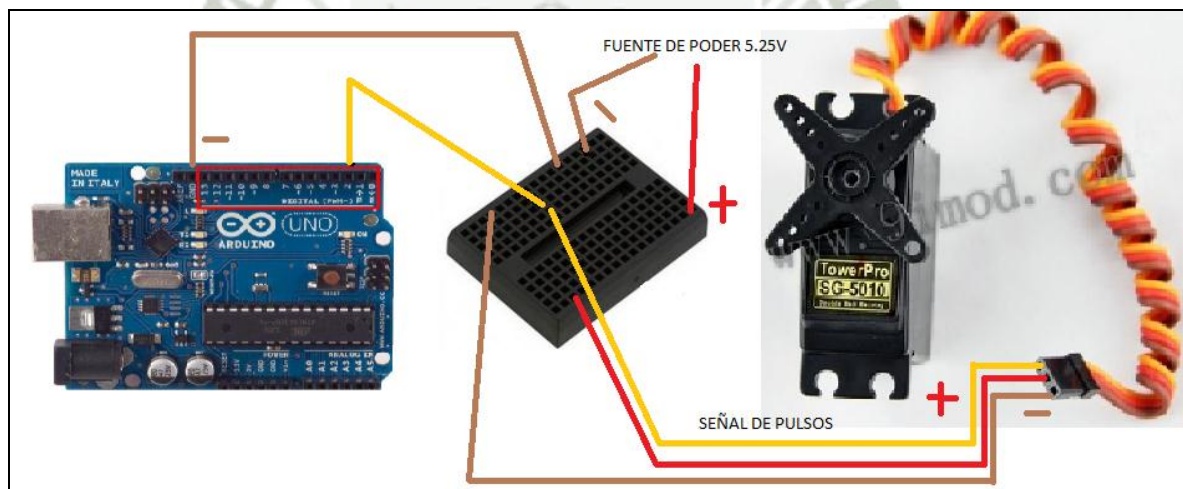


Figura 3. 101: Planos de conexión: Placa Arduino, protoboard, fuente de poder y servomotor
Fuente: Elaboración propia



Figura 3.102: Modulo del proyecto de visión artificial

Fuente: Elaboración propia

CAPITULO IV

PUESTA EN MARCHA

4.1. INICIACIÓN DEL PROGRAMA DE VISIÓN ARTIFICIAL

Primer paso

- Abrir el programa Measurement Automation de National Instrument.
- Hacer click en device and interfaces.
- Hacer click serial and parallel. (en este caso en la ventana de la derecha, se observara que la placa Arduino Uno ha sido detectada por nuestra computadora, y por defecto como com 3).

