

**SISTEMA DE INSPECCIÓN DIGITAL PARA MEDICIÓN
DIMENSIONAL Y GEOMÉTRICA EMPLEANDO VISIÓN ARTIFICIAL
COMPATIBLE CON ARQUITECTURA DE MANUFACTURA EN LAZO
CERRADO**

BRAYAN STIVEN FIGUEROA BETANCOURTH

Autor



**UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
PROGRAMA DE INGENIERÍA MECATRÓNICA
PAMPLONA
2022**

**SISTEMA DE INSPECCIÓN DIGITAL PARA MEDICIÓN
DIMENSIONAL Y GEOMÉTRICA EMPLEANDO VISIÓN ARTIFICIAL
COMPATIBLE CON ARQUITECTURA DE MANUFACTURA EN LAZO
CERRADO**

**TRABAJO DE GRADO PARA OPTAR POR EL TÍTULO DE
INGENIERO EN MECATRÓNICA**

BRAYAN STIVEN FIGUEROA BETANCOURTH

Autor

CRISTHIAN IVAN RIAÑO JAIMES

Director

CESAR AUGUSTO PEÑA CORTES

Co-Director

**UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
PROGRAMA DE INGENIERÍA MECATRÓNICA
PAMPLONA**

2022

Dedicatoria

Mi madre Luz Marina

Mis hermanos Diego Orlando, Jhon Jairo y José Julián

*Ellos me han brindado su apoyo y motivación tanto en
mi formación académica como en mi proyecto de vida.*

BRAYAN STIVEN FIGUEROA BETANCOURTH

Agradecimientos

Principalmente me agradezco a mí mismo por el conocimiento que he adquirido con esfuerzo y dedicación para avanzar en este propósito.

Al PhD. Cristhian Ivan Riaño Jaimes por la orientación para llevar a cabo la realización de este trabajo y contribución de su conocimiento en mi formación académica e investigativa.

Al PhD. Cesar Augusto Peña Cortes por la orientación e impartición de conocimiento para realizar este trabajo, la formación académica y fomentar la investigación.

Al PhD. Alberto José Alvarez por su orientación para la realización de las pruebas en los laboratorios de la Universidade de Brasilia.

A todos los docentes del programa de ingeniería mecatrónica de la Universidad de Pamplona por contribuir en mi formación académica.

BRAYAN STIVEN FIGUEROA BETANCOURTH

RESUMEN

En este proyecto de investigación se presenta el desarrollo de un sistema de medición compatible con tecnologías digitales y basado en visión artificial para operar dentro de una arquitectura en lazo cerrado, útil para compensar errores de manufactura. El requerimiento principal para el sistema de medición consiste en determinar los errores de desviación tanto dimensionales como geométricos en piezas prismáticas. En este documento, se presenta la identificación de las variables que intervienen en el proceso de medición, procedimiento para determinar el error, representación digital de cada característica medida, algoritmos para el reconocimiento automático de características y codificación de la transferencia de datos entre sistemas CAD/CAM. El sistema debe poseer una medición local (en máquina) y la posibilidad de ser utilizado en sitio fuera de la máquina. La transferencia de datos debe promover la integración de sistemas y superar barreras de interoperabilidad empleando lenguajes neutros para codificar resultados.

ABSTRACT

This research project seeks to develop a measurement system compatible with digital technologies and based on an artificial vision to operate within a closed-loop architecture that provides data to compensate for manufacturing errors. The main requirement for the measurement system is to determine both dimensional and geometric deviation errors in prismatic parts. The variables involved in these measurement processes must be identified, the error determined, each measured feature digitally represented, automatic feature recognition algorithms applied, and data transfer between CAD/CAM systems codified. The system must have a local measurement (on the machine) and the possibility of being used on-site outside the machine. Data transfer should promote system integration and overcome interoperability barriers using neutral languages to encode results.

Contents

1	Introducción	13
1.1	Planteamiento Del Problema	13
1.2	Propósito de Estudio	14
1.3	Objetivos de la Tesis	14
1.3.1	Objetivo General	14
1.3.2	Objetivos Específicos	14
1.4	Estructura del Libro	14
1.4.1	Capítulo 1	14
1.4.2	Capítulo 2	15
1.4.3	Capítulo 3	15
1.4.4	Capítulo 4	15
1.4.5	Capítulo 5	15
2	Marco de Referencias y Estado del Arte	16
2.1	Introducción	16
2.2	Marco de Referencias	16
2.2.1	La Manufactura y sus Principales Problemas	16
2.2.2	Máquinas Sofisticadas para Procesos de Manufactura	16
2.2.3	Métodos para la Compensación de Errores	17
2.2.4	Metrología	17
2.2.5	Estándares para Intercambio de Información	18
2.2.6	Métodos de Inspección Novedosos	18
2.2.7	Tolerancias GD&T	19
2.2.8	Sistemas Embebidos	23
2.3	Estado del Arte	24
2.3.1	Investigaciones de Ámbito Nacional	24
2.3.2	Investigaciones de Ambito Internacional	26
3	Arquitectura de Medición Asistida por Computadora	29
3.1	Introducción	29
3.2	Arquitectura de Medición Asistida por Computadora	29
3.3	Diseño de la Feature o Pieza de Estudio	31
4	Construcción y Ensamble del Sistema de Inspección Digital para Medición Dimensional y Geométrica	34
4.1	Introducción	34
4.2	Estructura del Soporte Para el Sistema De Inspección	35
4.3	Diseño CAD de Componentes Electrónicos y ensamble de elementos que Conforman el Sistema	38
4.4	Conexiones del Sistema de Inspección Digital	41
4.4.1	Conexiones para la Adquisición de Imágenes	41
4.4.2	Conexión final del sistema de inspección digital	43

5	Desarrollo del Sistema de Visión Artificial para la Inspección Digital en la Medición Dimensional y Geométrica	44
5.1	Introducción	44
5.2	Algoritmo para la Adquisición de Imágenes	45
5.2.1	Fase de Carga Del Algoritmo	46
5.2.2	Fase de Prueba del Algoritmo	47
5.3	Algoritmo para la Inspección Digital, Dimensional y Geométrica Empleando Técnicas de Visión Artificial	48
5.3.1	Obtención de Imágenes con MATLAB desde Dirección IP	49
5.3.2	Calibración de la Cámara del Sistema	49
5.3.3	Implementación de Técnicas de Visión Artificial para la Inspección Digital	51
5.3.4	Detección de la Zona de Trabajo	52
5.3.5	Detección de las 4 Esquinas de la Zona de Trabajo	55
5.3.6	Homografía 2D	56
5.3.7	Inspección de la Pieza o Feature	58
5.3.8	Reporte de Inspección	71
6	Implementación y Casos de Estudio	73
6.1	Introducción	73
6.2	Casos de Estudio	73
6.2.1	Feature o Pieza de Estudio	73
6.2.2	Resultados Obtenidos Sistema de Inspección Digital	75
6.2.3	Resultados Obtenidos Máquina de Medición por Coordenadas	76
6.2.4	Resultados Obtenidos Máquina de Medición por Coordenadas Visión Computacional	76
6.2.5	Análisis de Resultados	77
7	Conclusiones y Trabajos Futuros	78
7.1	Conclusiones	78
7.2	Trabajos Futuros	79
8	Apéndices y Anexos	83
8.1	Algoritmo Adquisición de imágenes	83
8.2	Calibración de la cámara	85
8.3	Detección de la Zona de Trabajo	86
8.4	Detección 4 Esquinas en la Zona de Trabajo	87
8.5	Homografía 2D	88
8.6	Inspección	89

List of Tables

1	Cableado de Fases de Carga y Prueba	43
2	líneas Regresión Lineal	63
3	Características a Inspeccionar de la Feature Caso de Estudio	74
4	Resultado Sistema de Inspección Digital Caso de Estudio	75
5	Resultados Maquina de Medición por Coordenadas Caso de Estudio . . .	76
6	Resultados Máquina de Medición por Coordenadas Visión Computacional Caso de Estudio	76

List of Figures

1	Perpendicularidad GD&T	20
2	Paralelismo	20
3	Circularidad	21
4	Linealidad	22
5	Posición	22
6	Módulo ESP32-CAM	23
7	Componentes del módulo ESP32-CAM	24
8	Arquitectura de Medición	30
9	CAD Feature o Pieza de Estudio	32
10	Plano de la Feature o Pieza de Estudio	32
11	Feature o Pieza de Estudio	33
12	CAD Soporte Sistema de Medición	35
13	Soporte Sistema de Medición	36
14	CAD Tapa Soporte Sistema de Medición	36
15	Tapa Soporte Sistema de Medición	37
16	CAD ESP32 CAM	38
17	CAD Batería LiPo	39
18	CAD Diodo LED	39
19	CAD Sistema de Inspección Digital	40
20	Sistema de Inspección Digital	40
21	Conexión ESP32 CAM y ARDUINO Fase de Carga	41
22	Conexión ESP32 CAM y ARDUINO Fase de Prueba	42
23	Conexión Sistema	43
24	Diagrama de Flujo Algoritmo para Adquisición de Imágenes	45
25	Modo Flash del ESP32 CAM	47
26	Asignación Dirección IP del ESP32 CAM	48
27	Imagen Obtenida con MATLAB desde la Dirección IP	49
28	Diagrama de Flujo Calibración Cámara	50
29	Imagen Patrón con mejoría en Distorsión Focal	51
30	Diagrama de Flujo Estructura Algoritmo Inspección Digital	52
31	Porción de la Zona de Trabajo con Respectivo Color	53
32	Porción de la Zona de Trabajo con Respectivo Color	53
33	Imagen Binaria de la Zona de Trabajo	54
34	Detección de Esquinas en Imagen Binaria de la Zona de Trabajo	55
35	Detección de Esquinas en Imagen RGB de la Zona de Trabajo	56
36	Homografía Zona de Trabajo	57
37	Porción de la Feature con Respectivo Color	58
38	Histograma de Colores Porción de la Feature	59
39	Imagen Binaria de la Feature	60
40	Imagen Patrón para calcular Paralelismo y Perpendicularidad	61
41	Fijación de Puntos en Límites de la Feature	62
42	Regresión Lineal de Puntos Fijados en Límites de la Feature	63
43	Imagen Binaria Negada de la Feature	68
44	Imagen Patrón para Circularidad	69
45	Centroide y Límites de la Región Circular en la Feature	70
46	Exportación de los Resultados de Inspección	71
47	Resultados Obtenidos con MATLAB	72
48	Resultados Exportados a Excel	72
49	Objetos de la Feature Caso de Estudio a Inspeccionar	74
50	Plano de la Feature de Estudio	75

ACRÓNIMOS

CAD: acrónimo de Diseño Asistido por Computadora.

CAM: acrónimo de Manufactura Asistida por Computadora.

CNC: acrónimo de Control Numérico Computarizado.

ISO: acrónimo de Manufactura Asistida por Computadora.

STEP: Estándar para el Intercambio de Modelo de Datos de Producto.

QIF: Acrónimo de Quicken Interchange Format.

GD&T: Acrónimo de Dimensionamiento Geométrico y Tolerancia.

ESP32CAM: sistema embebido el cual permite la captura y envío de imágenes por medio de sus protocolos de comunicación.

IoT: Acrónimo de Internet of Things o Internet de las Cosas.

XML: Acrónimo de Extensible Markup Language o Lenguaje de Mercado Extensible.

2D: acrónimo mencionado para la identificación de dos dimensiones en el espacio, las cuales corresponden en un plano cartesiano a las coordenadas xy .

3D: acrónimo mencionado para la identificación de tres dimensiones en el espacio, las cuales corresponden en un plano cartesiano a las coordenadas x, y, z .

Arduino: tarjeta de desarrollo embebida la cual permite la interacción entre dispositivos electrónicos mediante la adquisición y envío de señales, esta cuenta con código abierto.

WIFI: Tecnología de forma inalámbrica la cual permite transmisión y recepción de información entre dispositivos distintos.

MATLAB: Sistema de computo numérico el cual proporciona un entorno para el desarrollo integrado, este cuenta con un lenguaje propio de programación.

LiPo: Tipo de batería compuesta de Polímero de Litio.

1 Introducción

La industria 4.0 impulsa avances científicos y mejora los procesos de fabricación, buscando disminuir la intervención humana y aumentando la eficiencia, obteniendo como resultado la fabricación avanzada y conllevando así a nuevos desafíos de integración e interoperabilidad para los cuales la integración de sistemas digitales es fases de procesos de producción es una solución tecnológica. La integración de sistemas digitales recolecta datos e información necesaria para brindar ayuda en la toma de decisiones, además de establecer comunicación interpretable fácilmente por el operador.

Los sistemas de medición recolectan información que al ser retroalimentada ayuda significativamente en la toma de decisiones para el diseño y fabricación. Los datos de tolerancias se pueden generar empleando la inspección dimensional y geométrica, estos datos poseen información sobre la desviación de una pieza fabricada. Se puede determinar las propiedades superficiales de una pieza realizando una medición específica. Los factores como la calidad, la precisión y la repetibilidad en la fabricación de las piezas repercuten de manera positiva en el entorno de fabricación controladamente.

1.1 Planteamiento Del Problema

En los procesos de manufactura el error presente en el diseño de piezas afecta significativamente la producción, incurriendo en pérdidas económicas para la industria manufacturera, actualmente se han desarrollado distintas técnicas de inspección enfocadas en garantizar que estas piezas diseñadas cumplan con estándares y tolerancias de error establecidas. Algunas de las técnicas más avanzadas requieren de intervención humana haciendo que se tenga presente el error humano dentro de este proceso de medición, afectando la eficiencia y eficacia, en la era digital e industria 4.0 se contempla reducir la intervención humana y automatizar los procesos con el fin de darle robustez a los procesos de manufactura y evitar este tipo de afectaciones en el proceso de producción. En la mayoría de estos métodos de medición se cuenta con herramientas fijas por su tamaño considerable y de difícil transporte y adaptabilidad a cualquier proceso de manufactura, haciendo imposible una inspección dentro del proceso, puesto que solo se puede realizar la inspección de la pieza una vez diseñada por aparte y luego ajustar la máquina empleada para el proceso de manufactura de forma manual.

1.2 Propósito de Estudio

El propósito de estudio de esta tesis es diseñar e implementar un sistema de inspección digital para la medición dimensional y geométrica en procesos de manufactura, buscando precisión, economía y una fácil adaptabilidad a cualquier proceso. Este trabajo de investigación tiene como objetivo implementar técnicas de visión artificial para inspeccionar las piezas resultantes de los procesos de manufactura, buscando disminuir la interacción humana en estos, además de hacer un tratamiento de los datos para obtener una medida de las piezas en tiempo real.

1.3 Objetivos de la Tesis

1.3.1 Objetivo General

Desarrollar un sistema de medición compatible con tecnologías digitales utilizando visión artificial.

1.3.2 Objetivos Específicos

- Identificar las variables que intervienen en los procesos de manufactura y como emplear los sistemas de medición dentro de una arquitectura digital para corregir errores.
- Formular una estrategia de medición compatible con los conceptos de manufactura digital.
- Diseñar un sistema de medición que consiga obtener datos de dimensiones y geometrías de una pieza.
- Implementar un sistema de medición para corrección de errores en la fabricación.

1.4 Estructura del Libro

1.4.1 Capítulo 1

Define las tolerancias dimensionales y geométricas más utilizadas en los procesos de manufactura y su método de cálculo, además de una revisión de antecedentes estructurados de forma temporal.

1.4.2 Capítulo 2

Plantea una arquitectura propuesta de inspección y adquisición de parámetros de las piezas con el fin de implementar la medición asistida por computadora.

1.4.3 Capítulo 3

Detalla la construcción y el ensamble del sistema de inspección digital para medición dimensional y geométrica.

1.4.4 Capítulo 4

Presenta el desarrollo del algoritmo con técnicas de visión artificial para el sistema de inspección digital.

1.4.5 Capítulo 5

Implementación y casos de estudio del sistema de inspección para medición dimensional y geométrica en procesos de manufactura.

Capítulo 1

2 Marco de Referencias y Estado del Arte

2.1 Introducción

En este capítulo se presenta una revisión bibliográfica con el avance a lo largo de la historia en los procesos de manufactura; se hace un acercamiento a las máquinas sofisticadas, los métodos de inspección en procesos de fabricación, estándares de comunicación, procesos de manufactura en la industria 4.0 y sus desafíos actuales.

2.2 Marco de Referencias

2.2.1 La Manufactura y sus Principales Problemas

Los procesos de manufactura nacen como un mecanismo o proceso en el cual los materiales se transforman en artículos útiles para la sociedad, también es considerada como la estructuración y organización de acciones que permiten a un sistema lograr una tarea determinada [1]. Según el artículo *“Return on engineering: Design to cost for electric engine production”* Los costos de un producto están determinados por el desarrollo de la producción de este con un porcentaje del 70%. Para los fabricantes, una de las problemáticas mayormente presentes en la producción es la pérdida de materia prima, conllevando a afectaciones económicas de forma significativa [2].

2.2.2 Máquinas Sofisticadas para Procesos de Manufactura

Existen actualmente herramientas sofisticadas para los procesos de manufactura como las CNC e impresoras 3D con diversas aplicaciones, algunos de estos avances que se pueden encontrar en la literatura son: *“Diseño de una máquina herramienta de 3-ejes con tecnología CNC para operaciones de maquinado”* [3], *“Making CNC machine tools more open, interoperable and intelligent – a review of the technologies”* en el cual se plantea que el objetivo de la próxima generación de máquinas controladas numéricamente por computadora (CNC) es ser portátiles, interoperables y adaptables [4] y *“procesos de manufactura con tecnología 3D”* donde se describen y detallan los tipos de impresoras 3D que se pueden encontrar actualmente en el mercado teniendo presente la norma estándar

internacional ISO/ASTM 52900-2015, por otro lado, se mencionan algunos tipos de manufactura aditiva como lo son: Inyección de aglutinante, Deposición de energía directa, Extrusión de material, Inyección de material, Fusión por lecho de polvo, Laminación de hojas y fotopolimerización [5].

2.2.3 Métodos para la Compensación de Errores

En los procesos de manufactura, la intervención humana y distintos factores presentes en el proceso tienden a presentar errores, para lo cual se han desarrollado métodos para su inspección. En el artículo titulado *“Introducing engineering undergraduates to CNC machine tool error compensation”* presenta el método de compensación de errores de una manera accesible para el estudiante de ingeniería, este se basa en un modelo de matriz de transformación homogénea (MTH) el cual cuantifica los errores geométricos de una máquina herramienta CNC. El tratamiento matemático se reduce a solo elementos esenciales para enfatizar la comprensión física. Este método se aplicó a una fresadora de tres ejes para poder permite al estudiante universitario comprender el concepto con facilidad, además este modelo se desarrolla a partir de primeros principios, lo que no requiere que el estudiante invoque ninguna relación empírica [6].

2.2.4 Metrología

La metrología dimensional es una parte importante de cualquier sistema de fabricación. Consta de distintos componentes y requiere una base de conocimientos grande, diversa e interconectada. Cómo pasar información sin problemas con un costo mínimo y una pérdida de datos mínima entre diferentes componentes de un sistema de metrología dimensional es un problema importante que preocupa a los proveedores de software y hardware, desarrolladores de estándares y clientes. El artículo *“Dimensional metrology interoperability and standardization in manufacturing systems”* se centra en los cuatro elementos principales de un sistema de metrología dimensional: definición del producto, definición del plan del proceso de medición, ejecución del proceso de medición y análisis e informes de datos de calidad. Estos temas se discuten en relación con la situación actual en el desarrollo de estándares de metrología dimensional, como los son los estándares STEP (ISO 10303) los cuales son el producto de un esfuerzo internacional para lograr la interoperabilidad de los sistemas de fabricación, siendo esta una forma adecuada de resolver el problema de interoperabilidad dentro de los sistemas de metrología dimensional. Se propone un mayor desarrollo de los estándares STEP para que la información de Tolerancias y dimensiones geométricas (GD&T) ya disponible en STEP pueda vincularse con información de características de fabricación, tecnología de medición y resultados de medición. El modelo de datos STEP propuesto es un intento de proporcionar un estándar

que admita la generación automática del plan de proceso de medición para la medición en proceso en la máquina [7].

2.2.5 Estándares para Intercambio de Información

El estándar ISO 10303, también conocido como STEP (Standard for the Exchange of Product model data) ha sido diseñado para intercambiar de forma neutral datos relacionados con productos con el objetivo de compartir fácilmente la información de características de formulario entre múltiples socios de una empresa virtual [8]. En su tesis doctoral titulada *“Conceptual framework for closed-loop inspection based on the coordinate measurement technology adherent to step-nc”* el autor desarrolla una implementación computacional utilizando el entorno Java para manipular esquemas EXPRESS, generar bibliotecas con entidades, funciones, métodos y desarrollar una aplicación que permita leer, escribir y modificar archivos neutrales de intercambio STEP. Se expone la metodología seguida a través de un marco conceptual práctico para generar nuevas aplicaciones basadas en la norma ISO 10303. Se presentan tres casos de estudio para verificar la integración e interoperabilidad de la arquitectura en lazo cerrado [9].

QIF (Quicken Interface Format) es un estándar libre creado por el Consorcio de Estándares de Metrología Digital de marco XML unificado para sistemas QIF de calidad asistidos por computadora. QIF permite la captura, el uso y la reutilización de información relacionada con la metrología en los dominios de gestión del ciclo de vida del producto (PLM) y gestión de datos del producto (PDM) [10]. QIF modela las características de calidad y las funciones de medición, además puede cubrir casos de uso que incluyen ingeniería inversa, medición de calidad por lotes y medición de calidad discreta [11].

2.2.6 Métodos de Inspección Novedosos

Uno de los propósitos de los avances investigativos en torno a los sistemas de medición es buscar adaptabilidad e integración a distintos procesos de manufactura con el fin de obtener precisión en la medición, En el artículo titulado *“Integrated precision inspection system for manufacturing of moulds having CAD defined features”* Se desarrolla un sistema integrado de inspección de precisión para la fabricación de moldes con características definidas por CAD para lo cual se emplean estrategias de puntos de muestreo como lo son: distribución uniforme, distribución dependiente de la curvatura o distribución híbrida de las dos según la complejidad de la superficie esculpida. Las características de línea y plano se dividen en sub intervalos y los puntos de medición se distribuyen en posiciones aleatorias en el sub intervalo, es importante recalcar que los sub intervalos de números primos se consideran para una característica circular, a fin de evitar una posible distorsión

periódica de las características de medición. La planificación de la ruta de medición se realiza teniendo en cuenta la prevención de colisiones y la coincidencia de coordenadas entre las coordenadas. El resultado de la planificación es el código de máquina para una CMM específica que tenga capacidad CNC. El código de la máquina se descarga a una CMM específica y los resultados de la medición se retroalimentan a la computadora. Se desarrolla un nuevo algoritmo, llamado MINIMAXSURF, para evaluar el error de forma con precisión para superficies esculpidas. El algoritmo considera el radio de la sonda táctil y evalúa la tolerancia del perfil con éxito al eliminar los errores de configuración inevitables [12]. El CMM es esencialmente un robot cartesiano con una alta precisión, el cual está equipado con una sonda táctil. La sonda controlada por la computadora toca una secuencia de puntos en la superficie de un objeto físico que se quiere medir, y la CMM produce una secuencia de coordenadas x, y, z de los puntos de contacto. El flujo de coordenadas generado por los puntos de contacto es interpretado por algoritmos que admiten aplicaciones como ingeniería inversa, control de calidad y control de procesos [13].