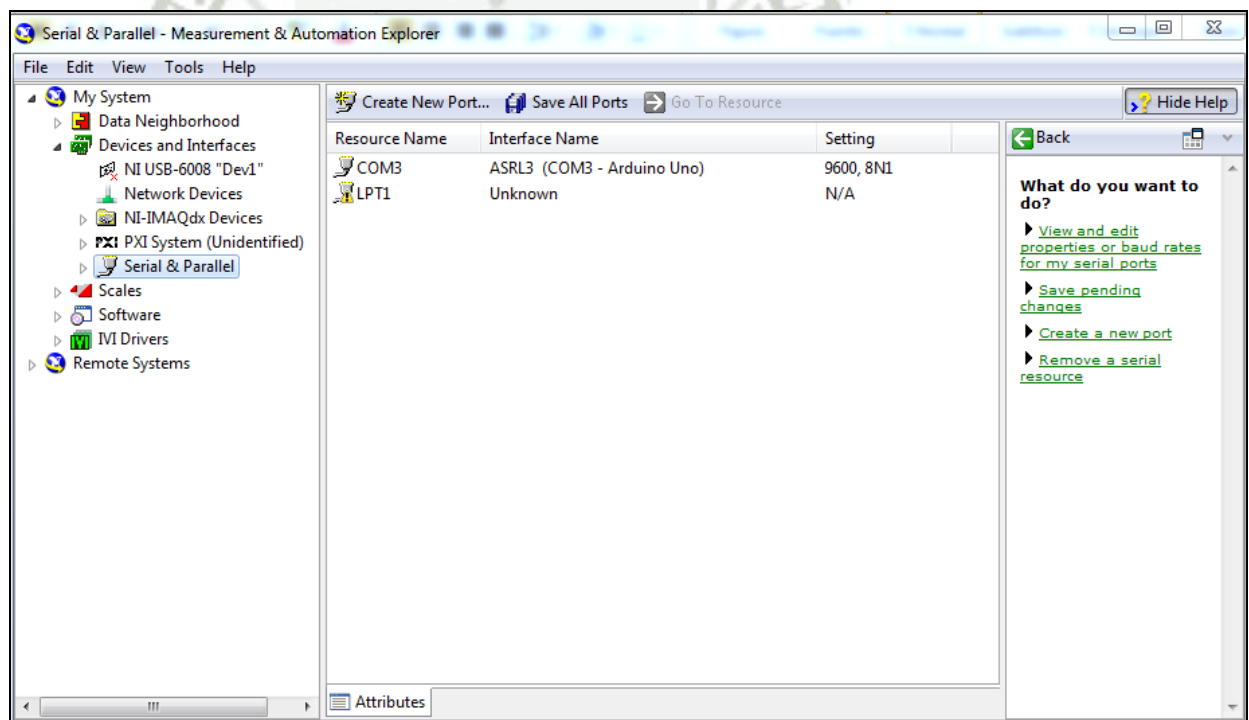


Figura 4. 1: Measurement Automation de National Instrument.
Fuente: Measurement Automation de National Instrument.

Segundo paso

- Abrir el programa Arduino versión 1.0.5, ir al menú herramientas y especificar qué tipo de tarjeta vamos a utilizar, en nuestro caso es Arduino Uno.

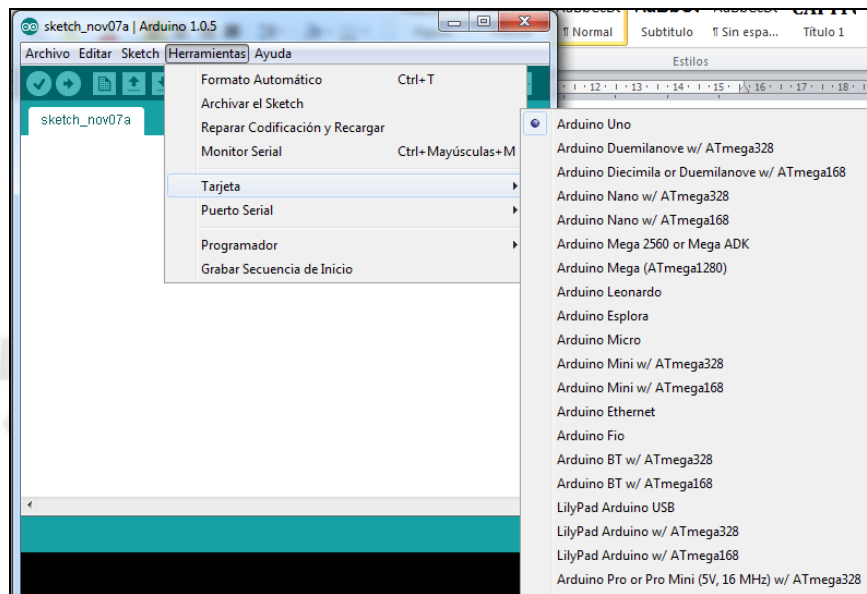


Figura 4. 2: Sketch Arduino 1.0.5 (tarjeta)
Fuente: Elaboración propia

- En el puerto serial se debe especificar la salida en la cual se encuentra la tarjeta Arduino Uno, según el measurement automation , se encuentra en el com 3.