2.2.7 Tolerancias GD&T

En algunas ocasiones, las tolerancias dimensionales resultan no ser lo suficientemente eficientes para comprobar si una pieza cumple o no con su propósito de diseño [14]. Con el fin de suplir y garantizar que la pieza cumpla con su finalidad, son empleadas las tolerancias geométricas, puesto que estas logran precisar mayormente una forma, posición, y orientación de la pieza de estudio, por lo tanto, es necesario establecer las tolerancias dimensionales, las tolerancias geométricas y la posición de las superficies a medir que para lograr los parámetros exigidos al producto fabricado [15].

En la actualidad existen 14 tipos de tolerancias geométricas por el número de símbolos y 15 tipos basados en la clasificación. Estos se agrupan en tolerancia de forma, tolerancia de orientación, tolerancia de ubicación y tolerancia de descentrado, los cuales se pueden usar para indicar todas las formas existentes [16].

Perpendicularidad

La cual especifica la precisión de la perpendicularidad existente de un objeto respecto a un plano de referencia, la perpendicularidad se expresa en milímetros y no en grados.

En la figura 1 se puede observar que el plano de prueba debe encontrarse dentro de dos planos paralelos perpendiculares al plano de referencia A y a una distancia de 0.03 mm [16].

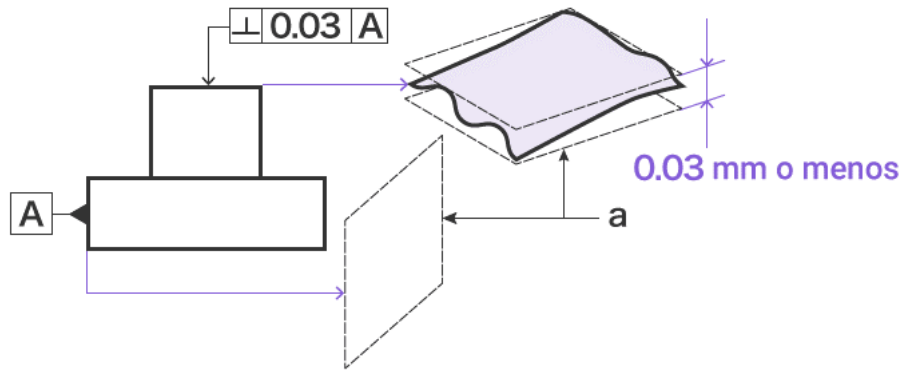


Figure 1: Perpendicularidad GD&T
Fuente:[16]

Paralelismo

Para determinar el paralelismo de la superficie de una pieza deben estar los elementos de la superficie dentro de 2 planos equidistantes por una distancia específica, los planos de restricción deben ser paralelos a un plano de referencia, por lo tanto, se puede concluir que existe paralelismo solo si la superficie está dentro del rango de tolerancias las cuales se encuentran limitadas por dos planos que son paralelos al plano de referencia, el cual debe cumplir las especificaciones de diseño.

En la figura 2 se puede contemplar que para determinar el paralelismo el plano de muestra debe ser paralelo al plano de referencia A, además debe estar entre dos planos separados por una distancia de 0.05 mm en la dirección indicada [16].

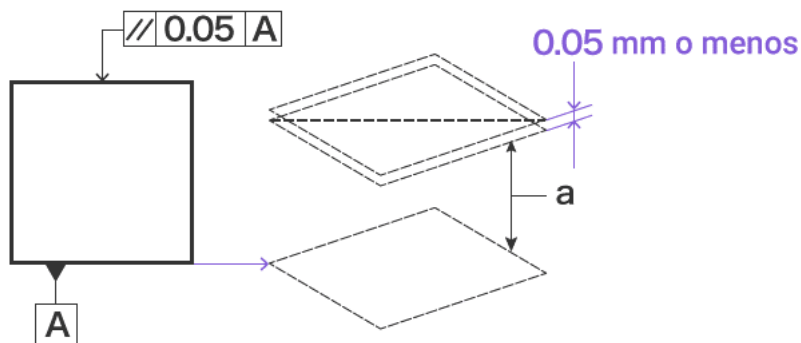


Figure 2: Paralelismo
Fuente:[16]

Circularidad

La circularidad determina la variación de la redondez de un círculo, cono u orificio. Con el fin de calcular la circularidad presente en una pieza se emplean este tipo de tolerancias para determinar si este se encuentra dentro del rango de tolerancias permitidas.

Para determinar la circularidad, la circunferencia exterior de cualquier sección transversal de un eje cortado perpendicularmente, debe estar entre dos círculos concéntricos a solo 0.1 mm de distancia en el mismo plano, como se puede evidenciar en la figura 3 [16].

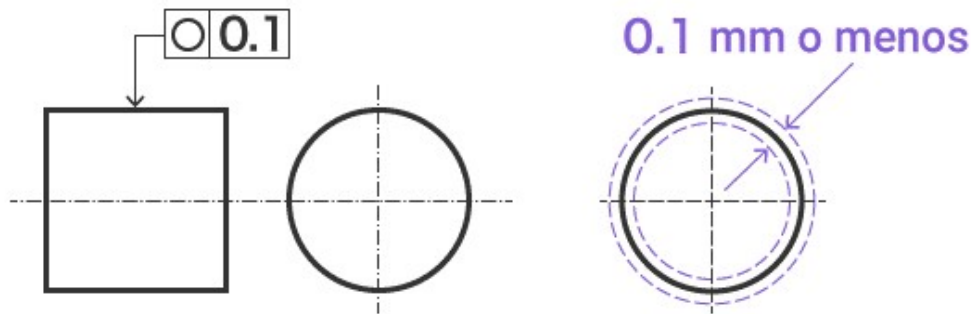


Figure 3: Circularidad

Fuente:[16]

Linealidad

La rectitud o linealidad especifican la rectitud precisa que debe tener un objeto, esta se aplica especialmente a líneas y se emplea con el fin de indicar la tolerancia de deformación existente en objetos largos.

Si un marco de tolerancia está conectado a la dimensión que indica el diámetro de un cilindro, el eje de este debe estar dentro de un cilindro con un diámetro de 0.1 mm como se evidencia en la figura 4 [16].

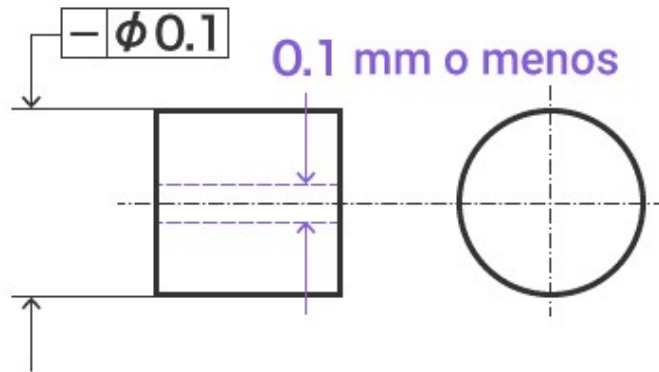


Figure 4: Linealidad
Fuente:[16]

Posición

Especifica la precisión de la posición en relación con un plano de referencia.

En la figura 5 se puede apreciar como el centro del círculo debe estar dentro de un círculo con un diámetro de 0.1 mm para poder determinar su posición [16].

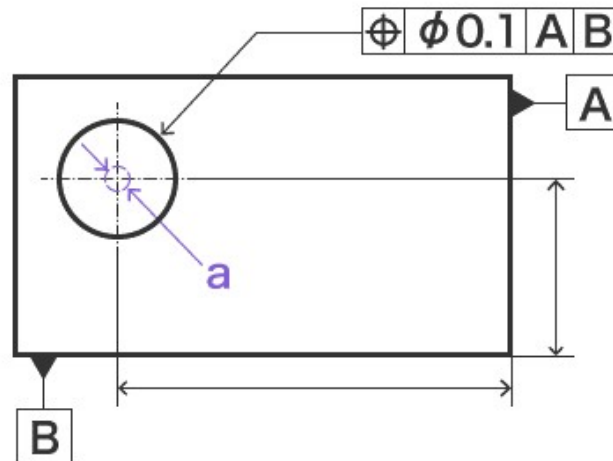


Figure 5: Posición
Fuente:[16]

2.2.8 Sistemas Embebidos

La medición digital busca integrar la visión artificial en sistemas embebidos de bajo costo con los procesos de manufactura para desarrollar sistemas portátiles y con una adaptabilidad a estos procesos.



Figure 6: Módulo ESP32-CAM
Fuente:[17]

El módulo ESP32-CAM que se puede apreciar en la figura 6 es un sistema embebido el cual tiene integrada una cámara de vídeo de 2 Mega Píxeles con una resolución de 1622 x 1200 Píxeles y una conexión para una tarjeta micro SD de hasta 4 GB, cuenta con WIFI y Bluetooth de fábrica lo cual permite una gestión de datos adentrándose en el concepto del Internet de las Cosas (IoT). Este módulo se puede programar con código abierto, su alimentación de funcionamiento es de 5V, transferencia de imagen entre 15 y 60 FPS y soporte para cámaras OV2640 y OV7670. Todos los atributos de este módulo permiten obtener como salida un vídeo en tiempo real asignándole una dirección IP para poder visualizar la captura remotamente.

En la figura 7 se puede observar los distintos componentes que se integran este sistema, permitiendo al usuario aumentar sus aplicaciones, disminuyendo costos y espacio, haciendo de este un módulo rentable.

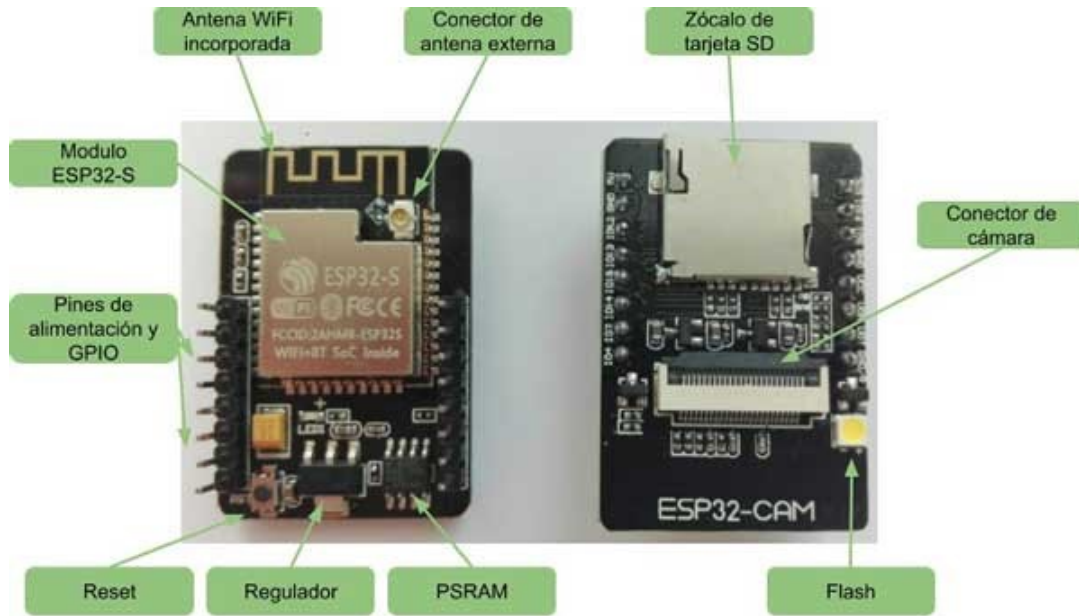


Figure 7: Componentes del módulo ESP32-CAM

Fuente:[17]

2.3 Estado del Arte

2.3.1 Investigaciones de Ámbito Nacional

Algunos trabajos e investigaciones desarrolladas en Colombia actualmente tienen como interés fortalecer los procesos de manufactura no solo a nivel nacional, sino internacional, teniendo en cuenta los aspectos más importantes de esta se han desarrollado proyectos investigativos de alto impacto.

En el artículo titulado *“Construcción de prototipo de una máquina CNC aplicando tolerancias dimensionales y geométricas”* se detalla la construcción de una máquina CNC router, la cual es empleada en el sector metal mecánico, esta máquina permite realizar un contorno en los grabados de materiales blandos. Para la construcción de esta máquina CNC se tuvieron en cuenta ajustes y tolerancias presentes en la norma ISO 286 e ISO 2786 para poder ser implementado como un módulo de enseñanza y aprendizaje de manufactura, buscando cubrir el déficit de estas máquinas de control numérico en la educación tecnológica [18].

Es necesario tener un acercamiento a las normas que rigen las tolerancias geométricas y dimensionales de las piezas como producto de los procesos de manufactura, en el proyecto investigativo titulado *“Desarrollo de un material de apoyo para la conceptualización de las tolerancias dimensionales y geométricas basados en las normas ISO-*

GPS 2011(2017), 286-1(2010), 286-2(2010) y 8015(2011) mediante el programa CAD (Solidworks)” se realiza un material de apoyo con el fin de fortalecer el acercamiento y conocimiento de las normas ISO-GPS 1101(2017), 286-1(2010), 286-2(2010), 8015(2011) que permita que el usuario pueda interpretar y aplicar dichos conocimientos dándole aporte a su formación como profesional, por medio de un software de diseño asistido por computadora (CAD) llamado SolidWorks en el cual explican de manera detallada el uso adecuado de estas normas, empleando ejemplos prácticos donde se robustece el conocimiento adquirido de forma interactiva con una serie de videos que realizan a lo largo del proyecto [19].

Las tolerancias dimensionales juegan un papel importante en la manufactura aditiva, buscando sustentar la viabilidad de realizar control dimensional y la caracterización a piezas producidas por la manufactura aditiva especialmente en ensambles mecánicos por medio del escaneo 3D, con este enfoque nace el trabajo de investigación titulado *“Caracterización de tolerancias dimensionales en ensambles mecánicos fabricados por manufactura aditiva”* donde este trabajo como objetivo principal pretende aportar en la satisfacción de la necesidad que actualmente se viene presentando en la industria para la optimización de procesos de fabricación de piezas de repuesto disminuyendo tiempo, costos y mejorando la disponibilidad, para lo cual toman como referente los modelos industriales europeos como España y Alemania donde se apuesta por desarrollar e implementar nuevas técnicas de manufactura aditiva junto con la implementación de equipos de alto desarrollo tecnológico buscando tener un acercamiento con la industria 4.0 en Colombia [20].

En el artículo de investigación titulado *“Implementación del escaneo óptico 3D como sistema de verificación en el SENA-CMM aplicado en piñones”* se muestra el análisis dimensional y geométrico de un piñón cónico donde se implementa metodología alternativa para la medición de las desviaciones presentes. Inicialmente, se realiza un escaneo óptico con el fin de adquirir una nube de puntos y luego se procesa generando geometrías experimentales contrastando con los parámetros de la geometría ideal y así poder analizar desviaciones geométricas y dimensionales presentes en el piñón cónico. Implementado este método indirecto de metrología y análisis, GD&T obtuvieron como resultado un mapeo de las desviaciones dimensionales que posee la superficie que fue maquinada, contrastada con la composición ideal del elemento en un 90% dentro del rango de las tolerancias permitidas [21].

En el artículo titulado *“Artificial vision applied to manufacturing process”* se presentan los distintos resultados obtenidos respecto a la implementación de proyectos basados en visión artificial y procesamiento de imágenes, los cuales han sido aplicados a piezas resultantes de un proceso de manufactura. Inicialmente, se obtuvieron modelos geométricos tridimensionales de las piezas de estudio capturando y procesando imágenes

termográficas. Luego se implementó un sistema de medición directo del desgaste de la herramienta de corte mediante imágenes HD en el software LabVIEW de National Instruments en una fase del proceso de fabricación de las piezas. En la fase final del proceso de fabricación de las piezas presentan el desarrollo del software diseñado, el cual con el fin de obtener un análisis metalográfico cuantitativo de micrografías para aceros hipoeutectoides [22].

2.3.2 Investigaciones de Ámbito Internacional

En la literatura podemos encontrar distintos estudios y avances importantes para la manufactura de tipo internacional como:

En el artículo titulado “*A Novel Geometric Error Compensation Method for Gantry-Moving CNC Machine Regarding Dominant Errors*” propone una estructura novedosa con pórtico móvil para mejorar la desventaja de la estructura de máquina CNC de tipo pórtico tradicional de ocupar demasiado espacio teniendo en cuenta que los errores geométricos tienen un impacto directo en la posición real de la herramienta, lo que reduce significativamente la precisión de las máquinas. Los errores de diferentes componentes siempre están acoplados y tienen efectos inciertos sobre el error geométrico total. Para realizar la compensación específica del error, inicialmente proponen un nuevo método de identificación que utiliza el análisis de sensibilidad global basado en valores (VLGSA) para encontrar los errores dominantes. En VLGSA, los factores de ponderación que muestran la influencia del rango de error se utilizan para modificar el modelo de error del sistema multicuerpo (MBS). Los resultados muestran que los errores dominantes en las tres direcciones contribuyen respectivamente con el 80%, 86% y 85% del error total en sus direcciones. Luego, los errores identificados por VLGSA se modelan mediante el ajuste lineal por mínimos cuadrados y los métodos de interpolación B-spline, respectivamente, de acuerdo con la característica de los datos de error. Finalmente, los modelos se aplican en un nuevo sistema de compensación en tiempo real desarrollado en el servosistema TwinCAT de Beckhoff. Los resultados experimentales de la máquina de fresado y grabado CNC tipo pórtico muestran que el método propuesto puede ayudar a descubrir los errores más dominantes, obteniendo como resultado la reducción aproximadamente del 90% del error total [23].

El trabajo titulado “*A generic interface and a framework designed for industrial metrology integration for the Internet of Things*” aborda la integración de dispositivos de medición en una arquitectura de IoT utilizando estándares abiertos proporcionando un marco, basado en IEC 62264 para Gestión de operaciones de calidad (QOM) e ISO 23952: 2020 - Marco de información de calidad (QIF) para describir las actividades de Garantía de calidad y Control de calidad y proporciona una interfaz genérica utilizando OPC UA para recibir y enviar información a las actividades de QOM, permitiendo la integración

con sistemas superiores como un ERP y la creación de indicadores clave de rendimiento (KPI) orientados a la calidad, por lo tanto, se proporciona un escenario experimental en la industria de fabricación de acero, que demuestra cómo la interfaz genérica puede admitir aplicaciones de software personalizadas mediante el uso de datos de metrología para respaldar, reducir los defectos de productos y procesos que conducen a la fabricación sin defectos (ZDM) [24].

El escaneo 3D es una técnica moderna utilizada para determinar la precisión dimensional de una pieza teniendo en cuenta parámetros propios de estas, en el documento titulado *“Dimensional Accuracy Evaluation of 3D – Printed Parts Using a 3D Scanning Surface Metrology Technique”* se emplea una pieza como muestra diseñada con una técnica de impresión 3D de modelado por deposición fundida y PLA como filamento principal, para realizar el escaneo se elige el escáner 3D DAVID SLS-2, ya que proporciona los datos necesarios para la comparación de las dimensiones, teniendo en cuenta la textura del extrusor, la altura de la capa y la velocidad de impresión se tienen dos niveles y tres réplicas durante la impresión. Para el diseño del experimento se utilizó un diseño factorial completo 2³ y se empleó el análisis de varianza para determinar si existe una diferencia de medias entre una variable dependiente continua y una variable independiente que comprende dos o más niveles y determinar si existe una interacción existente entre los factores. Teniendo en cuenta que los efectos de cada uno de los otros factores difieren mucho entre sí cuando se toman individualmente se rechazaron todas las hipótesis nulas producidas y se aceptaron las alternativas, lo que demuestra que no todas las medias entre los factores individuales son iguales y que existe una interacción entre los factores elegidos concluyendo que la temperatura del extrusor y la altura de la capa son los principales factores que contribuyen a la precisión dimensional [25].

MTConnect se ha convertido en un estándar de comunicación potencial para el dominio de la fabricación cibernética (CM) al establecer el monitoreo remoto de los procesos de fabricación a través de una estructura de informes de datos basada en XML y servicios RESTful. La publicación titulada *“Development and optimization of an MTConnect based edge computing node for remote monitoring in cyber manufacturing system”* presenta el desarrollo y la optimización de un nodo de computación perimetral basado en MTConnect en sistemas CM orientados a servicios mediante la utilización de almacenamiento en caché y procesamiento de datos en el perímetro. A diferencia de los agentes de MTConnect, los MTConnect Edge Nodes (MEN) propuestos no solo convierten los datos de mecanizado recopilados en mensajes XML y alojan servicios RESTful, sino que también actúan como un nodo de borde al adoptar un enfoque de “mantener hasta que se cambie” para decidir qué datos almacenar y al realizar un seguimiento de los datos transmitidos previamente a sus clientes para determinar qué datos transmitir a quién y cuándo. El objetivo principal es minimizar los requisitos y el costo de almacenamiento de datos, al tiempo que permite una transmisión rápida y un bajo uso de ancho de banda

sin aumentar la pérdida de información, obteniendo como resultado una reducción del 96,2 % en el tamaño de almacenamiento requerido con una reducción del 52,5 % en la latencia de comunicación promedio y una reducción del 99,5 % en el tamaño de mensaje promedio para MEN propuesto en diferentes escenarios de fabricación [26].

En la era de la industria 4.0 se ha planteado utilizar gemelos digitales en los procesos de medición y calibración, como se evidencia en la publicación titulada “*Digital twin-based analysis of volumetric error mapping procedures*” se presenta una metodología basada en un gemelo digital para simular todos los efectos relevantes en un procedimiento de calibración de una máquina herramienta basada en artefactos, desde la propia máquina con sus rangos de error esperados, hasta la geometría e incertidumbre del artefacto, las posiciones del artefacto en el espacio de trabajo, la incertidumbre de la sonda, modelo de compensación, etc. Al parametrizar todas las variables relevantes en el diseño del procedimiento de calibración, esta metodología de simulación puede utilizarse para analizar el efecto de cada variable de diseño en la incertidumbre del mapeo de errores, lo que es de gran ayuda para adaptar el procedimiento a cada caso específico. La metodología de simulación y las posibilidades de análisis se ilustran aplicándola en una máquina-herramienta fresadora de 3 ejes [27].

Capítulo 2

3 Arquitectura de Medición Asistida por Computadora

3.1 Introducción

Con el fin de garantizar y cumplir los criterios de diseño de los componentes industriales se emplea la inspección geométrica y dimensional, ya que esta constituye una parte integral del sistema de fabricación. Algunos datos como las dimensiones geométricas, tolerancias en la forma y la posición se pueden obtener del producto y son de alta la importancia para la evaluación, el análisis y la toma de decisiones, sin embargo, el proceso de inspección debe tener alta eficiencia y suministrar los datos suficientes para poder determinar el cumplimiento de los criterios de diseño. Con el fin de automatizar el proceso de inspección, urge el desarrollo de sistemas donde se puedan integrar el diseño e inspección asistidos por computadora.

3.2 Arquitectura de Medición Asistida por Computadora

En la figura 8 se expone un diagrama general con los componentes que conforman la arquitectura propuesta para medición asistida por computadora en procesos de manufactura. Este sistema de medición se adhiere al concepto de la industria 4.0, ya que cuenta con la integración de tecnologías digitales, por otro lado, el estándar STEP que se encuentra dentro de la norma ISO 10303 fue selecto para la estructura de datos, pues este estándar suministra los recursos, soporte y definición de productos o piezas con el fin de homogeneizar las distintas áreas existentes de la industria digital. El proceso de medición dimensional y geométrica de las piezas o producto se realiza por medio de una cámara y distintas técnicas de visión artificial para la obtención de datos de medición, generando datos de forma neutral, extensible y abierta que hacen posible el intercambio de información en los procesos de fabricación.

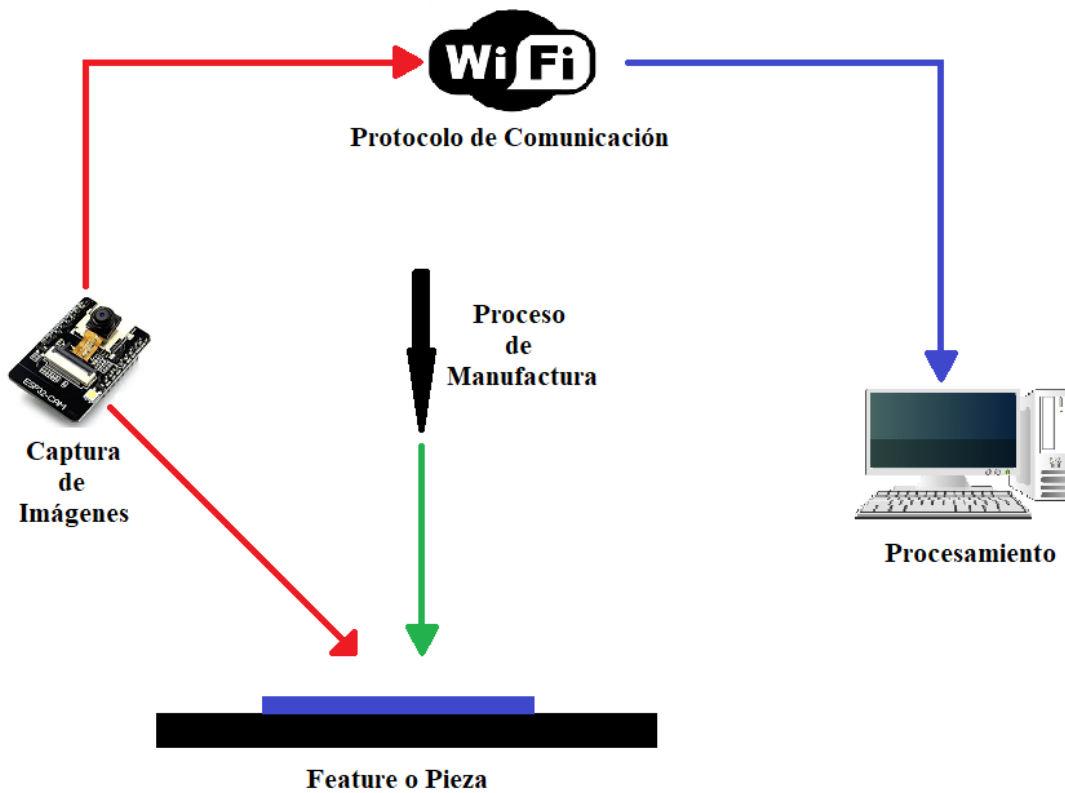


Figure 8: Arquitectura de Medición
Fuente: Autor

Esta arquitectura para medición incluye la recolección de datos e información en un formato neutro y con alta compatibilidad en procesos de manufactura con el fin de evadir transformaciones presentes provocados por incompatibilidad en los datos de los procesos de producción. La compatibilidad juega un papel importante en los procesos de manufactura para el intercambio de datos e información obtenidos con la inspección en las piezas o productos resultantes de procesos de manufactura.

El diseño del soporte del sistema de medición se desarrolla en el software SolidWorks para luego ser impresas por medio de una impresora 3D. Para el diseño de este sistema se emplean 2 software diferentes, ARDUINO que es un de código abierto para la captura de imágenes y MATLAB con el fin de procesar datos e inspeccionar la feature para la implementación de técnicas de visión artificial. Estas dos herramientas de desarrollo tienen lenguaje de programación tipo "C".

La captura de imágenes se hace por medio del módulo ESP32 CAM, el cual se encuentra en el soporte, es ajustable a distintos entornos, posee portabilidad y es inalámbrico. Para la programación de este módulo se emplea ARDUINO con el fin de asignar una dirección IP vía WIFI y almacenar las imágenes capturadas que luego son enviadas a la etapa de procesamiento.

En la etapa de procesamiento se emplea el software MATLAB donde se accede a la cámara del módulo que compone el sistema por medio de la dirección IP asignada y se diseña distintas etapas de un algoritmo para poderlas inspeccionar correctamente la feature en el siguiente orden:

1. Calibración de la cámara.
2. Implementación de técnicas de visión artificial para la detección de la zona de trabajo.
3. Detección de las esquinas de la zona de trabajo.
4. Implementación de homografía en 2D.
5. implementación de técnicas de visión artificial para la detección de la feature.
6. Inspección digital implementando visión artificial.
7. Procesamiento de los datos obtenidos y generación de reporte de inspección en archivo neutro.

Este sistema ofrece como salida un archivo neutro de la inspección de las piezas con el cual el usuario puede interactuar y comprender de una manera más sencilla y útil.

3.3 Diseño de la Feature o Pieza de Estudio

Para el diseño de la feature o pieza de estudio se empleó el software SolidWorks con el cual se realizó está como se puede observar en la figura 9 donde se presenta la vista isométrica del diseño CAD de la feature o pieza de estudio. El diseño de esta feature se basa teniendo en cuenta las cualidades que se desean inspeccionar de esta, donde se requiere como resultado el error presente de la perpendicularidad, paralelismo y circularidad en su geometría.

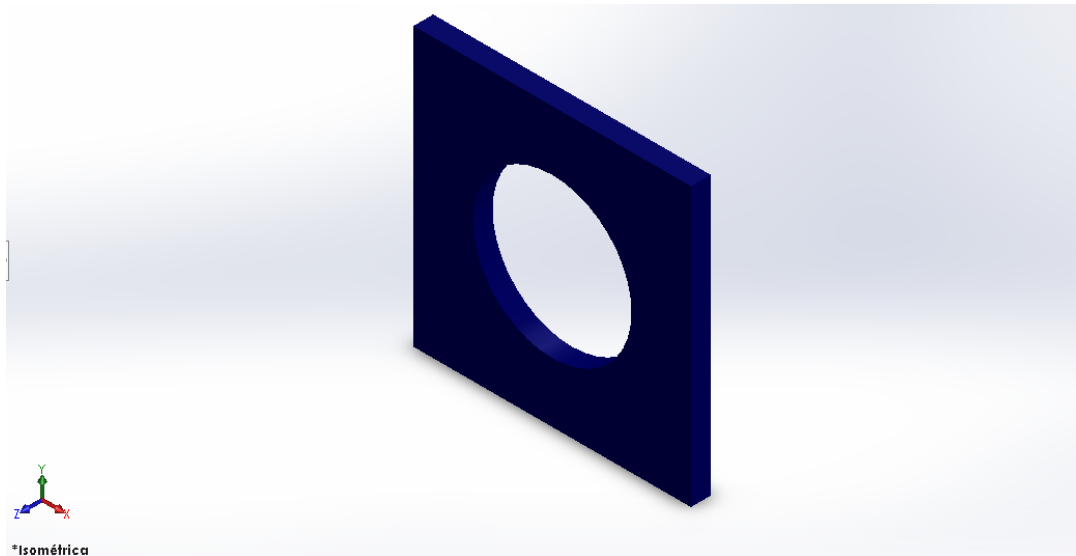


Figure 9: CAD Feature o Pieza de Estudio
Fuente: Autor

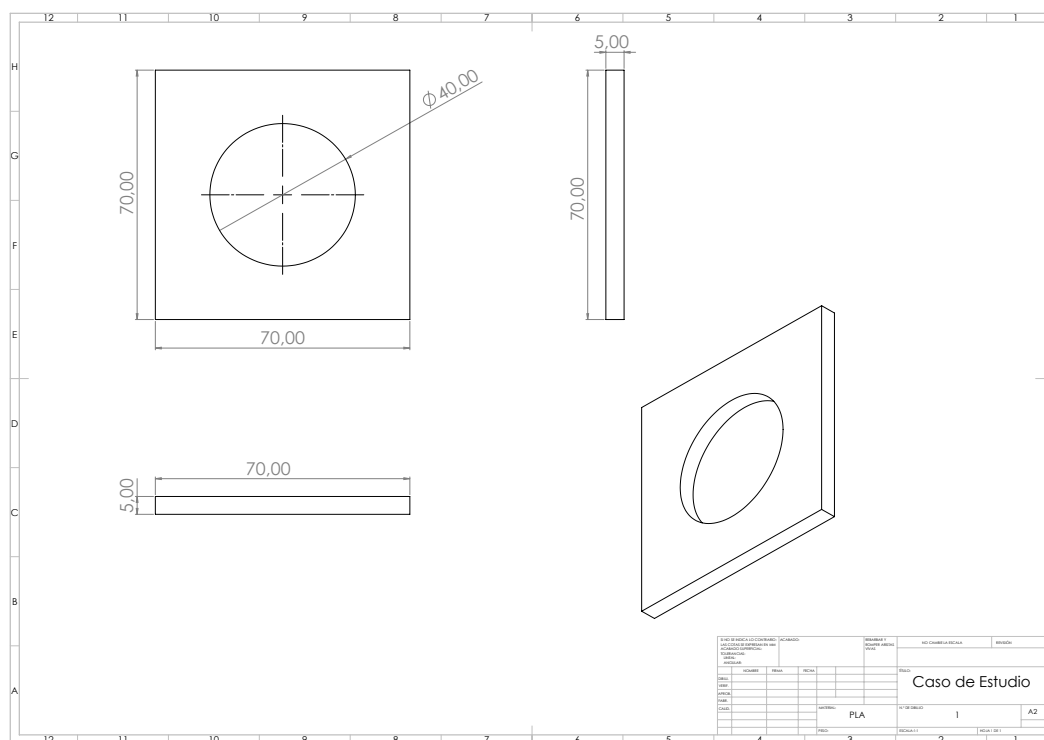


Figure 10: Plano de la Feature o Pieza de Estudio
Fuente: Autor

La figura 10 muestra el plano de la feature o pieza de estudio con sus respectivas medidas, el cual fue desarrollado en SolidWorks. Este plano cuenta con distintas vistas para tener una mejor perspectiva sobre las cotas que tienen sus geometrías.

Seguidamente de la realización del diseño CAD de la feature de estudio se convierte en formato STL y se llevará a la respectiva construcción real empleando una impresora 3D con el objetivo de realizar su inspección en un entorno real.



Figure 11: Feature o Pieza de Estudio
Fuente: Autor

La figura 11 se exhibe el resultado obtenido después de llevar a cabo el proceso de impresión 3D de la Feature de estudio previamente diseñada.

Capítulo 3

4 Construcción y Ensamble del Sistema de Inspección Digital para Medición Dimensional y Geométrica

4.1 Introducción

Los software de Diseño Asistido por Computadora (CAD) tienen como objetivo el diseño en 2D y 3D de piezas para su posterior análisis, simulación e impresión empleando impresoras 3D. Además de ser unos de los avances más significativos en la fabricación aditiva, las impresoras 3D tiene la finalidad de llevar a la realidad un objeto tridimensional, el cual es creado al superponer capas del material base de impresión de una forma sucesiva.

Las impresoras 3D cuentan con distintos materiales o filamentos para la impresión, tales como:

- Filamento PLA (Ácido Poliláctico).
- Filamento ABS (Acrilonitrilo Butadieno Estireno).
- Filamento PETG (Tereftalato de Polietileno Glicolizado).
- Náilon.
- TPU (Poliuretano Termoplástico).
- PVA (Alcohol Polivinílico).
- HIPS (Poliestireno de Alto impacto).
- Compuestos (Fibra de Carbono, Kevlar, Fibra de Vidrio).
- Resinas.

La facilidad de interacción máquina y usuario, la eficiencia y el costo son unos de los principales factores que hacen que las impresoras 3D sea la mejor opción a la hora de la fabricación por adición. Teniendo en cuenta que la calidad del producto está definida por la precisión de la máquina, las máquinas de bajo costo nos dan como resultado una baja precisión comparada a las de alto costo.

Este capítulo detalla la construcción y el ensamble del sistema de inspección digital para medición dimensional y geométrica. Presenta el diseño de piezas 3D, las cuales integran el sistema de inspección digital tanto en software como los resultados de impresión. La elección y conexión de los componentes electrónicos de este sistema y su diseño en software CAD. Finalmente, el ensamble de todos los componentes del sistema de inspección digital para medición dimensional y geométrica en software CAD como en Físico.

4.2 Estructura del Soporte Para el Sistema De Inspección

Para el diseño de la estructura del soporte sistema de inspección digital se diseñaron distintas piezas en el software llamado “SolidWorks” con el objetivo de diseñar con medidas reales el tamaño ideal para el sistema, para luego imprimir estas piezas en 3D y así tener como resultado la estructura final para el sistema.

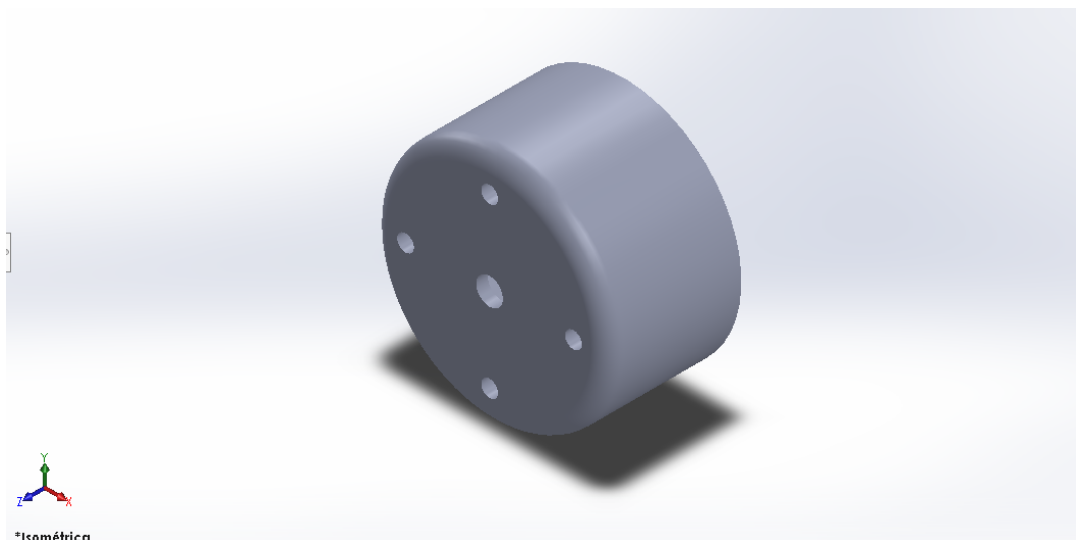


Figure 12: CAD Soporte Sistema de Medición

Fuente: Autor

En la figura 12 se presenta el diseño en el software CAD llamado “SolidWorks” del soporte para la estructura del sistema de inspección digital para medición dimensional y geométrica, teniendo en cuenta las dimensiones y peso de todos los componentes que lo conforman. El objetivo principal de esta pieza es contener los distintos componentes que integran el sistema.

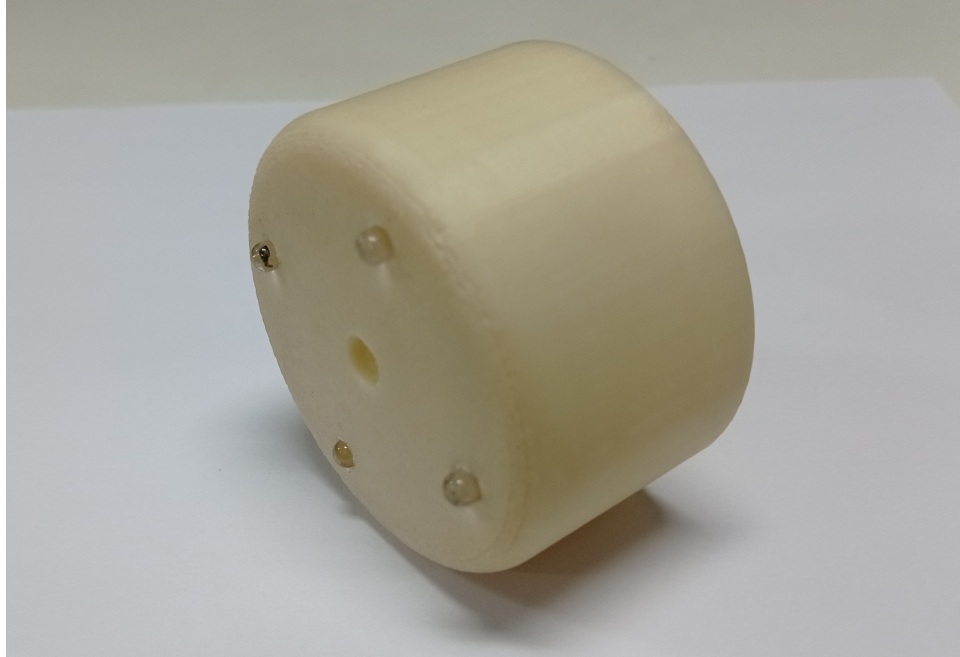


Figure 13: Soporte Sistema de Medición
Fuente: Autor

Como se puede observar en la figura 13 se tiene como resultado la pieza impresa anteriormente diseñada en el software CAD del sistema de inspección digital, esta pieza una vez diseñada paso por un proceso de conversión a código “G” para su posterior impresión en la máquina de impresión 3D.

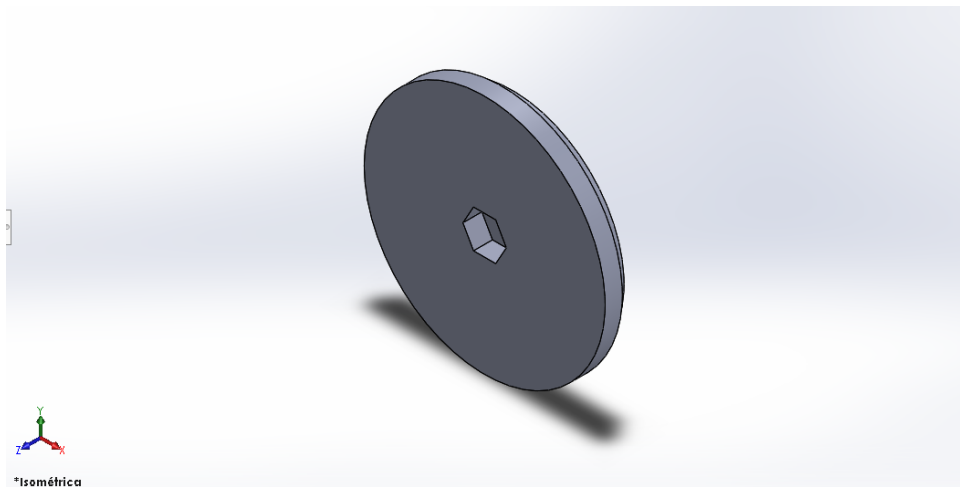


Figure 14: CAD Tapa Soporte Sistema de Medición
Fuente: Autor

En la figura 14 se puede visualizar el diseño en el software CAD llamado “Solid-Works” de la tapa del soporte para la estructura del sistema de inspección digital para medición dimensional y geométrica, teniendo en cuenta las dimensiones del soporte y de la herramienta de sujeción del sistema.



Figure 15: Tapa Soporte Sistema de Medición
Fuente: Autor

En la figura 15 se presenta el resultado de impresión de la tapa del soporte del sistema de inspección digital diseñado, esta pieza una vez diseñada paso por un proceso de conversión a código “G” para su posterior impresión. La finalidad principal de esta pieza es el encaje con el soporte para proteger los componentes de posibles accidentes y acople con herramientas de sujeción en distintos procesos de manufactura.

El tipo de material seleccionado para la impresión de estas piezas fue el ABS, ya que este presenta un mejor acabado, poca flexibilidad y es extremadamente resistente, haciendo este material ideal para la impresión del sistema porque generalmente se emplea para aplicaciones industriales, además de una impresión tipo sólida del 100% haciendo de este sistema muy resistente.

4.3 Diseño CAD de Componentes Electrónicos y ensamble de elementos que Conforman el Sistema

Empleando el software “SolidWorks” se diseñaron las piezas electrónicas a detalle y con sus respectivas medidas de cada uno de los componentes que integran el sistema de inspección digital. El diseño CAD de estos componentes fue de gran utilidad para la creación estructural geométrica del soporte del sistema de una forma óptima y adecuada, realizando un ensamble de todas ellas.

Además del diseño CAD de los componentes electrónicos que conforman el sistema de inspección digital, se emplearon estos para realizar el ensamble de estos y así tener una perspectiva final del sistema planteado.

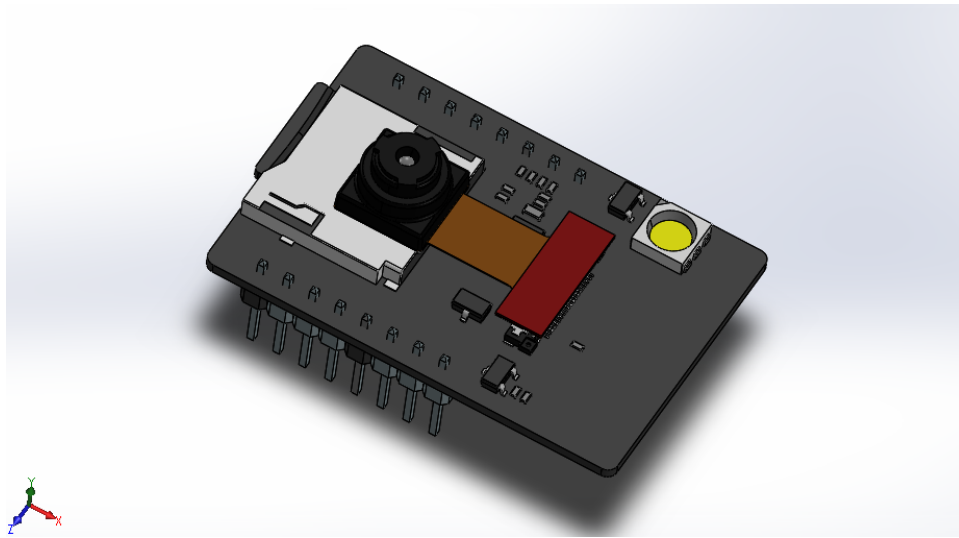


Figure 16: CAD ESP32 CAM

Fuente: Autor

En la figura 16 se muestra el diseño en el software CAD del módulo ESP32 CAM el cual fue empleado para realizar el diseño del soporte donde se aloja el sistema de medición, para este diseño las medidas son fundamentales, ya que están estandarizadas por el creador del módulo, además que esta cuenta con sus componentes como la cámara y tarjeta micro SD diseñadas. Este Diseño CAD cuenta con todos los componentes detallados minuciosamente para buscar realizar la pieza, soporte con mayor precisión, optimización de materiales y espacio.

Se diseña el CAD de la batería de alimentación con la finalidad de elegir una batería que supla el voltaje de operación del sistema de inspección digital y además este dentro del rango de las medidas del sistema.