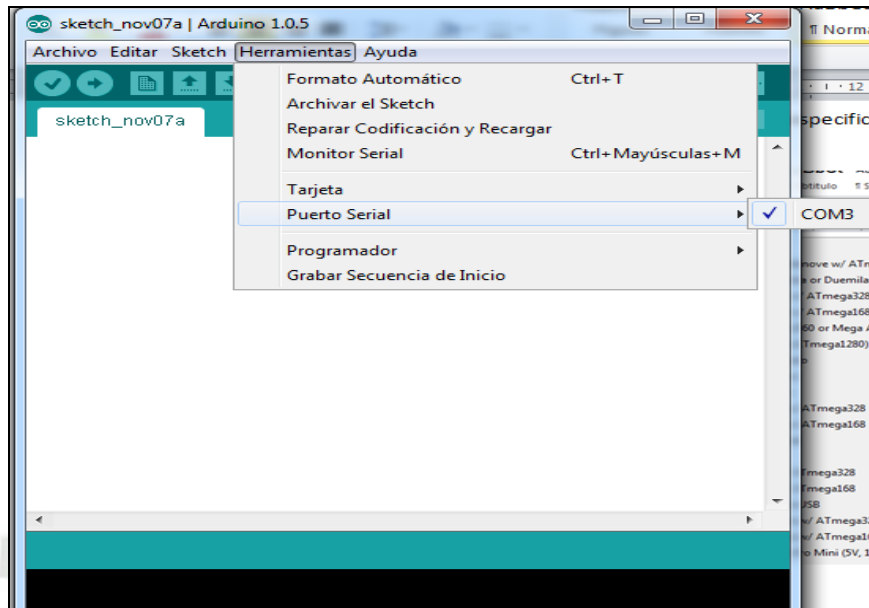


Figura 4. 3: Sketch Arduino 1.0.5 (puerto serial)

Fuente: Elaboración propia

- Ir al menú archivo de la ventana del Arduino, hacer click en abrir y escoger el archivo file del sketch de arduino, LIFE_Base.

Tenemos arduino conectado al usb y reconocido por el sistema, ahora toca programar el arduino con el sketch que comunica la placa con labview. El sketch de arduino se instala con LIFA, por lo que tenemos que ir a la carpeta donde se ha instalado. La ruta es un poco larga, así que mejor la miramos en el pantallazo de la izquierda. En el directorio LVIFA_Base, se encuentra el archivo LVIFA_Base.pde que es el sketch que debemos abrir. Se nos abrirán además todas las librerías. Nuevamente se selecciona el tipo de placa que estamos utilizando, que en nuestro caso es una Arduino Uno, se selecciona desde el menú herramientas, en la opción tarjeta y también la salida del puerto serial que es el com 3.

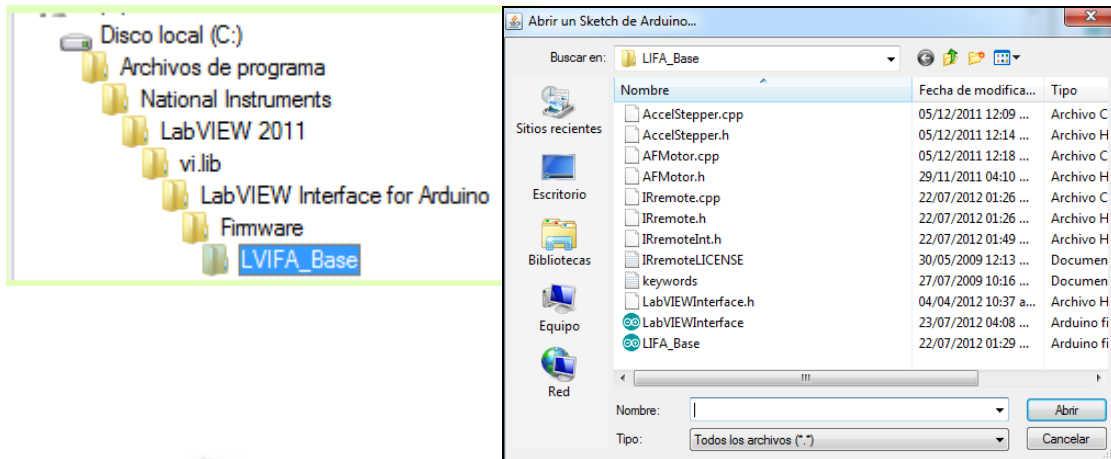


Figura 4. 4: Abrir Lifa_base
Fuente: Elaboracion propia

- Para terminar se puede programar la placa con labview haciendo click en el botón cargar.