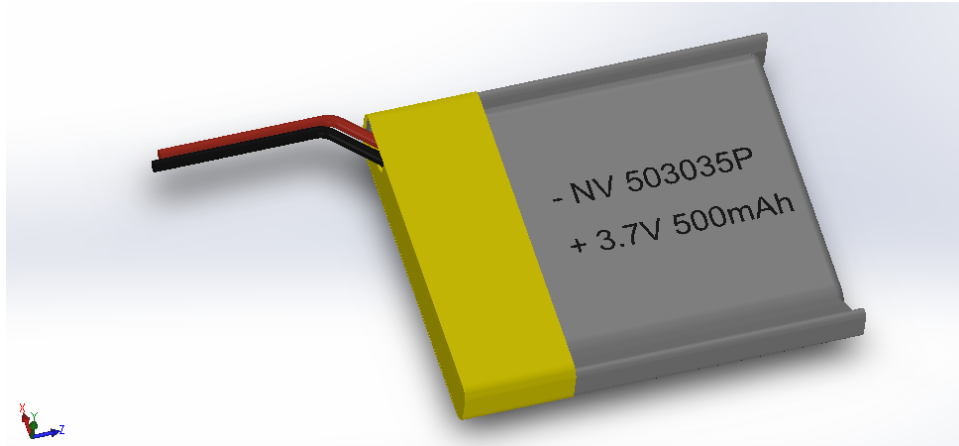


Figure 17: CAD Batería LiPo

Fuente: Autor

La figura 17 ilustra el diseño CAD de La batería LiPo diseñada teniendo en cuenta sus especificaciones respecto a sus dimensiones. Esta batería se encuentra en el mercado y suplente la alimentación del sistema, por lo cual su implementación la hace favorable, ya que es de bajo costo, fácil acceso y uso.

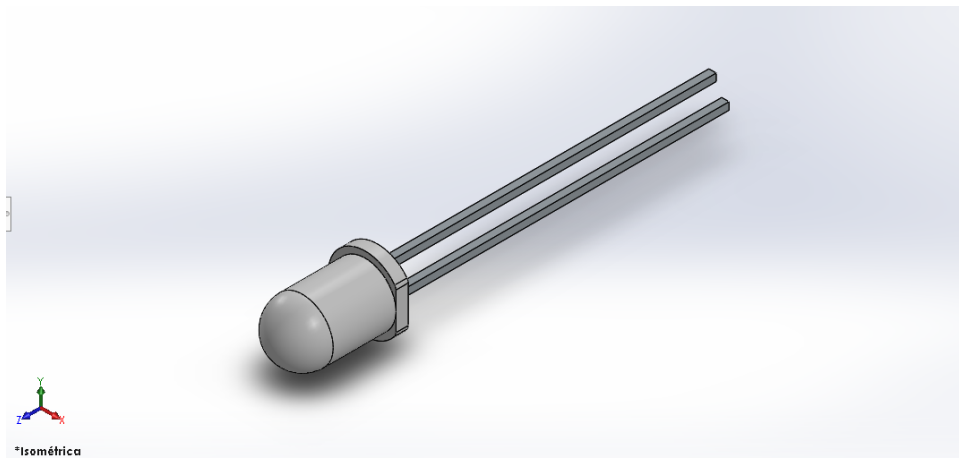


Figure 18: CAD Diodo LED

Fuente: Autor

La figura 18 muestra el diseño CAD de los diodos LED de 5 mm que integran el sistema. Estos diodos LED son de suma importancia ya que suplen la necesidad de iluminar la feature a inspeccionar con el objetivo de eliminar sombras presentes que pueden alterar la inspección de la feature, además posibilita su operación en ausencia de luminosidad externa.

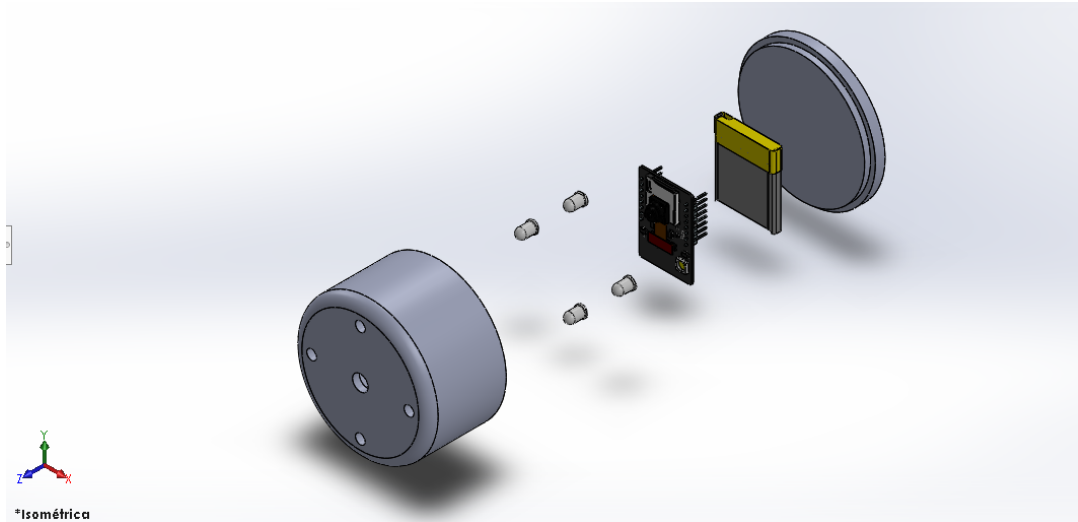


Figure 19: CAD Sistema de Inspección Digital

Fuente: Autor

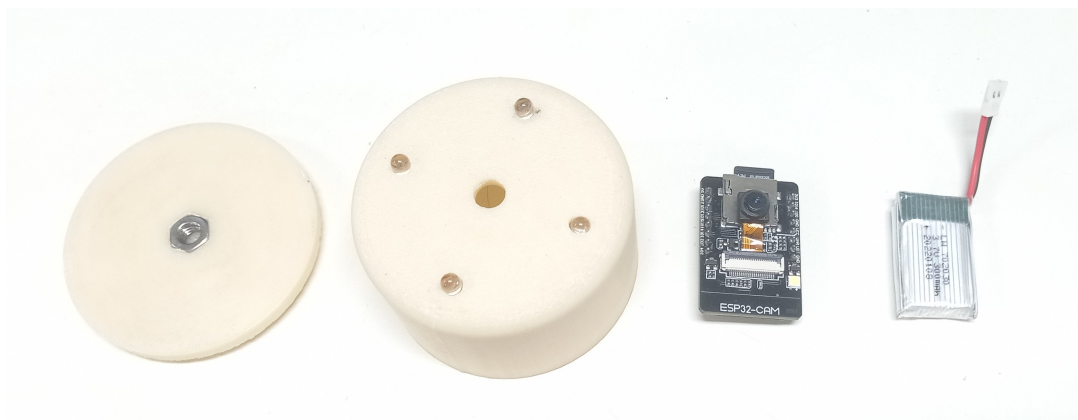


Figure 20: Sistema de Inspección Digital

Fuente: Autor

En la figura 19 se puede apreciar el ensamble de todas las piezas previamente diseñadas conformando el sistema para la toma de datos de inspección digital. Una vez realizada la simulación del CAD del sistema se presenta en la figura 20 el sistema con todos sus elementos que lo componen, donde se puede observar el módulo ESP32 CAM, la batería LiPo de alimentación del sistema, Diodos LED, el soporte y la tapa impresos en 3D.

4.4 Conexiones del Sistema de Inspección Digital

El sistema de inspección digital tiene distintas fases en la conexión de sus componentes con el fin de obtener como resultado un sistema óptimo, robusto y eficaz, por esta razón la alimentación y configuraciones de los componentes debe ser de estricto cuidado y teniendo en cuenta la hoja de características de cada elemento. La vida útil de cada componente que comprende el sistema está ligado a su correcto uso, alimentación y protección.

La correcta conexión de un circuito electrónico es fundamental, pues este se ve seriamente afectado de forma positiva o negativa, además de poder comprometer su correcto funcionamiento a la hora ejecutar una tarea específica.

4.4.1 Conexiones para la Adquisición de Imágenes

Para cargar el algoritmo de programación al módulo ESP32 CAM desde la tarjeta ARDUINO UNO se debe tener en cuenta las distintas configuraciones de carga de algoritmo y de prueba de funcionamiento.

Fase de Carga

En Esta fase se carga el algoritmo al módulo ESP32-CAM por medio de la tarjeta ARDUINO para lo cual se debe tener en cuenta la conexión que se presenta en la figura 21.

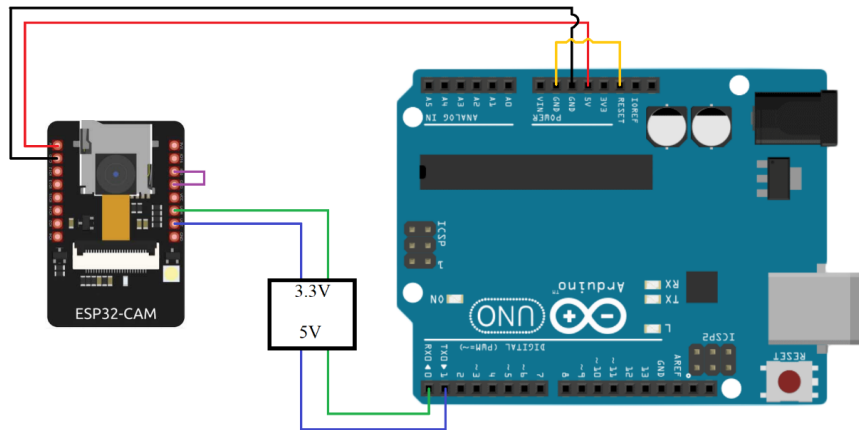


Figure 21: Conexión ESP32 CAM y ARDUINO Fase de Carga
Fuente: Autor

La figura 21 ilustra la correcta conexión para carga del algoritmo entre el módulo ESP32 CAM y la tarjeta embebida ARDUINO UNO donde se puede apreciar la conexión entre el GND y EL RESET de esta, ya que con esta configuración se omite el microcontrolador ATMEGA 328P para poder cargar el código directamente desde la tarjeta. La conexión entre los pines de comunicación respectivamente no se puede hacer de forma directa, por lo cual se debe aplicar un divisor de voltaje, puesto que la Arduino cuenta con 5V de salida y los pines del módulo ESP32 CAM solo tolera 3,3V de entrada.

Fase de Prueba

Para esta configuración, además de todas las conexiones anteriormente explicadas, debe existir una conexión entre los pines GPIO0 y GND, así se garantiza que la tarjeta queda lista para la fase de carga del algoritmo de programación.

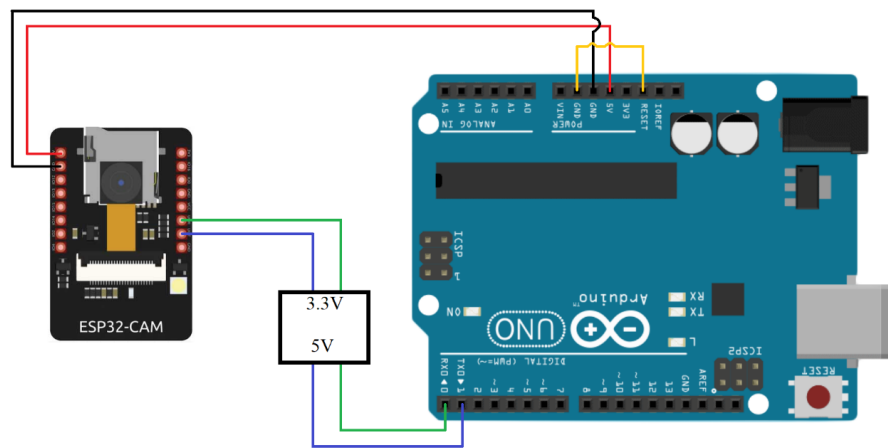


Figure 22: Conexión ESP32 CAM y ARDUINO Fase de Prueba

Fuente: Autor

La figura 22 se presenta la conexión entre el módulo ESP32 CAM y la tarjeta embebida ARDUINO UNO, en esta fase de prueba se tiene la misma conexión que la fase de carga. Esta conexión se distingue de la fase de carga por la desconexión entre los pines GPIO0 y GND, lo cual habilita para que se puedan hacer las respectivas pruebas con el algoritmo previamente cargado con el fin de comprobar la carga de este correctamente.

En la siguiente tabla se presenta la conexión de los pines en las distintas configuraciones para carga y prueba del algoritmo:

COLOR CABLE	CARGA	PRUEBA
Rojo	5V (ARDUINO) – 5V (ESP32 CAM)	5V (ARDUINO) – 5V (ESP32 CAM)
Negro	GND (ARDUINO) – GND (ESP32 CAM)	GND (ARDUINO) – GND (ESP32 CAM)
Naranja	GND (ARDUINO) – RESET (ARDUINO)	GND (ARDUINO) – RESET (ARDUINO)
Verde	RX0 (ARDUINO) – GPIO3 (ESP32 CAM)	RX0 (ARDUINO) – GPIO3 (ESP32 CAM)
Azul	TX0 (ARDUINO) – GPIO1 (ESP32 CAM)	TX0 (ARDUINO) – GPIO1 (ESP32 CAM)
Morado	GND (ESP32 CAM) – GPIO0 (ESP32 CAM)	

Table 1: Cableado de Fases de Carga y Prueba

Fuente: Autor

4.4.2 Conexión final del sistema de inspección digital



Figure 23: Conexión Sistema

Fuente: Autor

El módulo ESP32 CAM recibe alimentación de 3.3 V en adelante el cual como se puede observar en la figura 23 se está alimentando con una batería LiPo de 3.7 V a 300 mAh supliendo así su voltaje de operación, una vez obtenida esta conexión el sistema ya puede capturar las imágenes en tiempo real, luego las envía a una dirección IP que se le asignó por medio de un protocolo de comunicación TCP/IP vía WIFI en la fase de programación para así poder acceder a la data de imágenes capturadas y poder procesarlas desde el algoritmo desarrollado con técnicas de visión artificial desde un ordenador el cual está encargado de inspeccionar la pieza dimensional y geoméricamente.

Capítulo 4

5 Desarrollo del Sistema de Visión Artificial para la Inspección Digital en la Medición Dimensional y Geométrica

5.1 Introducción

Este capítulo presenta detalladamente la construcción de los algoritmos de programación para el sistema de inspección digital en la medición dimensional y geométrica para el cual se emplean distintas técnicas de procesamiento de imágenes y programación. Inicialmente se programa la fase de adquisición de las imágenes. Seguidamente se hace una adquisición de las imágenes desde la dirección IP asignada con el protocolo de comunicación. Se desarrolla un código con calibración de la cámara y finalmente el código con distintas técnicas de visión artificial y tratamiento digital de imágenes para tener los resultados esperados.

El principal objetivo de la visión artificial en los procesos industriales es automatizar los procesos, buscando aumentar la eficiencia y disminuyendo los errores presentes a la hora de producción impactando positivamente en la calidad del producto. La visión artificial es aplicable a múltiples tareas dependiendo su finalidad, los usos más comunes para los cuales empleada son:

- Identificación.
- Clasificación.
- Rastreo
- Detección de objetos.
- Inspección visual de objetos.
- Colocación de features o piezas.
- medición de features o piezas.

En la era digital la visión artificial ha sido una de las tecnologías fundamentales ya que es aplicable a la industria en múltiples ámbitos optimizando material y tiempo de producción mejorando la calidad del producto y sentando las bases de la industria 4.0.

5.2 Algoritmo para la Adquisición de Imágenes

La primera fase del este sistema es la adquisición de imágenes para lo cual se ha desarrollado un algoritmo basado en lenguaje “C” en el entorno de programación libre “ARDUINO” para lo cual se debe tener en cuenta las conexiones del sistema que se presentan en la figura 21 para poder ser cargado correctamente este código de programación y la figura 22 para la respectiva prueba.

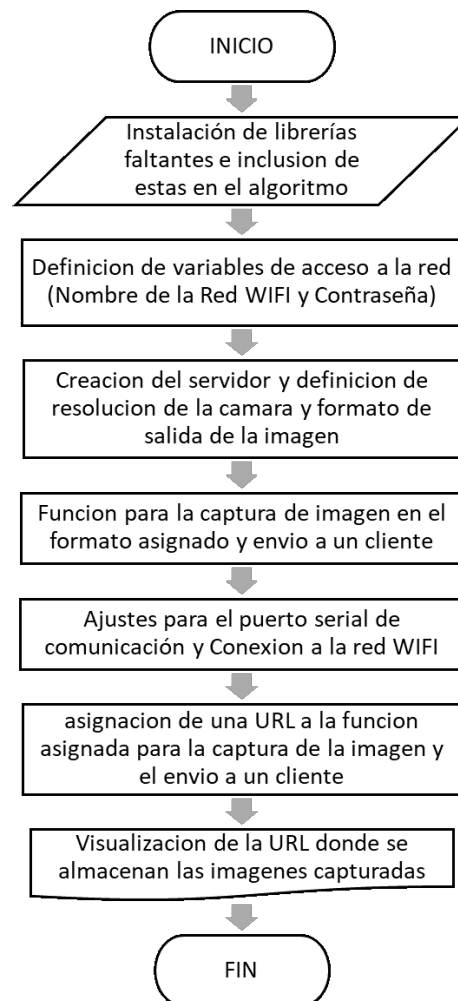


Figure 24: Diagrama de Flujo Algoritmo para Adquisición de Imágenes

Fuente: Autor

En la figura 24 se presenta el algoritmo diseñado para la adquisición de imágenes, este algoritmo cuenta con distintas fases para su diseño, el objetivo principal de este algoritmo es adquirir las imágenes y enviarlas a un servidor por medio del protocolo de comunicación TCP/IP vía WIFI. Esto se desarrolla en el módulo ESP32 CAM y la Tarjeta Arduino.

La instalación de librerías faltantes es de mucha importancia ya que el software “AR-DUINO” no cuenta con estas, por lo tanto, se deben instalar además de configurarse para poder ser utilizadas. se debe verificar que el paquete de la tarjeta ESP32 CAM este instalado.

Inicialmente se elige un IDLE en blanco en el software Arduino para el desarrollo del algoritmo. la inclusión de las librerías previamente instaladas es de alta importancia para este. En la primera fase del proceso de programación se deben definir las variables de acceso a la red WIFI dentro de nuestro campo de trabajo la cual la red debe tener internet para poder asignar una dirección IP. La segunda fase se encarga está compuesta por la creación del servidor y de la definición de la resolución de la cámara integrada al módulo ESP32 CAM y el formato que se le quiere dar a la imagen. como tercera fase se crea una función en la cual se acondiciona el funcionamiento para llevar a cabo la captura de la imagen en el formato anteriormente asignado, además, de establecer el envío de las imágenes capturadas al cliente. La cuarta fase resalta el tener en cuenta los ajustes para el puerto de comunicación serial son muy importantes para establecer una comunicación entre el módulo y la tarjeta por lo tanto se deben configurar a la velocidad adecuada, la conexión a la red WIFI nos proporciona una dirección IP para el envío de la data al servidor. Como fase final se realiza una asignación de una URL a la función previamente creada para la captura de imágenes y el envío a un al cliente para así tener como resultado final la visualización en el Monitor Serie del Software de la URL con la dirección IP ya asignada donde se almacenan las imágenes capturadas con la resolución y el formato previamente establecido.

Una vez finalizado el proceso de programación y contando con los elementos en físico con su correcta conexión se procede a realizar la carga del algoritmo al modulo ESP32 CAM teniendo en cuenta las fases de carga y prueba de funcionamiento.

5.2.1 Fase de Carga Del Algoritmo

Para la fase de carga del algoritmo para la adquisición de imágenes inicialmente se debe tener en cuenta la conexión de la figura 21 la cual se describe detalladamente en el capítulo anterior. Esta fase es la encargada de realizarse el cargue de forma correcta del algoritmo previamente diseñado al modulo ESP32 CAM para someterlo a la fase de prueba.

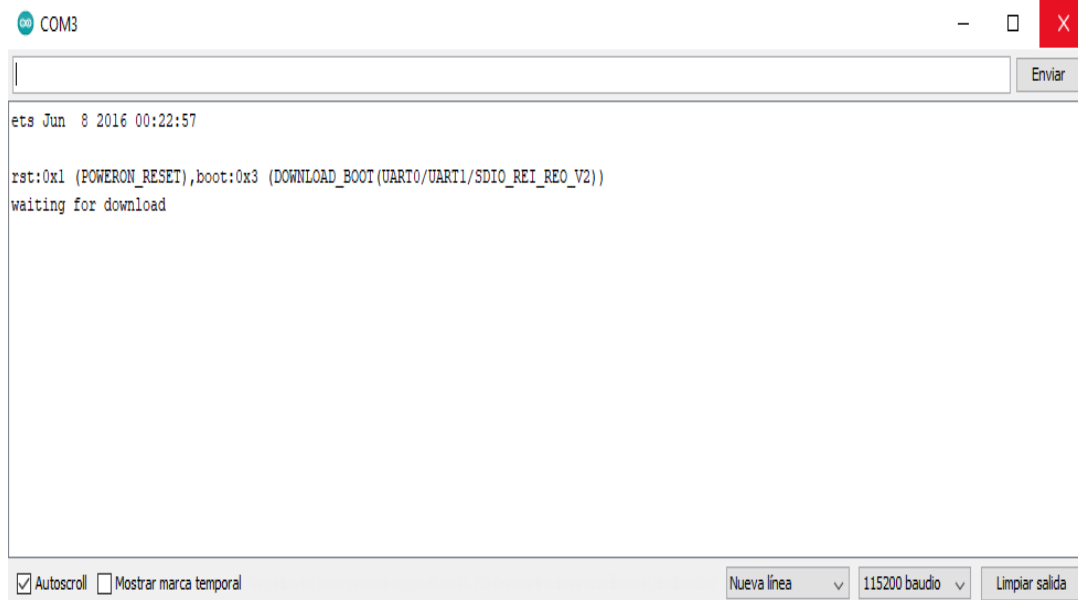


Figure 25: Modo Flash del ESP32 CAM

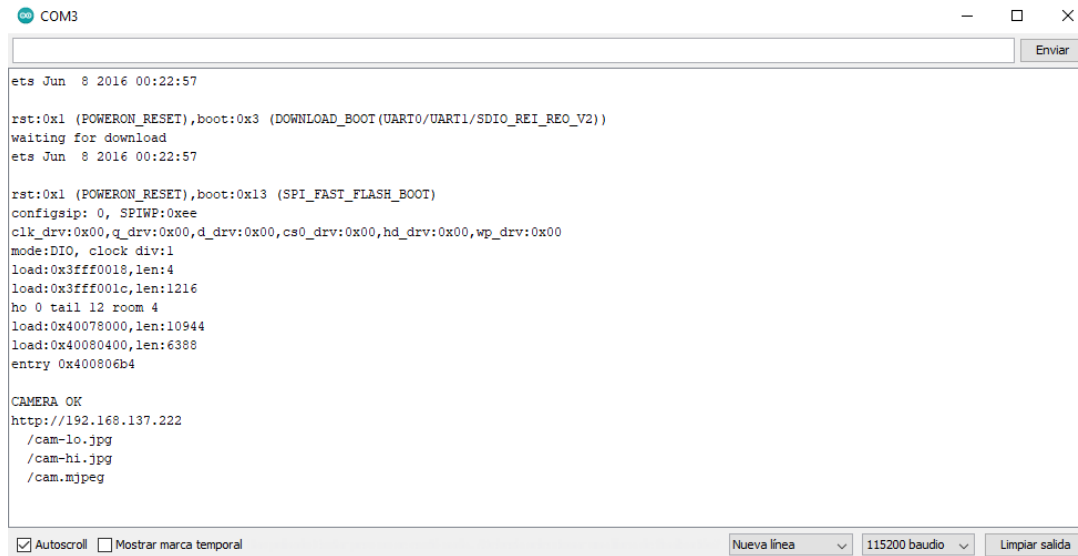
Fuente: Autor

Teniendo en cuenta las conexiones de la figura 21 y todos los parámetros de suma importancia, se realiza la conexión al computador desde el cual se va a compilar el algoritmo, seguido se abre el monitor serial del Arduino, se presiona el botón RESET del módulo ESP32 CAM y se obtiene como resultado en el monitor serial un mensaje indicando que se encuentra en modo Flash como se puede evidenciar en la figura 25, por lo tanto, el módulo está correctamente conectado y listo para la carga del algoritmo. Cumpliendo con estos parámetros, finalmente se realiza la carga del algoritmo de programación al módulo.

5.2.2 Fase de Prueba del Algoritmo

Realizada la carga del algoritmo al módulo ESP32 CAM se realizan la fase de prueba con el fin de garantizar la correcta carga del algoritmo. En esta fase del algoritmo para la adquisición de imágenes inicialmente se tiene en cuenta la conexión de la figura 22, ya que esta describe detalladamente las conexiones correctas para la ejecución de esta fase.

Teniendo la conexión correcta del módulo ESP32 CAM para esta fase se procede a presionar el botón RESET con el objetivo de probar si la carga del algoritmo fue exitosa, si el cargue fue exitoso, el módulo indicara esto como se presenta en la figura 26.



```
ets Jun 8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x3 (DOWNLOAD_BOOT(UART0/UART1/SDIO_REI_REQ_V2))
waiting for download
ets Jun 8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:10944
load:0x40080400,len:6388
entry 0x400806b4

CAMERA OK
http://192.168.137.222
/cam-lo.jpg
/cam-hi.jpg
/cam.mjpeg
```

Figure 26: Asignación Dirección IP del ESP32 CAM

Fuente: Autor

La figura 26 presenta gráficamente el mensaje adquirido desde el monitor serial del software “ARDUINO” en el que se obtiene como resultado la dirección IP asignada a nuestro módulo ESP32 CAM donde se almacenan las imágenes obtenidas en tiempo real con el formato de imagen previamente establecido.

Con la finalización de esta fase se tiene que la carga del algoritmo fue de manera satisfactoria y así mismo ya está lista la dirección IP para la extracción de las imágenes desde el cliente.

5.3 Algoritmo para la Inspección Digital, Dimensional y Geométrica Empleando Técnicas de Visión Artificial

Los procesos de inspección dimensional y geométrica buscan una mayor precisión y confiabilidad en la inspección de las tolerancias en una pieza producto de un proceso de manufactura. La calidad de las imágenes juega un papel importante a la hora del procesamiento digital de estas y aún más en los procesos asociados a visión artificial. La medición e inspección es un proceso que requiere de rigurosidad y precisión, por lo tanto, se ha desarrollado este algoritmo teniendo en cuenta la extracción de imágenes, calibración de la cámara y distintas técnicas de visión artificial de una forma minuciosa, teniendo en cuenta los parámetros que afectan este proceso con el fin de reducir la operabilidad y el error humano.

5.3.1 Obtención de Imágenes con MATLAB desde Dirección IP

Inicialmente, se extrae la imagen que está almacenada en una dirección IP anteriormente creada, para lo cual se emplea la función del software “MATLAB” llamada “ipcam” para poder acceder a esta dirección y poder capturar las imágenes desde el sistema creado. La dirección que vamos a emplear es la creada con el formato “mjpeg”, ya que con esta podemos acceder a las capturas en tiempo real, mientras que las otras 2 direcciones es para capturar solo una imagen y la función “ipcam” no acepta estos tipos de formatos.

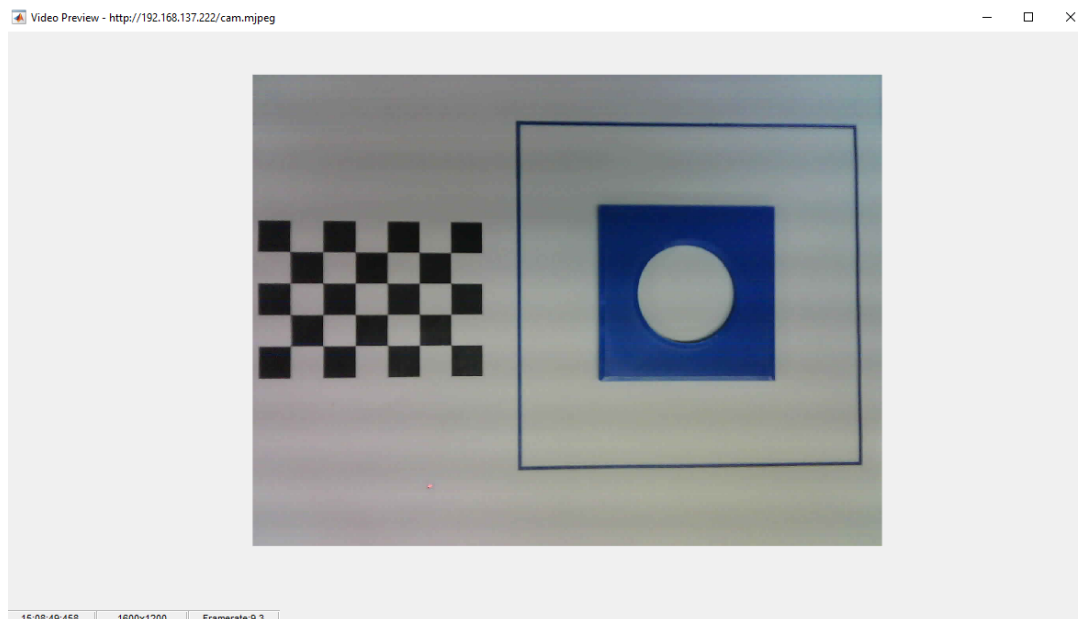


Figure 27: Imagen Obtenida con MATLAB desde la Dirección IP

Fuente: Autor

En la figura 27 se presenta el acceso a esta dirección directamente desde el software MATLAB accediendo a las imágenes capturadas desde el sistema de captura de imágenes y almacenadas en la dirección IP asignada.

5.3.2 Calibración de la Cámara del Sistema

Una vez exista acceso desde el software a la dirección IP se debe realizar una calibración de la cámara para así obtener parámetros de las imágenes capturadas y darles mejoría a estas.

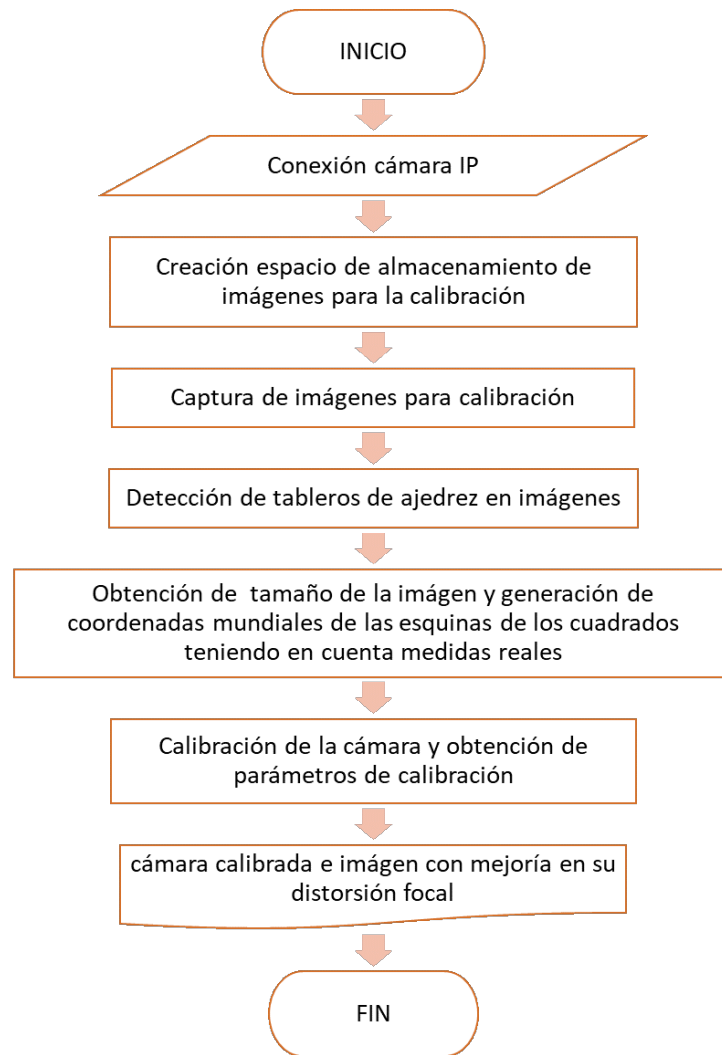


Figure 28: Diagrama de Flujo Calibración Cámara
Fuente: Autor

En la figura 28 se detalla el proceso para la calibración de la cámara de nuestro sistema, inicialmente se parte desde la conexión de la cámara con la dirección IP directamente desde MATLAB. Seguido es de suma importancia crear un espacio donde se almacenarán las imágenes capturadas para la calibración. Una vez capturadas las imágenes se procede a realizar una detección de tablero de ajedrez para así poder obtener el tamaño de la imagen y generar las coordenadas mundiales de las esquinas de los cuadros de este, teniendo en cuenta las medidas reales existentes entre cada cuadro del tablero. Con estos parámetros se hace la calibración de la cámara y se obtienen los parámetros de calibración por código. Finalmente, se obtiene como resultado la cámara del sistema de adquisición de imágenes calibrada y la imagen de la pieza que se va a inspección con técnicas de visión artificial con mejoría en su distorsión focal como se muestra en la figura 29.

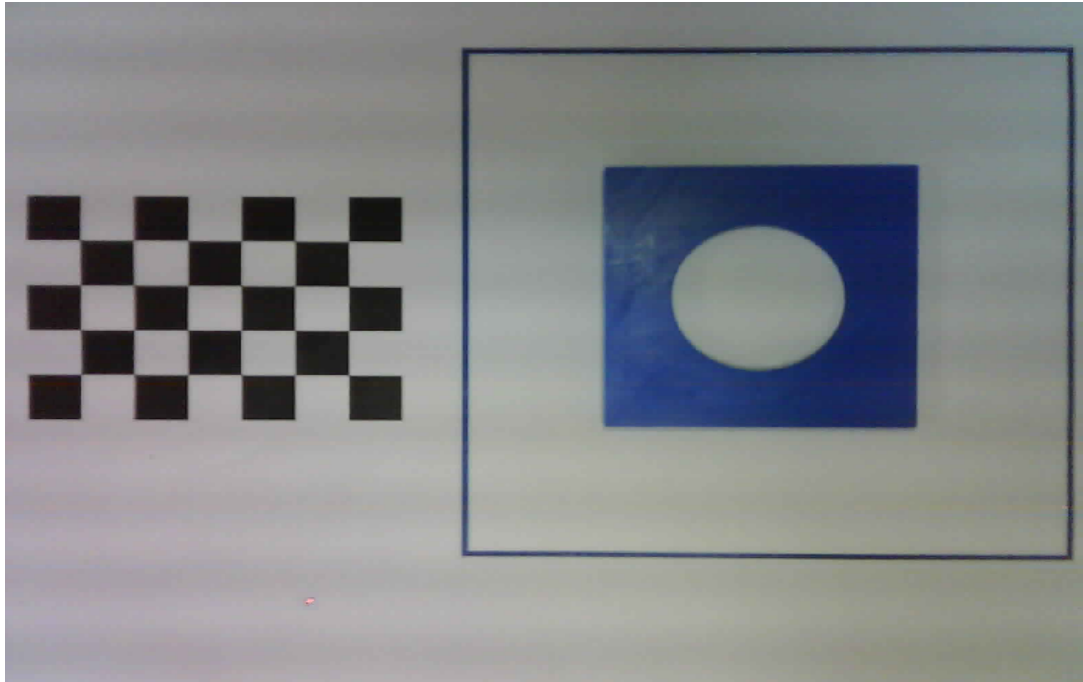


Figure 29: Imagen Patrón con mejoría en Distorsión Focal

Fuente: Autor

La calibración de la cámara y obtención de parámetros de la calibración es fundamental para mejorar la distorsión focal, esto nos ayuda a que haya una mejor focalización en la imagen, aumentando la efectividad en la inspección digital y disminuyendo errores por distorsión los cuales afectan significativamente la precisión de la inspección de la feature.

5.3.3 Implementación de Técnicas de Visión Artificial para la Inspección Digital

La visión artificial nace con el objetivo de facilitar procesos que conllevan de intervención, análisis e interacción humana. Esta cumple tareas como adquirir, procesar, analizar, comprender e identificar el mundo real por medio de imágenes gráficas y así procesar esta información de forma numérica por medio de un ordenador.

Calibrada la cámara y teniendo como resultado la imagen patrón, se desarrolla un algoritmo con técnicas de visión artificial para la imagen obtenida con el objetivo de realizar la inspección digital de esta de acorde a como se detalla en el diagrama de flujo de la figura 30.

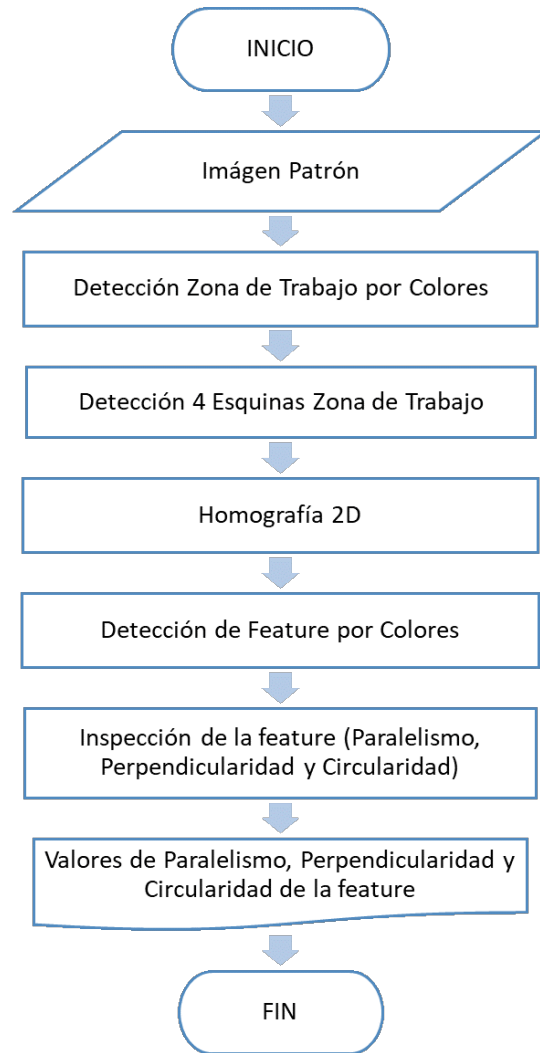


Figure 30: Diagrama de Flujo Estructura Algoritmo Inspección Digital
Fuente: Autor

5.3.4 Detección de la Zona de Trabajo

Inicialmente, se realiza la detección de la zona de trabajo en la cual dentro de esta se encuentra la feature a inspeccionar, para esto se hallan los datos de los píxeles correspondientes al color azul, ya que este compone la zona de trabajo para lo cual se realiza un recorte a una porción de esta como se muestra en la figura 31.

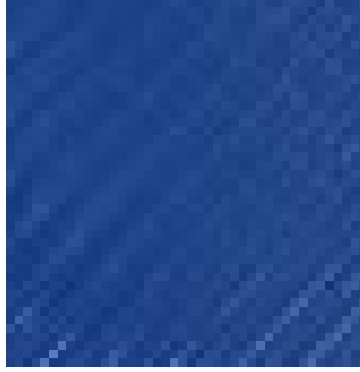


Figure 31: Porción de la Zona de Trabajo con Respectivo Color
Fuente: Autor

Seguido a esto se hallan los valores correspondientes a los píxeles de la porción extraída de la zona de trabajo realizando un histograma de colores para determinar los componentes en las matrices de colores en el espacio de color RGB.

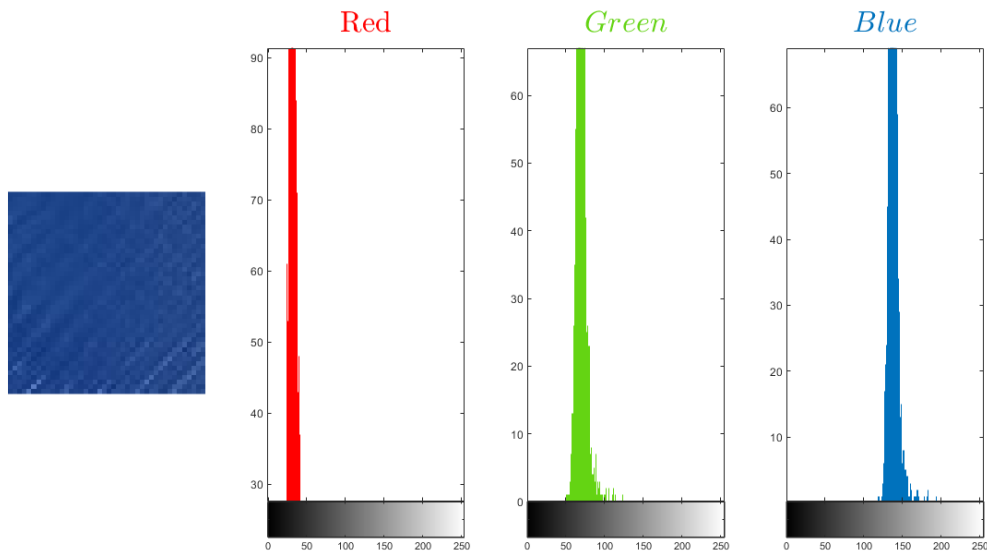


Figure 32: Porción de la Zona de Trabajo con Respectivo Color
Fuente: Autor

En la figura 32 se puede apreciar el histograma de colores del color de la porción de la zona de trabajo, donde se tiene 31 para el componente rojo, 70 para el componente verde y 138 para el componente azul. Los valores obtenidos del histograma de colores se utilizan para encontrar los píxeles similares al color de la zona de trabajo, para lo cual se emplea la ecuación 1 para calcular el error cuadrático medio.

$$ECMZ = \sqrt{(Z_R - 31)^2 + (Z_G - 70)^2 + (Z_B - 138)^2} \quad (1)$$

Donde:

ECMZ: Error cuadrático medio del color de la zona de trabajo.

Z_R : Valor de los n puntos en la matriz de rojos de la zona de trabajo.

Z_G : Valor de los n puntos en la matriz de verde de la zona de trabajo.

Z_B : Valor de los n puntos en la matriz de azules de la zona de trabajo.

Se procesa la imagen de la zona de trabajo con los parámetros del error obtenido acondicionados, una vez obtenida esta imagen pasa por un proceso de filtración para la eliminación de ruido u objetos pequeños que no corresponden a la zona de trabajo obteniendo como resultado la figura 33.

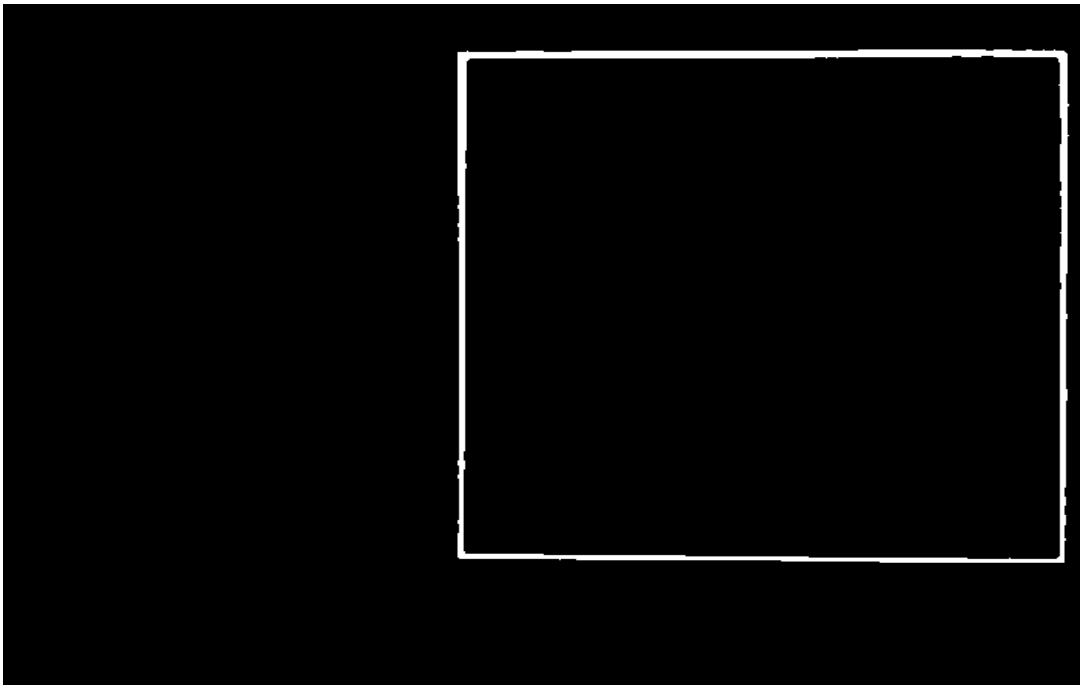


Figure 33: Imagen Binaria de la Zona de Trabajo

Fuente: Autor

La figura 33 presenta como resultado la imagen binarizada de la zona de trabajo en la cual se va a hacer la respectiva inspección a la feature. Esta imagen se obtuvo luego de la implementación de filtros y demás métodos por medio de algoritmo.

5.3.5 Detección de las 4 Esquinas de la Zona de Trabajo

Con la imagen binaria de la zona de trabajo se busca la detección de las esquinas de esta, por lo cual se extrae el tamaño de la imagen y se almacena en una matriz, se crean una matriz idéntica a la de la imagen original donde se almacenan valores de X e Y con el fin de realizar distintas operaciones matriciales para obtener los parámetros que se emplean en la búsqueda de las esquinas de la zona de trabajo en la imagen binaria.

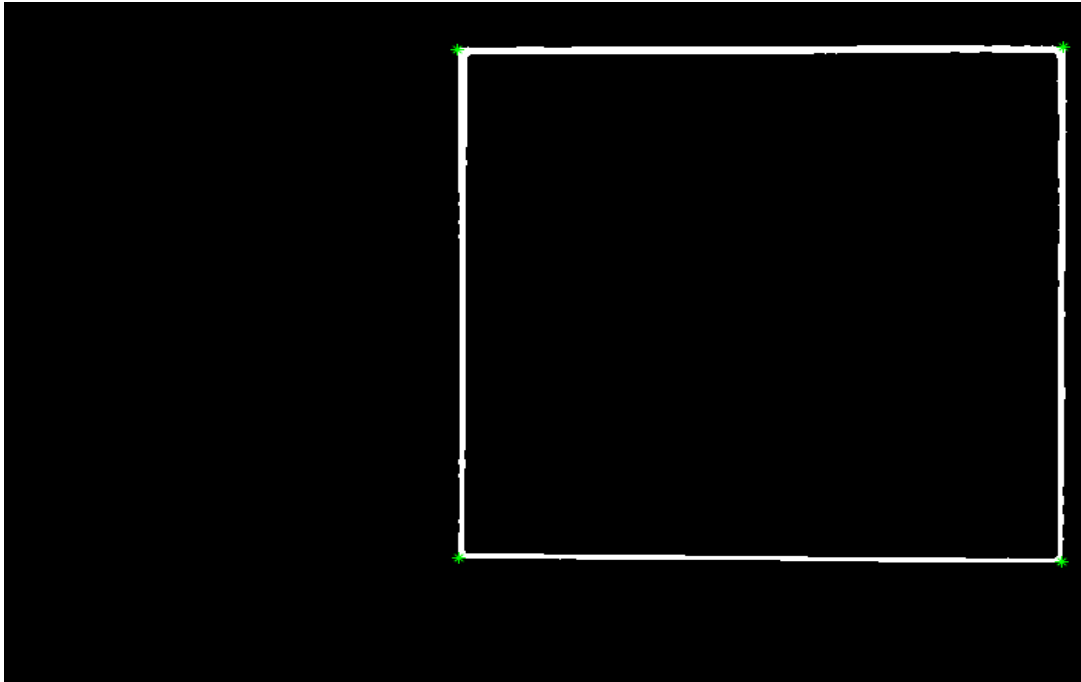


Figure 34: Detección de Esquinas en Imagen Binaria de la Zona de Trabajo

Fuente: Autor

En la figura 34 se presenta la detección de las esquinas en la imagen binaria de la zona de trabajo, donde se puede observar las esquinas de la zona de trabajo detectadas representadas con un “*” de color verde.