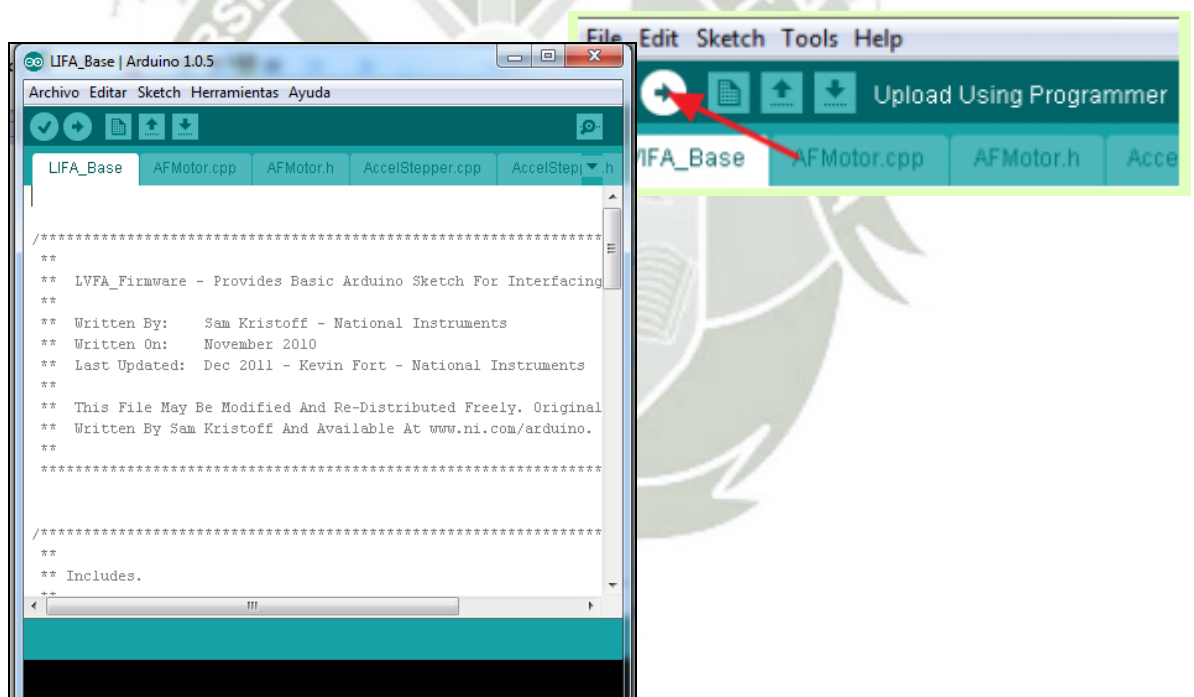


Figura 4. 5: Cargar Lifa Base
Fuente: Elaboración propia

Tercer paso

- Abrir los programas de Identificación y clasificación de figuras geométricas por colores y por formas de labview.

Prender los booleanos x4, x5, x6 tanto como para la identificación por colores y por forma, esto ayudara para que se cumpla la condición deseada en la tabla de verdad del flip flop rs .



Figura 4. 6: Boléanos encendido para iniciar programa

Fuente: Elaboración propia

Para la utilización del servomotor se tiene que especificar la salida mediante visa source, el cual nos permite escoger el puerto de salida, como ya hemos visto anteriormente se utilizando el com 3. Entonces si tenemos 3 ventanas visa source por programa, que son para los 3 movimiento del servomotor según lo identificado, en todos escogemos la salida de puerto com 3, que corresponde a la placa del Arduino Uno y a su vez esta comunicara al servomotor.