Con este resultado se emplea una transformación de imagen binaria a espacio de color a RGB con el objetivo de tener una mejor perspectiva visual por parte del usuario, obteniendo como resultado la figura 35.

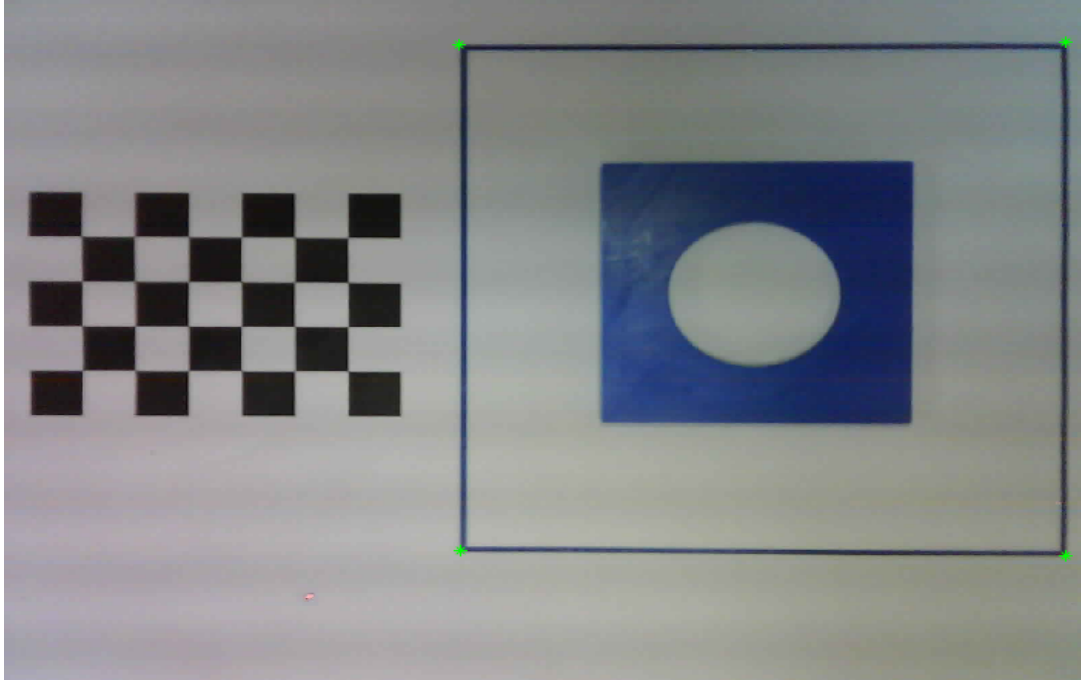


Figure 35: Detección de Esquinas en Imagen RGB de la Zona de Trabajo

Fuente: Autor

En la figura 35 se pueden observar la detección de las esquinas de la zona de trabajo en la imagen en espacio de color RGB luego de hacer la transformación de espacio de color de la imagen binaria.

5.3.6 Homografía 2D

La homografía se emplea para cambiar la perspectiva de la imagen, como se puede observar en la imagen 35 esta se encuentra desalineada, por lo tanto, requiere de emplearse una homografía en 2D para el ajuste de esta. Con la aplicación de la homografía se permite realizar cambios de coordenadas relativas de la imagen que ha sido capturada por la cámara empleada en el sistema, consiguiendo como resultado la orientación correcta de la imagen para la inspección digital.

La detección de las esquinas es de suma importancia para realizar la homografía en 2D, ya que se tiene como valores reales que la zona de trabajo es un cuadrado que cuenta con las mismas medidas cada uno de sus lados. Teniendo en cuenta estos parámetros se aplica la homografía en 2D a la imagen de la figura 35.



Figure 36: Homografía Zona de Trabajo

Fuente: Autor

En la figura 36 se exhibe la imagen obtenida luego de aplicar la homografía en 2D, se puede observar la imagen alineada y con proporcionalidad a la realidad, también se descarta el resto de la imagen y se conserva la zona de trabajo con la feature que se va a inspeccionar. Con este resultado ya se puede inspeccionar la pieza y evitar errores por desalineación de la cámara e imagen.

5.3.7 Inspección de la Pieza o Feature

La inspección de una pieza o feature permite realizar la comprobación dimensional en proceso o justo después de finalizada la construcción. De esta manera se determina si la pieza cumple correctamente con las dimensiones establecidas con el fin de disminuir el error presente en la construcción de esta.

En esta sección del algoritmo se diseñó la inspección teniendo como base la imagen obtenida con la homografía, a la cual se le aplicaron técnicas de visión artificial con el fin de calcular paralelismo, perpendicularidad y circularidad presentes en la pieza o feature que se encuentra dentro de la zona de trabajo.

se requiere la detección de la feature a inspeccionar partiendo de la figura 36, para esto se hallan los datos de los píxeles correspondientes al color que la componen realizando un recorte a una porción de esta como se muestra en la figura 37.



Figure 37: Porción de la Feature con Respectivo Color
Fuente: Autor

Seguido a esto se hallan los valores correspondientes a los píxeles de la porción extraída de la feature con ayuda de un histograma de colores para determinar los componentes en las matrices de colores en el espacio de color RGB.

En la figura 38 se puede observar el histograma de colores del color de la porción de la feature donde se tiene un valor de 31 para el componente rojo, 70 para el componente verde y 138 para el componente azul. Los valores obtenidos del histograma de colores son utilizados con el fin de encontrar los píxeles similares al color de la feature, para lo cual se emplea el error cuadrático medio con la ecuación 2.

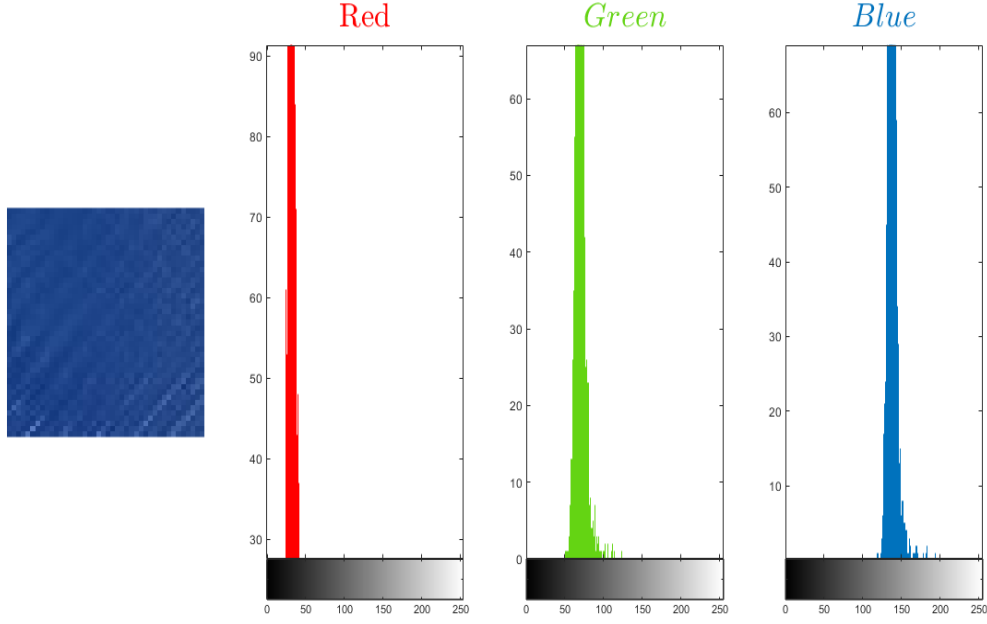


Figure 38: Histograma de Colores Porción de la Feature
Fuente: Autor

La ecuación empleada para calcular el error está dada por:

$$ECMF = \sqrt{(F_R - 31)^2 + (F_G - 70)^2 + (F_B - 138)^2} \quad (2)$$

Donde:

$ECMF$: Error cuadrático medio del color de la feature.

F_R : Valor de los n puntos en la matriz de rojos de la feature.

F_G : Valor de los n puntos en la matriz de verde de la feature.

F_B : Valor de los n puntos en la matriz de azules de la feature.

Luego se procesa la imagen de la feature con los parámetros del error calculado acondicionados. La imagen obtenida pasa por un proceso de filtración para la eliminación de ruido u objetos pequeños que no corresponden con la feature como se presenta en la figura 39.

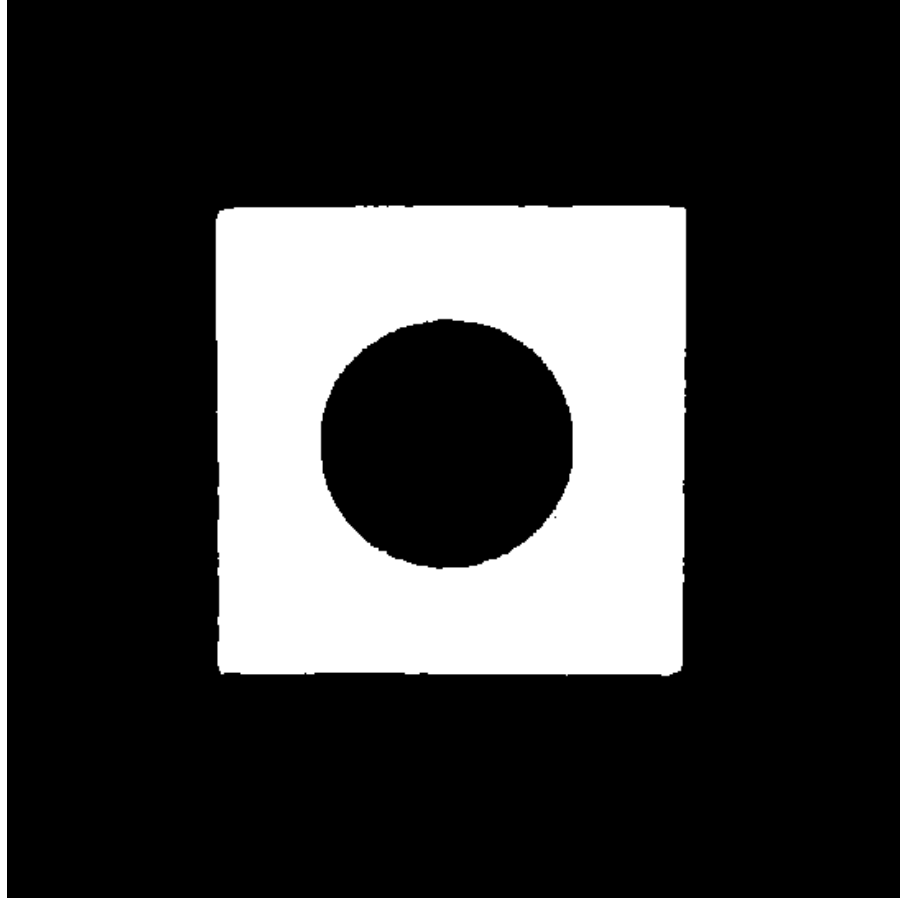


Figure 39: Imagen Binaria de la Feature
Fuente: Autor

Paralelismo y Perpendicularidad

El paralelismo y la perpendicularidad hacen parte de las tolerancias de orientación, es decir, determinan la orientación que posee la forma respecto a una referencia.

Partiendo de la imagen patrón de la feature presentada en la figura 39 para hallar el paralelismo y la perpendicularidad presentes en las líneas que conforman su estructura, se emplean técnicas de visión artificial para tratar la imagen con el fin de mantener la zona que se va a trabajar.

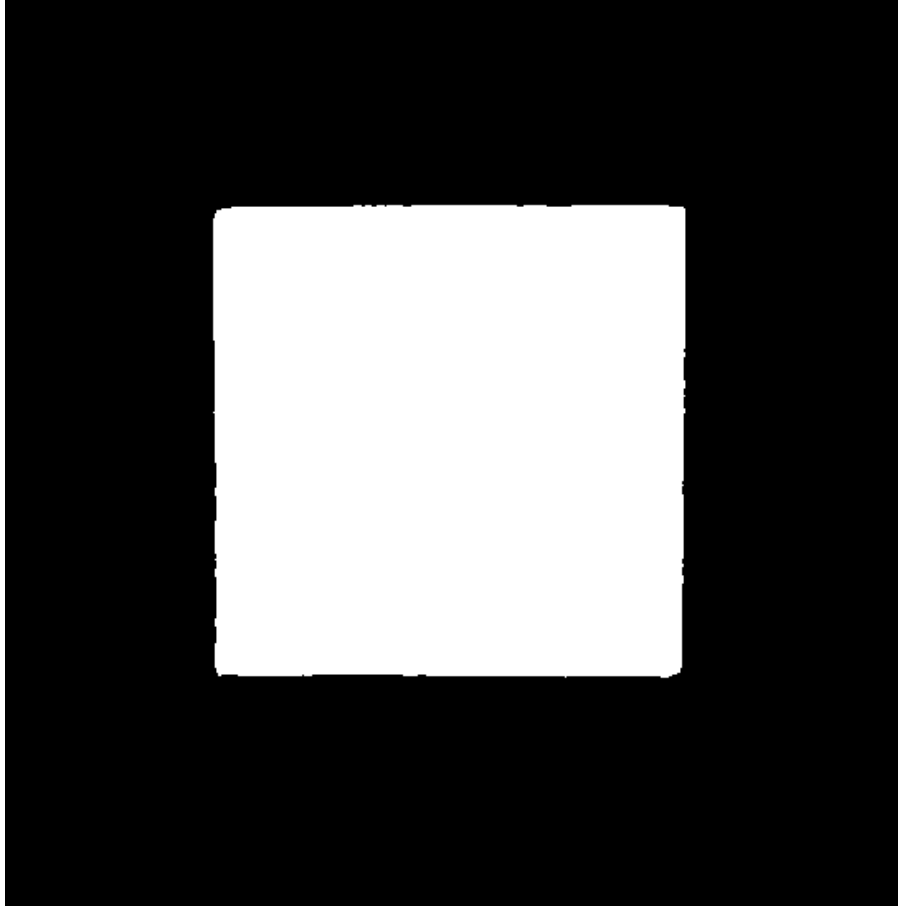


Figure 40: Imagen Patrón para calcular Paralelismo y Perpendicularidad

Fuente: Autor

En la figura 40 se presenta la imagen patrón obtenida donde se observa la silueta de la feature para calcular el paralelismo y la perpendicularidad, esta imagen se obtuvo como resultado de la aplicación de distintas técnicas de visión artificial y filtros.

La figura 40 anteriormente obtenida nos muestra que la silueta de la feature posee una forma de cuadrado y se comprende entre 4 líneas según el principio de la geometría, a esta figura se le quieren fijar puntos en los límites de la silueta para obtener nuestras líneas que conforman su geometría.

A la imagen patrón se le fijaron los puntos en los límites de la feature encontrando inicialmente los límites del elemento. Finalmente, con los puntos fijados en los límites se hace una transformación de imagen binaria a espacio de color RGB teniendo como resultado la figura 41.

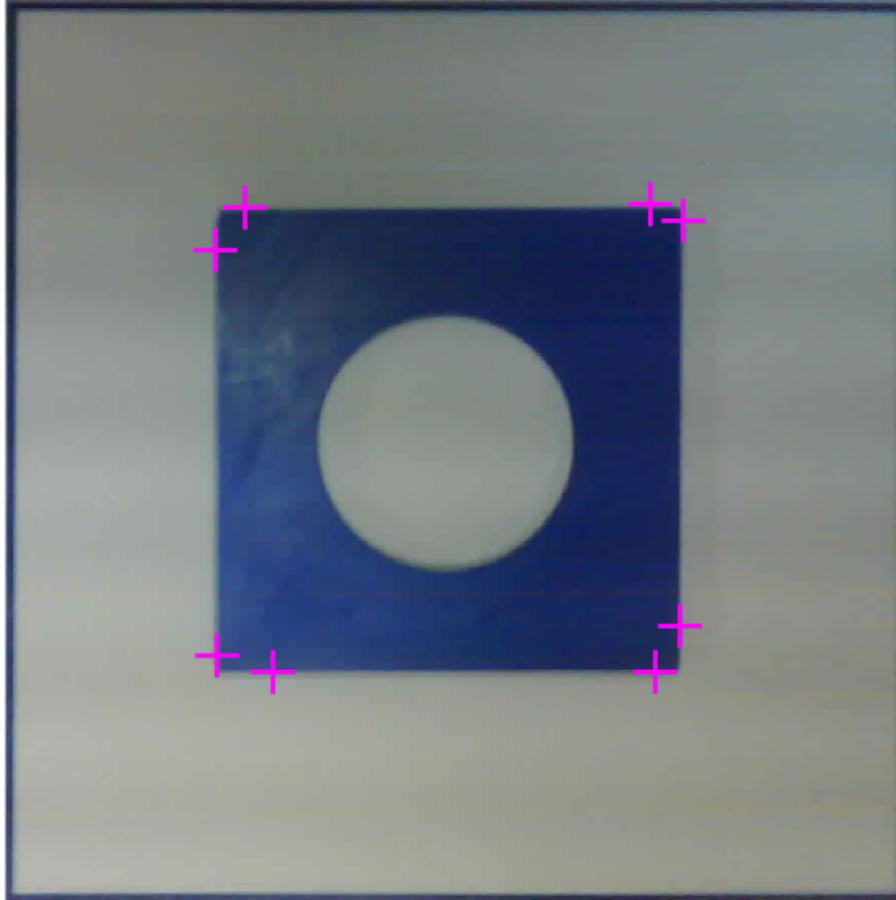


Figure 41: Fijación de Puntos en Límites de la Feature

Fuente: Autor

La figura 41 exhibe la fijación de puntos en los límites de la feature para el cálculo del paralelismo y perpendicularidad, donde los puntos fijados están representados por el símbolo “+” de color magenta.

Luego de fijar los puntos en los límites se hace una extracción de puntos en X e Y teniendo en cuenta los puntos que conforman cada línea, se asignan las coordenadas a cada uno de los 8 puntos con el objetivo de realizar una regresión lineal entre estos, para lo cual se emplea la función “fit” de MATLAB que es una herramienta que toma los puntos que le proporcionamos respecto al rango de puntos de límites fijos y procesa en una función matemática para poder calcular la regresión lineal de esta data. Finalmente, se grafican las regresiones obtenidas conformando cada lineal, como se ilustra en la figura 42.

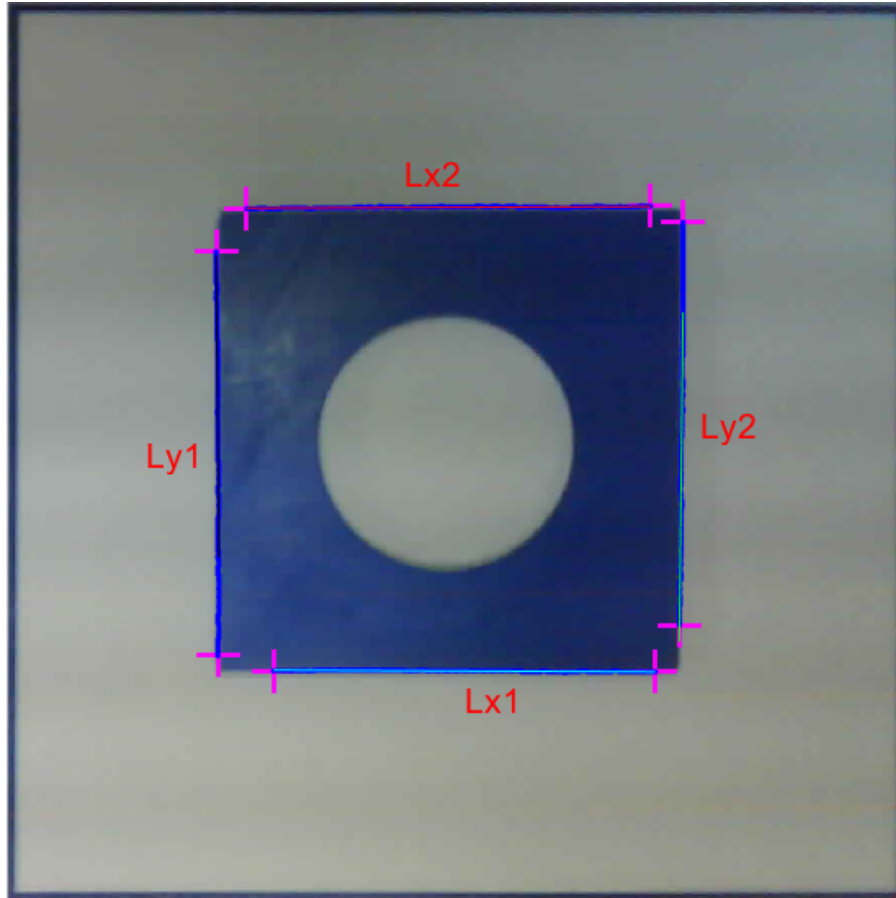


Figure 42: Regresión Lineal de Puntos Fijados en Límites de la Feature
Fuente: Autor

En la figura 42 se puede apreciar la feature a inspeccionar, la cual tiene los puntos fijados, las líneas resultantes de la regresión entre los puntos fijados en los límites. Cada línea cuenta con un color distinto y una etiqueta para su diferenciación. En la tabla 2 se presenta el nombre, color y puntos que componen cada línea obtenida de las regresiones lineales obtenidas.

ETIQUETA LÍNEA	COLOR LÍNEA	RANGO DE PUNTOS FIJOS
Ly1	Negro	P1 - P2
Lx2	Rojo	P3 - P4
Ly2	Verde	P5 - P6
Lx1	Cíen	P7 - P8

Table 2: líneas Regresión Lineal
Fuente: Autor

Con los resultados anteriores extraemos la pendiente obtenida con la regresión lineal directamente desde la función “fit” partiendo de la siguiente expresión matemática:

$$Y = m * X + b \quad (3)$$

partiendo de la ecuación 3 se despeja en función de la pendiente ”m” para obtener la pendiente obteniendo:

$$m = Y - \frac{b}{X} \quad (4)$$

Donde:

m: Pendiente de la línea.

Y: Valor calculado.

X: Variable independiente.

b: Valor de la ordenada donde la línea de regresión intercepta con el eje Y.

Obtenida la pendiente de cada línea resultado de la regresión lineal, se les debe hacer una conversión de radianes a grados, ya que inicialmente se encuentran en radianes empleando la siguiente fórmula:

$$\theta = A_R * \frac{180}{\pi} \quad (5)$$

Donde:

θ : Ángulo calculado en grados.

A_R : Ángulo de la pendiente en radianes.

π : constante matemática.

Los ángulos en grados de cada línea producto de la regresión lineal se emplean para hallar el error de ángulo presente en la feature de cada línea.

Paralelismo

Partiendo de la figura 42 donde se puede observar que la pendiente de cada línea producto de la regresión lineal posee un ángulo distinto de 0, por lo tanto se emplea una matriz de rotación respecto a la línea de referencia según el plano de inspección presentado en la figura 50, para realizar esta operación se emplea la ecuación matemática 6

$$Rot = \begin{bmatrix} \cos(\theta_R) & -\sin(\theta_R) \\ \sin(\theta_R) & \cos(\theta_R) \end{bmatrix} * \begin{bmatrix} X_I \\ Y_I \end{bmatrix} \quad (6)$$

Donde:

Rot : Rotación de la data de línea a inspeccionar respecto a la línea de referencia.

θ_R : Ángulo en grados de la inclinación de la línea de referencia.

X_I : Data de valores en X de la línea a rotar.

Y_I : Data de valores en Y de la línea a rotar.

Con los valores obtenidos de la rotación de la línea a inspeccionar respecto a la línea de referencia se halla el valor mínimo y máximo con relación a la línea de referencia, esto con el objetivo de calcular el error presente definido en píxeles entre ellos empleando la 7

$$ErP_{Pa} = \max(A_I) - \min(A_I) \quad (7)$$

Donde:

ErP_{Pa} : Error presente en línea a inspeccionar en píxeles.

A_I : data rotada respecto al ángulo de referencia.

Como el resultado del error, está en píxeles, se debe convertir a unidades de milímetros (mm), por lo cual inicialmente se emplea la imagen patrón que se presenta en la figura 40, se extraen las propiedades y se calcula la diferencia entre los valores máximos y mínimos para cada eje coordenado empleando la ecuación 8.

$$DP_{pa} = \max(A_P) - \min(A_P) \quad (8)$$

Donde:

DP_{pa} : Diferencia de píxeles en cualquier eje.

A_P : eje en el cual se quiere calcular el error.

Seguidamente, se emplea la ecuación 9 con la cual se tiene como resultado el valor de cada píxel en unidades de milímetros. Esta ecuación requiere de una medida real de entrada, ya que la fórmula consiste en una ecuación de regla de 3.

$$VP = \frac{VR}{DP_{pa}} \quad (9)$$

Donde:

VP : Valor de cada píxel en milímetros.

VR : Valor real de la medida de la pieza en cualquier eje y representada en milímetros.

DP_{pa} : Diferencia de píxeles en cualquier eje.

Finalmente, se utiliza la ecuación 10 con la cual se convierte el valor del error presente entre la línea a inspeccionar y la línea de referencia de píxeles a milímetros.

$$Erm_{pa} = ErP_{pa} * VP \quad (10)$$

Donde:

Erm_{pa} : Error presente representado en milímetros.

ErP_{pa} : Error presente en línea a inspeccionar en píxeles.

VP_{pa} : Valor de cada píxel en milímetros.

Perpendicularidad

La principal característica de la perpendicularidad es que para que exista esta debe existir un ángulo de 90° respecto a la referencia, en este proyecto se tomó como referencia las líneas especificadas en el plano de inspección presentado en la figura 50 en el capítulo de casos de estudio.

Para este caso se emplea la ecuación 6 para rotar los grados de inclinación presente en la línea de referencia y así mismo rotar la data respecto a esta. Seguidamente, se calcula el error en píxeles presente en la inspección de la línea o elemento teniendo en cuenta la línea de referencia empleando la ecuación 11.

$$ErP_{pe} = \max(B_I) - \min(B_I) \quad (11)$$

Donde:

ErP_{pe} : Error presente en línea a inspeccionar en píxeles.

B_I : data rotada respecto al ángulo de referencia.

Luego se realiza una conversión de unidades de píxel a mm, por lo cual inicialmente se extraen las propiedades de la imagen patrón que se presenta en la figura 40 y se calcula la diferencia entre el valor minio y el máximo para el eje coordenado que se quiere inspeccionar con la ecuación 12.

$$DP_{pe} = \max(B_P) - \min(B_P) \quad (12)$$

Donde:

DP_{pe} : Diferencia de píxeles en cualquier eje paralelo.

B_P : eje en el cual se quiere calcular el error.

Con el resultado de la diferencia de píxeles se utiliza la ecuación 9 para realizar una conversión de sus unidades a milímetros (mm) teniendo en cuenta una medida real, es decir, calculamos cuantos milímetros equivale 1 píxel.

Finalmente, se obtiene el valor de cada píxel y este se opera como lo dicta la ecuación 13 con el objetivo de obtener el valor del error presente en la inspección de perpendicularidad en unidades de milímetros (mm).

$$Erm_{pe} = ErP_{pe} * VP \quad (13)$$

Donde:

Erm_{pe} : Error presente representado en milímetros.

ErP_{pe} : Error presente en línea a inspeccionar en píxeles.

VP : Valor de cada píxel en milímetros.

Circularidad

Partiendo de la imagen binaria de la feature figura 39 se aplican distintas técnicas de negación de imagen y filtros para eliminar el ruido presente, obteniendo como resultado la figura 43.

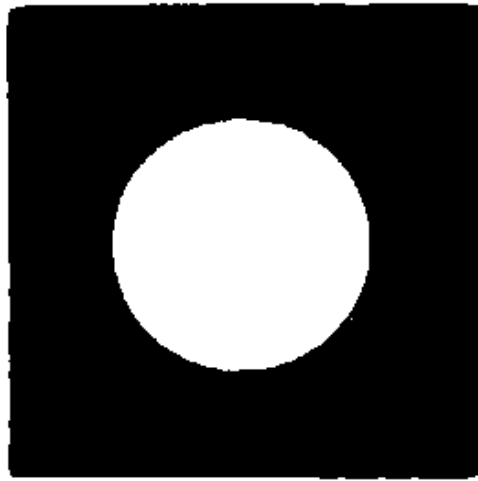


Figure 43: Imagen Binaria Negada de la Feature
Fuente: Autor

La figura 43 presenta la imagen obtenida como resultado de la negación y aplicación de filtros para la eliminación de ruido u objetos pequeños de la imagen binaria de la feature.

Como se desea eliminar las líneas en las cuales se calcula el paralelismo y la perpendicularidad, se implementa una compuerta AND entre la figura 43 y la figura 40. La compuerta AND garantiza la suma de las imágenes para así eliminar los demás elementos y quedarnos con el que se quiere inspeccionar en este caso.

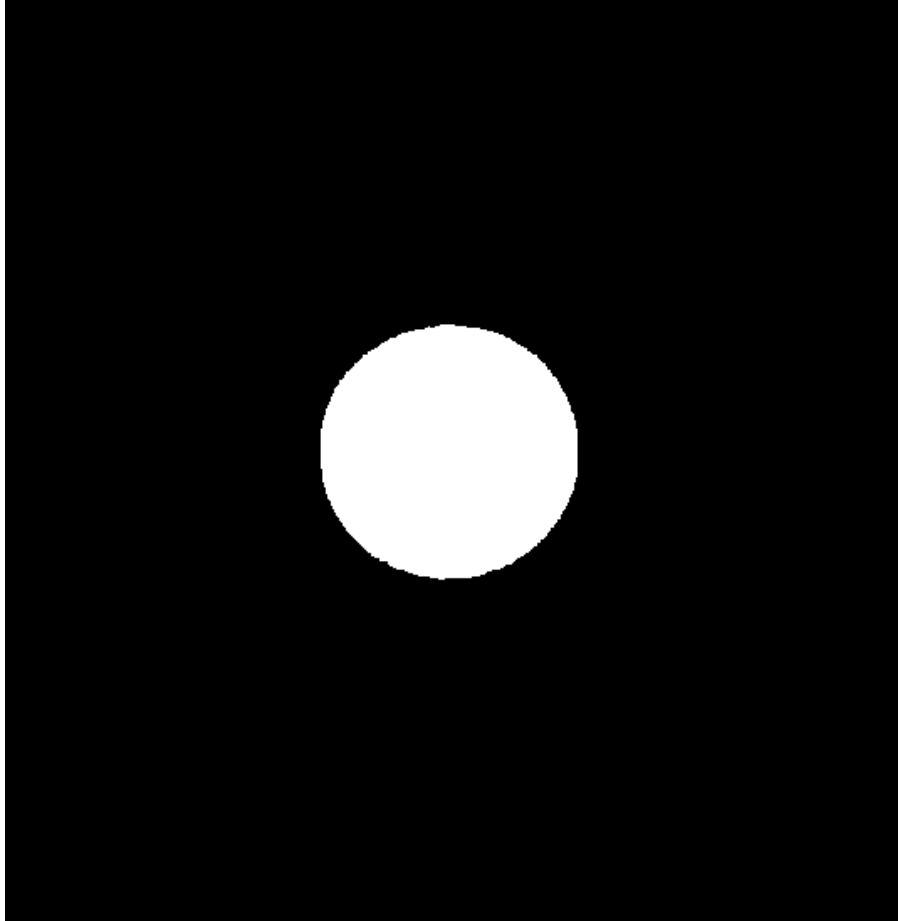


Figure 44: Imagen Patrón para Circularidad
Fuente: Autor

En la figura 44 se puede observar el resultado de emplear una compuerta AND entre las 2 figuras anteriormente mencionadas. También se evidencia en la imagen la eliminación de las líneas que componen la estructura de la feature. También se muestra la imagen patrón con la cual se va a medir la circularidad en esta feature donde se han descartado los demás componentes que corresponde a la geometría circular en el centro de la feature.

Seguidamente, se toma la imagen patrón y a esta se le trazan los límites de la región, esto es aplicable, ya que es una imagen binaria y de esta queremos obtener los límites del elemento con forma circular de la feature para finalmente realizar la extracción de los datos de interés de esta región.

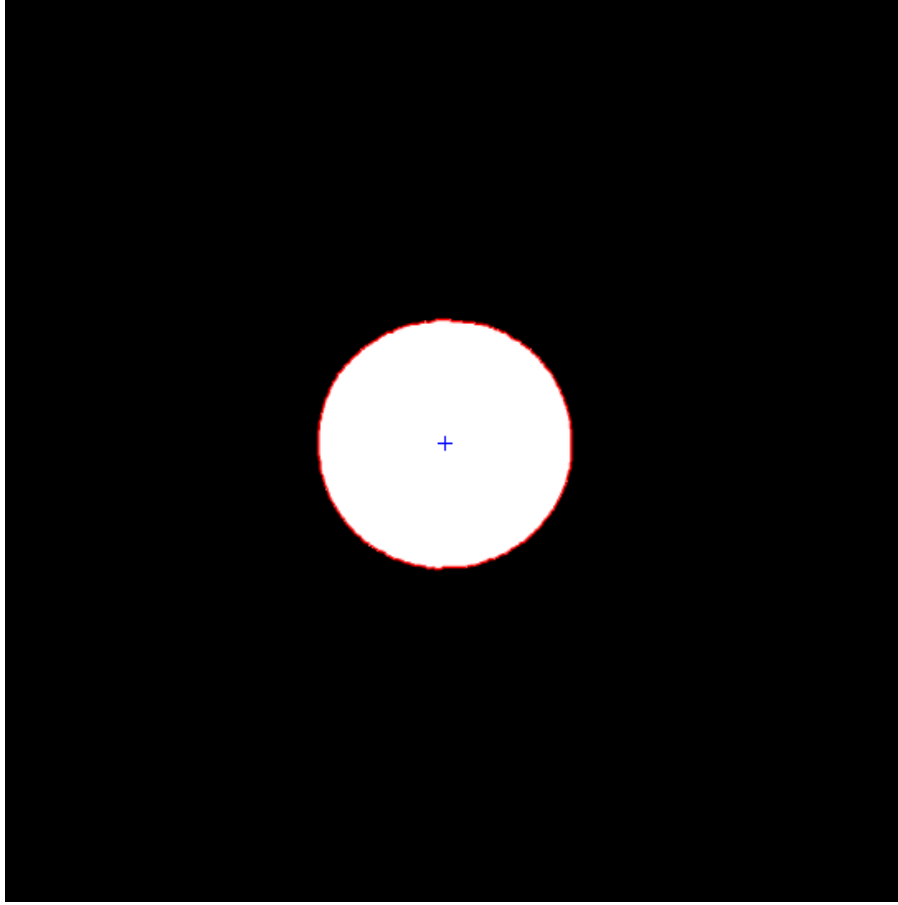


Figure 45: Centroide y Limites de la Región Circular en la Feature
Fuente: Autor

La figura 45 exhibe el límite que comprende la región de estudio con un trazo de color rojo, también se extraen los parámetros para obtener el centroide de la región circular, este centroide se identifica con el símbolo "+" de color azul e indica la coordenada del punto centro de la región circular.

Se extraen estos parámetros y se hallan los puntos que corresponden a la distancia radial máxima y mínima de la región, esto se realiza con la idea de aplicar la fórmula 14 para calcular la diferencia entre el punto máximo y el punto mínimo de esta y así obtener el resultado del error de circularidad en píxeles.

$$ErP_c = DR_{max} - DR_{min} \quad (14)$$

Donde:

ErP_c : Error presente de circularidad representado en píxeles.

DR_{max} : Distancia radial máxima de la región.

DR_{min} : Distancia radial mínima de la región.

Con el error previamente calculado en píxeles procedemos a convertirlo en milímetros, para lo cual empleamos la ecuación 9 con la cual se calcula el valor de cada píxel en milímetros y luego se calcula el error en unidades de milímetros utilizando la fórmula matemática 15.

$$Erm_c = ErP_c * VP \quad (15)$$

Donde:

Erm_c : Error de circularidad presentado en milímetros.

ErP_c : Error presente de circularidad representado en píxeles.

VP : Valor de cada píxel en milímetros.

5.3.8 Reporte de Inspección

en esta sección del algoritmo se diseñó un reporte de inspección de salida donde se evidencian todos los resultados calculados, teniendo como finalidad darle una mejor perspectiva e interacción con el usuario facilitándole los resultados.

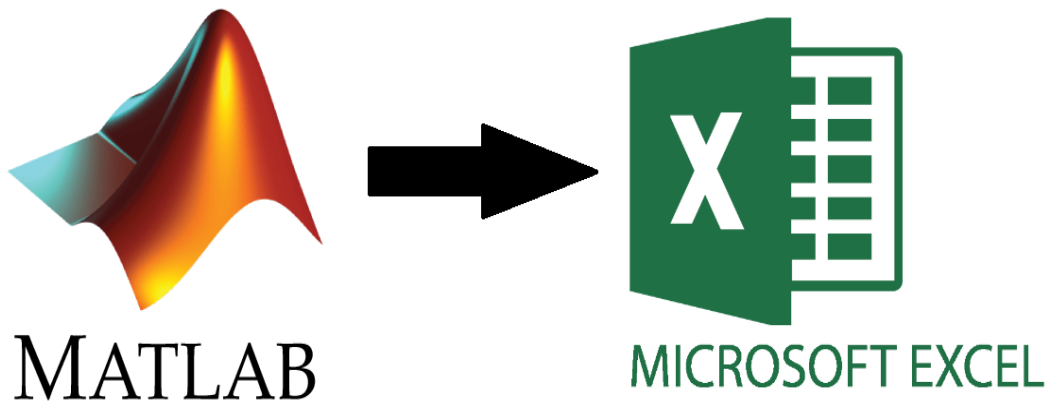


Figure 46: Exportación de los Resultados de Inspección

Fuente: Autor

La figura 46 muestra de manera gráfica el transmisor y receptor de los datos provenientes de la previa inspección de la feature, dando entender que los datos se exportan de MATLAB a MICROSOFT EXCEL de forma automática.

Tipos_de_Tolerancia	Elementos	Tolerancia	Error	Diferencia
{'Paralelismo' }	{'Lx1 - Lx2'}	0.1	0.40716	{'0.30716'}
{'Paralelismo' }	{'Ly1 - Ly2'}	0.1	0.32944	{'0.22944'}
{'Perpendicularidad'}	{'Lx1 - Ly1'}	0.1	0.29277	{'0.19277'}
{'Perpendicularidad'}	{'Lx1 - Ly2'}	0.1	0.82893	{'0.72893'}
{'Perpendicularidad'}	{'Lx2 - Ly1'}	0.1	0.64496	{'0.54496'}
{'Perpendicularidad'}	{'Lx2 - Ly2'}	0.1	0.44627	{'0.34627'}
{'Circularidad' }	{'C' }	0.2	0.91586	{'0.71586'}

Figure 47: Resultados Obtenidos con MATLAB

Fuente: Autor

La figura 47 presenta los resultados obtenidos de la inspección y se puede observar que en MATLAB se ha diseñado una tabla ordenada con los respectivos nombres y valores, esta tabla será exportada a un archivo neutro, en este caso MICROSOFT EXCEL.

Tipos de Tolerancia	Elementos	Tolerancia	Error	Diferencia
Paralelismo	Lx1 - Lx2	0,1	0,40715886	0,30716
Paralelismo	Ly1 - Ly2	0,1	0,329439963	0,22944
Perpendicularidad	Lx1 - Ly1	0,1	0,292774707	0,19277
Perpendicularidad	Lx1 - Ly2	0,1	0,828929433	0,72893
Perpendicularidad	Lx2 - Ly1	0,1	0,644959447	0,54496
Perpendicularidad	Lx2 - Ly2	0,1	0,446274988	0,34627
Circularidad	C	0,2	0,915862332	0,71586

Figure 48: Resultados Exportados a Excel

Fuente: Autor

La figura 48 exhibe los resultados obtenidos de la inspección en MATLAB y previamente exportados a MICROSOFT EXCEL. También podemos observar la tabla que se creó y los datos agrupados de forma correcta.

Capítulo 5

6 Implementación y Casos de Estudio

6.1 Introducción

En este capítulo se presenta inicialmente los casos de estudio. Se dan a conocer los resultados obtenidos del proceso de inspección empleando el sistema de inspección digital diseñado a las features o piezas de estudio. Se muestran los resultados obtenidos de la inspección realizada con la máquina de medición por coordenadas CMM Crysta-Plus M574 de la marca MITUTOYO por contacto y por medio de visión computacional. Posteriormente, se realiza un análisis y comparación de los resultados obtenidos en cada inspección con los distintos métodos.

6.2 Casos de Estudio

Se sometió la feature o pieza del caso de estudio a un proceso de inspección tanto con el sistema diseñado como distintos métodos. Se indican los resultados obtenidos con cada uno de los métodos aplicados y finalmente se realiza una comparación entre los resultados obtenidos.

6.2.1 Feature o Pieza de Estudio

La pieza del caso de estudio fue diseñada e impresa en 3D. Esta pieza se sometió a inspección con el fin de obtener el valor del error de la perpendicularidad, paralelismo y circularidad con el sistema de inspección digital diseñado y los demás métodos para así determinar si cumple dentro de la tolerancia establecida.

En la figura 49 se muestra la feature del caso de estudio en la cual se puede observar el nombre asignado a los objetos presentes, estos se han tomado como referencia para realizar la inspección con los distintos métodos.

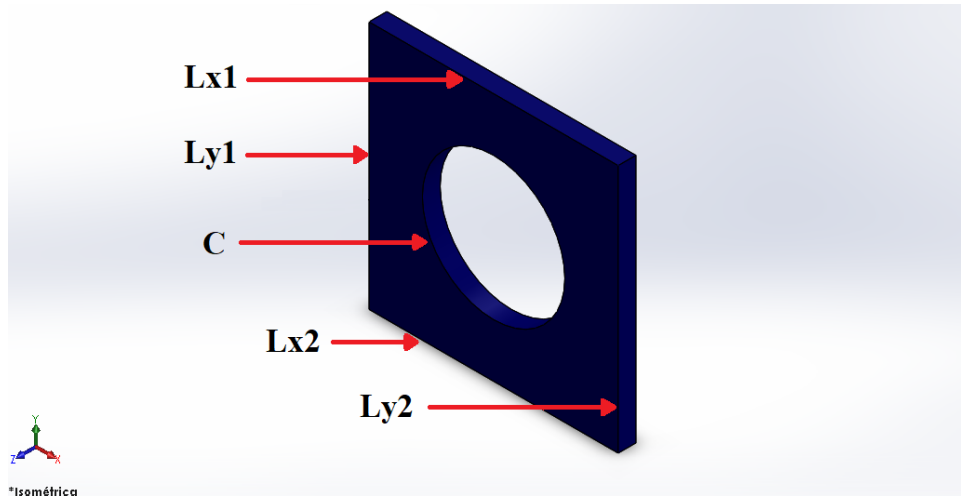


Figure 49: Objetos de la Feature Caso de Estudio a Inspeccionar
Fuente: Autor

En la tabla 3 se presentan las características de cada uno de los objetos de la feature que se inspeccionaron.

NOMBRE	GEOMETRÍA	LONGITUD/DIÁMETRO (mm)
Lx1	Lineal	70
Lx2	Lineal	70
Ly1	Lineal	70
Ly2	Lineal	70
C	Circular	40

Table 3: Características a Inspeccionar de la Feature Caso de Estudio
Fuente: Autor

Como la inspección de esta feature se va a realizar por medio de una imagen tomada con una cámara, esta se desarrolla en 2D. Se somete a este proceso de inspección la cara superior de la feature y se involucran los elementos señalados anteriormente para la respectiva inspección dimensional y geométrica.

Teniendo en cuenta las medidas y la feature a inspeccionar se realiza un plano de inspección a esta fijando las tolerancias de cada característica a inspeccionar, el cual se muestra en la figura 50

La figura 50 muestra el plano diseñado para la feature o pieza de estudio. Este plano cuenta con el elemento a tolerar y la tolerancia definida para cada uno, además de exhibirnos las líneas de referencia con el fin de calcular el error de cada operación.

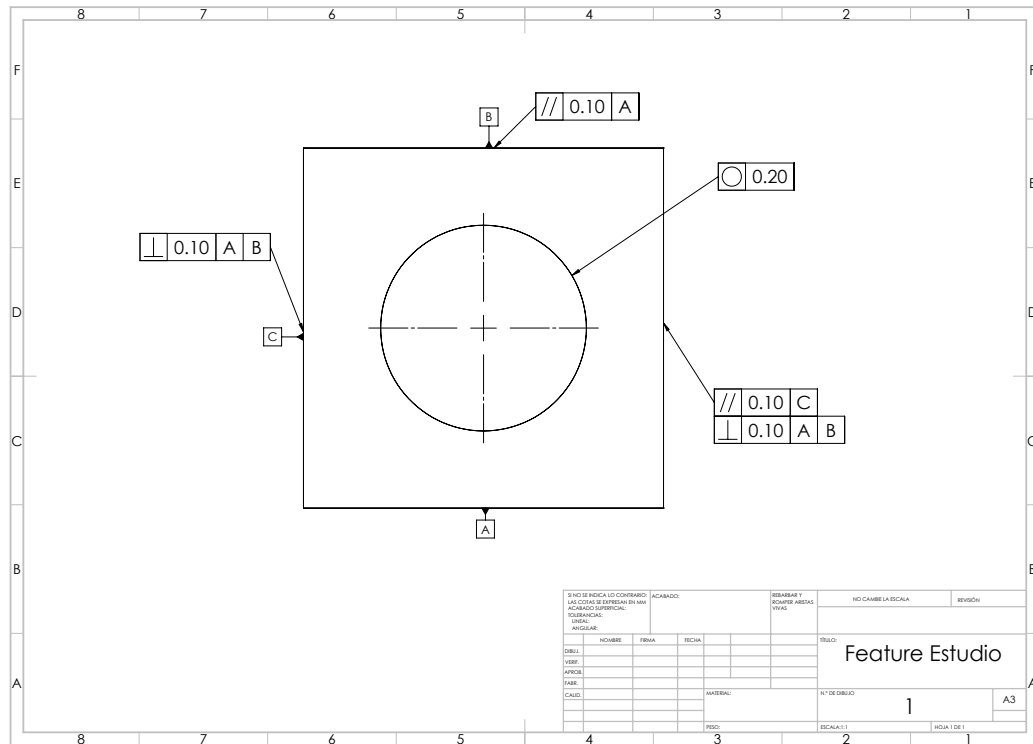


Figure 50: Plano de la Feature de Estudio
Fuente: Autor

6.2.2 Resultados Obtenidos Sistema de Inspección Digital

Se realizó la inspección digital de la feature con el sistema diseñado para la inspección digital empleando técnicas de visión artificial, obteniendo los resultados del paralelismo, perpendicularidad y circularidad como se presenta en la tabla 4.