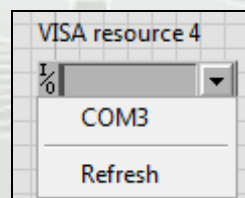


Figura 4. 7: Visa resource

Fuente: Elaboracion propia

Cuarto Paso

- Correr el programa

CONCLUSIONES

- Se diseñó un correcto sistema de identificación y clasificación por visión artificial
- Se Implementó adecuadamente algoritmos y diagrama de bloques para realizar un correcto control que permita comparar patrones para la identificación y clasificación de objetos.
- Se Implementó un correcto prototipo de visión artificial en el cual se pueda visualizar los objetos mediante el software Labview y a su vez identificarlos y clasificarlos mediante controladores.
- Este proyecto aporta información en el campo de Visión artificial, constituyéndose una herramienta con fines educativos.
- La tonalidad de los colores juega un papel importante en la detección de colores, para así poder ser diferenciados, con la ayuda de parámetros externos, para obtener una mejor calidad de imagen.

RECOMENDACIONES

- Crear un ambiente para que se mantengan siempre las mismas condiciones, ubicando correctamente la fuente de luz para que ayude al proceso de captación de imágenes y utilizando fuentes de luz que no emitan reflejo en los objetos.
- Definir correctamente los patrones, utilizando colores de tonalidades fuertes en los objetos y contar con objetos que tengan un contorno bien definido.
- Posicionar correctamente la perilla de la cámara web, para obtener una imagen de video clara.
- Si se requiere utilizar velocidades mayores, se tiene que contar con una cámara web de mejor resolución de video y una computadora con un procesador mayor.
- La cámara web solo debe estar enfocada en el área donde se va a realizar la detección.
- Para proyectos futuros de visión artificial, se puede partir desde esta investigación y en su defecto agregar parámetros en el desarrollo de la programación que me permitan discriminar los objetos, que no contemplan mis patrones, posiblemente con la utilización de pistones, el cual permitirá retirar de la faja transportadora las figuras geométricas que no correspondan a mis patrones pregrabados.

BIBLIOGRAFIA

- Álvarez Anton Juan Carlos y otros. “Instrumentación virtual con LabVIEW”. Editorial Servicio de Publicaciones de la Universidad de Oviedo. 1° edición. 2009
- F.J.Cortijo Bon, “Introducción al reconocimiento de formas”.
- Frenzel, Louis L.. Sistemas electrónicos de comunicaciones (Tercera reimpresión edición). México D.F.: Alfaomega. mayo de 2003.
- Gonzalez, Rafael C. –Woods, Richard E. “Digital Image Processing” Second Edition, Using Matlab,
- Klinger T., "Image Processing with Labview and Imaq Vision", Editorial Prentice Hall, 2003.
- Palacios Municio, Enrique - Remiro Dominguez, Fernando - Lopez Perez, Lucas José. “Microcontrolador PIC16F84. Desarrollo de proyectos”. Editorial RA-MA S.A. Editorial y Publicaciones, 3ª edición. 2009
- R.O. Duda, P.E. Hart, D.G. Stork. Capítulo 1. "Pattern classification". Editorial Wiley, 2000.
- Albornoz Carrasco, Juan y Lagos Norabuena, Pablo. Guía de iniciación de labview, universidad de la frontera, departamento de Ing electrica.
- Centro Regional de Optimización y Desarrollo de Equipo Chihuahua. “Programación básica en labview”, agosto 2004, (adjunto PDF).
- Data sheet servomotor tower pro sg5010, www.datasheet4u.com.Tabla[2]
- National Instrument, IMAQ Vision for LabVIEW™, , 2003.
- National Instruments, IMAQ Vision for Measurement Studio™ User Manual, 2003.

- National Instruments. “Análisis y procesamiento de imágenes”, 2003.
- Universidad de Magallanes, Facultad de Ingeniería. “Sistemas de monitoreo y gestión de variables industriales utilizando dspic y labview”
- Universidad Nacional de Quilmes – Ing. en Automatización y Control Industrial. “Aspectos de un sistema de visión artificial”, Cátedra: Visión Artificial Octubre de 2005,Tabla [1].