SÍMB	CARACT GEOMÉTRICA	ELEMENTO	TOLERANCIA (mm)	ERROR (mm)	DIFERENCIA (mm)
//	Paralelismo	Lx1 - Lx2	0.1	0.40716	0.30716
//	Paralelismo	Ly1 - Ly2	0.1	0.32944	0.22944
⊥	Perpendicularidad	Lx1 - Ly1	0.1	0.29277	0.19277
⊥	Perpendicularidad	Lx1 - Ly2	0.1	0.82893	0.72893
⊥	Perpendicularidad	Lx2 - Ly1	0.1	0.64496	0.54496
⊥	Perpendicularidad	Lx2 - Ly2	0.1	0.44627	0.34627
O	Circularidad	C	0.2	0.91586	0.71586

Table 4: Resultado Sistema de Inspección Digital Caso de Estudio
Fuente: Autor

6.2.3 Resultados Obtenidos Máquina de Medición por Coordenadas

Se efectuó la inspección de la feature por el método de tacto empleando la máquina de medición por coordenadas Crysta-Plus M574 fabricada por la empresa MITUTOYO obteniendo los resultados del paralelismo, perpendicularidad y circularidad como se presenta en la tabla 5.

SÍMB	CARACT GEOMÉTRICA	ELEMENTO	TOLERANCIA (mm)	ERROR (mm)	DIFERENCIA (mm)
//	Paralelismo	Lx1 - Lx2	0.1	0.096	xxxxxx
//	Paralelismo	Ly1 - Ly2	0.1	0.053	xxxxxx
⊥	Perpendicularidad	Lx1 - Ly1	0.1	0.090	xxxxxx
⊥	Perpendicularidad	Lx1 - Ly2	0.1	0.065	xxxxxx
⊥	Perpendicularidad	Lx2 - Ly1	0.1	0.027	xxxxxx
⊥	Perpendicularidad	Lx2 - Ly2	0.1	0.057	xxxxxx
O	Circularidad	C	0.2	0.133	xxxxxx

Table 5: Resultados Maquina de Medición por Coordenadas Caso de Estudio
Fuente: Autor

6.2.4 Resultados Obtenidos Maquina de Medición por Coordenadas Visión Computacional

Se llevo a cabo la inspección de la feature por el método de visión computacional o "Vision Probe" empleando la máquina de medición por coordenadas Crysta-Plus M574 fabricada por la empresa MITUTOYO obteniendo los resultados del paralelismo, perpendicularidad y circularidad como se presenta en la tabla 6.

SÍMB	CARACT GEOMÉTRICA	ELEMENTO	TOLERANCIA (mm)	ERROR (mm)	DIFERENCIA (mm)
//	Paralelismo	Lx1 - Lx2	0.1	0.146	0.0460
//	Paralelismo	Ly1 - Ly2	0.1	0.092	0.0080
⊥	Perpendicularidad	Lx1 - Ly1	0.1	0.161	0.0610
⊥	Perpendicularidad	Lx1 - Ly2	0.1	0.258	0.1580
⊥	Perpendicularidad	Lx2 - Ly1	0.1	0.015	xxxxxx
⊥	Perpendicularidad	Lx2 - Ly2	0.1	0.115	0.0150
O	Circularidad	C	0.2	0.102	xxxxxx

Table 6: Resultados Máquina de Medición por Coordenadas Visión Computacional Caso de Estudio

Fuente: Autor

6.2.5 Análisis de Resultados

Uno de los enfoques que resalta la industria 4.0 es la digitalización de procesos industriales, pues su principio es el procesamiento de datos o procesos de forma digital. Como lo dicta la industria 4.0 la integración de tecnologías digitales para capturar y evaluar datos, teniendo como fin emplearse para la toma de decisiones, mejorando así la eficiencia y eficacia de los procesos industriales. Este trabajo presenta la construcción de un sistema de bajo costo para la inspección digital, dimensional y geométrica aplicable a features producto de procesos industriales de forma automática y así determinar si esta cumple con los criterios de diseño.

Los métodos de inspección y máquinas actuales requieren de intervención humana en un 100% y cuentan con código propietario, el sistema que se desarrolla en este trabajo investigativo reduce significativamente la intervención humana y cuenta con codificación de código abierto en el cual se pueden conseguir el valor de distinto tipo de tolerancias geométricas y dimensionales.

Dentro de las herramientas o sistemas utilizados para la inspección geométrica y dimensional existen las máquinas de medición por coordenadas o CMM, las cuales poseen precios elevados respecto a su precisión. En este trabajo se realizó la inspección de la feature o pieza de estudio empleando el sistema diseñado de bajo costo, el cual oscila entre \$1000 USD y \$1500 USD y, por otro lado, se empleó una Máquina de Medición por Coordenadas Crysta-Plus M574 producida por la marca MITUTOYO la cual su costo oscila entre \$15000 USD y \$20000 USD, este precio varía dependiendo de los elementos adicionales.

El sistema diseñado posee ventajas de costo respecto a la Máquina de Medición por Coordenadas CMM, cabe resaltar que la precisión es una desventaja en el sistema de inspección digital comparado con la CMM, ya que esta posee una precisión micrométrica.

Las tolerancias definidas para la inspección de esta feature implementando los demás métodos fue la misma y la cual se indica en la figura 50 esta se aplicó a los distintos métodos con el objetivo de visualizar la diferencia del error presente en cada método.

Los procesos de inspección por máquinas de coordenadas se ven seriamente afectados por factores externos, entre ellos: la temperatura, calibración de la punta de contacto y error humano. El sistema de inspección digital diseñado en esta investigación descarta factores como la temperatura, calibración y disminuye el error humano. La precisión de este sistema no solamente está sujeta a la calibración, sino también a la calidad del dispositivo de captura de imágenes.

La exportación de los resultados a archivos neutros permite adherir estos al concepto de lazo cerrado de control en procesos de manufactura, además de facilitar la interacción y comprensión por parte del usuario.

Capítulo 6

7 Conclusiones y Trabajos Futuros

7.1 Conclusiones

- La investigación teórica realizada referente a los procesos de manufactura y sus estándares de calidad permitió adentrarnos en el contexto de requerimiento de implementación de sistemas de medición con el fin de suplir estos estándares, además de incurrir en factores económicos positivos en la producción en masa.
- Las tolerancias GD&T definen las características conducentes para la determinación de errores dimensionales y geométricos, las cuales se tuvieron en cuenta para modelar el CAD de la Feature o Pieza de Estudio que se inspeccionó.
- Luego de un estudio de las características en que comprenden las tolerancias GD&T se pudo deducir que la perpendicularidad, paralelismo y circularidad son características que se pueden medir en 2D, lo cual permite que sea aplicable tecnología de visión artificial por medio del tratamiento digital de imágenes.
- La impresión 3D posibilita la transformación de un prototipo creado por computadora a la realidad mediante la adición de material de impresión por medio de la estructuración de capas, en este proyecto se implementó esta tecnología de impresión para crear la Feature o Pieza de Estudio logrando una alta calidad en la impresión de esta, impactando de forma positiva en el aumento de precisión en la inspección.
- La arquitectura propuesta para el sistema de inspección digital desarrollado proporciona una herramienta de fácil uso y bajo costo, este sistema cumple con la labor de inspeccionar el paralelismo, perpendicularidad y circularidad de forma automática, reduciendo significativamente el error e intervención humana.
- En los procesos de medición se busca obtener una mayor precisión en la inspección, por lo tanto, se diseñó el soporte del sistema para la captura de las imágenes, teniendo en cuenta que la luz es uno de los principales causantes de sombras que alteran la precisión en el proceso de la implementación de técnicas de visión artificial para la inspección digital.

- Algunos parámetros externos como la temperatura, calibración e intervención humana afectan negativamente los procesos de medición, este sistema no se ve seriamente afectado por factores de temperatura y la calibración se realiza de forma automática.
- La presentación de resultados en lenguajes neutros ofrece al empleador del sistema tener una mejor y fácil perspectiva sobre los resultados obtenidos, por otra parte, permite la posibilidad de ser retroalimentado a un lazo cerrado de manufactura aditiva.
- Comparando el sistema digital diseñado con la Máquina de Medición por Coordenadas CRYSTA-PLUS M574 y el sistema para medición de Visión Computacional de MITUTOYO se obtuvo que la precisión del sistema es menor que estos, aun así el sistema presentó buenos resultados siendo este de bajo costo en comparación a estos sistemas sofisticados y disminuyendo significativamente el tiempo de inspección.

7.2 Trabajos Futuros

El sistema de inspección digital dimensional y geométrica para procesos de manufactura proporciona los datos del error obtenido para cada elemento inspeccionado en un archivo neutro con el fin de emplearse en trabajos futuros en compensación de errores para aplicarse a un lazo cerrado de control, es decir, compensación de errores sin intervención por parte humana aumentando la calidad, la precisión y disminuyendo tiempo de producción.

La implementación del archivo neutro de salida con los datos obtenidos de los errores de inspección permitiría corregir los errores en tiempo real en la máquina que esté diseñando el producto, como también se estipula en la norma ISO 10303:242.

References

- [1] O. Erazo, “Procesos de manufactura en ingeniería industrial,” in *Universidad Nacional Abierta y a Distancia - UNAD*, 2008.
- [2] A. Kampker, C. Deutskens, K. Kreisköther, M. K. Büning, and M. Kuhn, “Return on engineering: Design to cost for electric engine production,” in *2015 5th International Electric Drives Production Conference (EDPC)*, 2015, pp. 1–6.
- [3] E. J. Muñoz Montenegro, W. Sánchez Rosales, C. A. Cárdenas Pérez, J. L. Sáenz Benavides *et al.*, “Diseño de una máquina herramienta de 3-ejes con tecnología cnc para operaciones de maquinado,” 2019.
- [4] X. W. Xu and S. T. Newman, “Making cnc machine tools more open, interoperable and intelligent—a review of the technologies,” *Computers in Industry*, vol. 57, no. 2, pp. 141–152, 2006.
- [5] J. Costa, “Procesos de manufactura con tecnología 3d,” *CTS Cafe*, pp. 40–49, 2019.
- [6] A. Bhattacharyya, T. L. Schmitz, S. W. Payne, P. R. Choudhury, and J. K. Schueller, “Introducing engineering undergraduates to cnc machine tool error compensation,” *Advances in Industrial and Manufacturing Engineering*, vol. 5, p. 100089, 2022.
- [7] Y. Zhao, X. Xu, T. Kramer, F. Proctor, and J. Horst, “Dimensional metrology interoperability and standardization in manufacturing systems,” *Computer Standards & Interfaces*, vol. 33, no. 6, pp. 541–555, 2011.
- [8] M. P. Bhandarkar and R. Nagi, “Step-based feature extraction from step geometry for agile manufacturing,” *Computers in industry*, vol. 41, no. 1, pp. 3–24, 2000.
- [9] C. I. R. Jaimes, “Conceptual framework for closed-loop inspection based on the coordinate measurement technology adherent to step-nc,” 2021.
- [10] R. R. Lipman, “Software to report product and manufacturing information in qif files,” *Journal of Research of the National Institute of Standards and Technology*, vol. 124, p. 1, 2019.
- [11] Y. F. Zhao, J. A. Horst, T. R. Kramer, W. Rippey, and R. J. Brown, “Quality information framework—integrating metrology processes,” *IFAC Proceedings Volumes*, vol. 45, no. 6, pp. 1301–1308, 2012.
- [12] H. J. Pahk, M. Jung, S. Hwang, Y. Kim, Y. Hong, and S. Kim, “Integrated precision inspection system for manufacturing of moulds having cad defined features,” *The International Journal of Advanced Manufacturing Technology*, vol. 10, no. 3, pp. 198–207, 1995.

- [13] G. Schaffer, “Taking the measure of cmms,” in *Am Mach*, no. 126, 1982, p. 145–160.
- [14] N. Hernández, “Metrología dimensional,” in *Metrología y Normalización*, 2010.
- [15] L. Espinosa Martínez and R. R. González Pérez, “Relación entre las tolerancias dimensionales y geométricas en los procesos de fabricación,” *Centro Azúcar*, vol. 46, no. 3, pp. 90–98, 2019.
- [16] KEYENCE, “Fundamentos de gd&t,” in *KEYENCE CORPORATION*, 2022.
- [17] C. Pascual, “Esp32 cam introducción y primeros pasos,” in *Programar Fácil*, 2022.
- [18] J. E. J. Ortega, E. G. O. Ponce, L. A. Z. Barragán, and M. M. Z. Angamarca, “Construcción de prototipo de una máquina cnc aplicando tolerancias dimensionales y geométricas,” *TECH CARLOS CISNEROS*, vol. 1, no. 01, 2020.
- [19] B. D. Acosta Peña, D. F. Salazar Carrascal *et al.*, “Desarrollo de un material de apoyo para la conceptualización de las tolerancias dimensionales y geométricas basados en las normas iso-gps 2011 (2017), 286-1 (2010), 286-2 (2010) y 8015 (2011) mediante el programa cad (solidworks).” 2020.
- [20] I. D. Parra Guzmán, V. D. Pardo Patiño *et al.*, “Caracterización de tolerancias dimensionales en ensambles mecánicos fabricados por manufactura aditiva.” 2021.
- [21] Y. G. Reyes *et al.*, “Implementación del escaneo óptico 3d como sistema de verificación en el sena-cmm aplicado en piñones,” *Revista Metalnova*, vol. 1, no. 3, pp. 5–10, 2020.
- [22] L. L. H. Cortés, J. A. F. Casallas, and V. E. R. Rosas, “Artificial vision applied to manufacturing process,” *Visión electrónica*, vol. 15, no. 1, p. 6, 2021.
- [23] H. Lu, Q. Cheng, X. Zhang, Q. Liu, Y. Qiao, and Y. Zhang, “A novel geometric error compensation method for gantry-moving cnc machine regarding dominant errors,” *Processes*, vol. 8, no. 8, p. 906, 2020.
- [24] J. Sousa, J. P. Mendonça, and J. Machado, “A generic interface and a framework designed for industrial metrology integration for the internet of things,” *Computers in Industry*, vol. 138, p. 103632, 2022.
- [25] E. R. Biglete, J. C. D. Cruz, M. S. Verdadero, M. C. E. Manuel, A. R. Altea, A. J. O. Lubi, A. G. R. Gatpayat, and C. D. B. Santos, “Dimensional accuracy evaluation of 3d-printed parts using a 3d scanning surface metrology technique,” in *2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC)*. IEEE, 2020, pp. 185–190.

- [26] S. N. Al Sunny, X. Liu, and M. R. Shahriar, “Development and optimization of an mtconnect based edge computing node for remote monitoring in cyber manufacturing systems,” in *2020 IEEE International Conference on Fog Computing (ICFC)*. IEEE, 2020, pp. 38–43.
- [27] B. Inigo, N. Colinas-Armijo, L. N. L. de Lacalle, and G. Aguirre, “Digital twin-based analysis of volumetric error mapping procedures,” *Precision Engineering*, vol. 72, pp. 823–836, 2021.

Apéndices

8 Apéndices y Anexos

8.1 Algoritmo Adquisición de imágenes

```
#include <WebServer.h>
#include <WiFi.h>
#include <esp32cam.h>

const char* WIFI_SSID = "pc3";
const char* WIFI_PASS = "gr@co12345";

WebServer server(80);

static auto loRes = esp32cam::Resolution::find(320, 240);
static auto hiRes = esp32cam::Resolution::find(1600, 1200);

void
serveJpg()
{
    auto frame = esp32cam::capture();
    if (frame == nullptr) {
        Serial.println("CAPTURE FAIL");
        server.send(503, "", "");
        return;
    }
    Serial.printf("CAPTURE OK %dx%d %db\n", frame->getWidth(), frame->getHeight(),
        static_cast<int>(frame->size()));

    server.setContentLength(frame->size());
    server.send(200, "image/jpeg");
    WiFiClient client = server.client();
    frame->writeTo(client);
}

void
handleJpgLo()
{
    if (!esp32cam::Camera.changeResolution(loRes)) {
        Serial.println("SET-LO-RES FAIL");
    }
    serveJpg();
}

void
handleJpgHi()
{
    if (!esp32cam::Camera.changeResolution(hiRes)) {
        Serial.println("SET-HI-RES FAIL");
    }
    serveJpg();
}

void
handleJpg()
{
    server.setHeader("Location", "/cam-hi.jpg");
    server.send(302, "", "");
}

void
handleMjpeg()
{
    if (!esp32cam::Camera.changeResolution(hiRes)) {
        Serial.println("SET-HI-RES FAIL");
    }
}
```

```

Serial.println("STREAM BEGIN");
WiFiClient client = server.client();
auto startTime = millis();
int res = esp32cam::Camera.streamMjpeg(client);
if (res <= 0) {
    Serial.printf("STREAM ERROR %d\n", res);
    return;
}
auto duration = millis() - startTime;
Serial.printf("STREAM END %dfrm %0.2ffps\n", res, 1000.0 * res / duration);
}

void
setup()
{
    Serial.begin(115200);
    Serial.println();

    {
        using namespace esp32cam;
        Config cfg;
        cfg.setPins(pins::AiThinker);
        cfg.setResolution(hiRes);
        cfg.setBufferCount(2);
        cfg.setJpeg(80);

        bool ok = Camera.begin(cfg);
        Serial.println(ok ? "CAMERA OK" : "CAMERA FAIL");
    }

    WiFi.persistent(false);
    WiFi.mode(WIFI_STA);
    WiFi.begin(WIFI_SSID, WIFI_PASS);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
    }

    Serial.print("http://");
    Serial.println(WiFi.localIP());
    Serial.println(" /cam-lo.jpg");
    Serial.println(" /cam-hi.jpg");
    Serial.println(" /cam.mjpeg");

    server.on("/cam-lo.jpg", handleJpgLo);
    server.on("/cam-hi.jpg", handleJpgHi);
    server.on("/cam.jpg", handleJpg);
    server.on("/cam.mjpeg", handleMjpeg);

    server.begin();
}

void
loop()
{
    server.handleClient();
}

```

8.2 Calibración de la cámara

```
% Sistema Automatico de Inspeccion Digital en Procesos de Manufactura
% Brayan S. Figueroa
% Universidad de Pamplona

%Obtención imágenes cámara ip para calibración
mkdir('calibracion2');
cd('calibracion1');
cam = ipcam('http://192.168.137.222/cam.mjpeg')
%cam = webcam(1)
preview(cam)
disp('presione barra espaciadora para tomar las imagenes de referencia')
pause;

for i=1:8
    imageFileName = sprintf('image%d.bmp', i)
    imag = snapshot(cam);
    imwrite(imag,imageFileName);
    imageFileNames{i} = imageFileName;
    figure(2);
    montage(imageFileNames, 'Size', [4 3]);
    pause;
end

clear cam;

%Definir imagenes para el proceso
imageFileNames = {'C:\Users\BRAYAN\Desktop\tesis\codigos y vision\calibracion1\image1.
bmp',...
'C:\Users\BRAYAN\Desktop\tesis\codigos y vision\calibracion1\image2.bmp',...
'C:\Users\BRAYAN\Desktop\tesis\codigos y vision\calibracion1\image3.bmp',...
'C:\Users\BRAYAN\Desktop\tesis\codigos y vision\calibracion1\image4.bmp',...
'C:\Users\BRAYAN\Desktop\tesis\codigos y vision\calibracion1\image5.bmp',...
'C:\Users\BRAYAN\Desktop\tesis\codigos y vision\calibracion1\image6.bmp',...
'C:\Users\BRAYAN\Desktop\tesis\codigos y vision\calibracion1\image7.bmp',...
'C:\Users\BRAYAN\Desktop\tesis\codigos y vision\calibracion1\image8.bmp',...
};
%Detectar tableros de ajedrez en imágenes
[imagePoints, boardSize, imagesUsed] = detectCheckerboardPoints(imageFileNames);
imageFileNames = imageFileNames(imagesUsed);

%obtengo el tamaño de la imagen
originalImage = imread(imageFileNames{1});
[mrows, ncols, ~] = size(originalImage);

%Genera coordenadas mundiales de las esquinas de los cuadrados
squareSize = 12.2857; % en milímetros
worldPoints = generateCheckerboardPoints(boardSize, squareSize);

%Calibración de la camara
[cameraParams, imagesUsed, estimationErrors] = estimateCameraParameters(imagePoints,
worldPoints,...
'EstimateSkew', false, 'EstimateTangentialDistortion', false, ...
'NumRadialDistortionCoefficients', 2, 'WorldUnits', 'millimeters', ...
'InitialIntrinsicMatrix', [], 'InitialRadialDistortion', [], ...
'ImageSize', [mrows, ncols]);

%grafica proyeccion de errores
h1=figure; showReprojectionErrors(cameraParams);

%Grafica ubicaciones de patrones
h2=figure; showExtrinsics(cameraParams, 'CameraCentric');

%errors de estimacion de parametros
displayErrors(estimationErrors, cameraParams);

%remocion de efectos de distorsion
undistortedImage = undistortImage(originalImage, cameraParams);
imshow(undistortedImage)
imwrite(undistortedImage, 'C:\Users\BRAYAN\Desktop\tesis\codigos y
vision\piezar_prueba.bmp');
```

8.3 Detección de la Zona de Trabajo

```
%% Leemos la imagen con la feature

RGB = imread('piezar_prueba.bmp');
f1=figure(1); , set(f1,'Color',[1,1,1]);
imshow(RGB);

%% color Zona de Trabajo

a= imread('imh.jpg');
subplot(1,4,1);
imshow(a);
rc=a(:, :,1);
subplot(1,4,2);
imhist(rc);
gc=a(:, :,2);
subplot(1,4,3);
imhist(gc);
bc=a(:, :,3);
subplot(1,4,4);
imhist(bc);

%% Detección de la zona de trabajo

RGBZ = double(RGB);          % Imagen RGB de la zona de trabajo en la imagen
colorZ = [31, 70, 138];      % Valor de pixeles de la zona de trabajo en la imagen

% Se obtiene el error cuadratico medio
Err = ( (RGBZ(:, :,1)- colorZ(1)).^2 + (RGBZ(:, :,2)- colorZ(2)).^2 + (RGBZ(:, :,3)- colorZ(3)).^2 ).^0.5;
Err_tolerable = 85;          % Valor del Error tolerable

%acondicionamiento de parametros del error en la imagen de la zona de
%trabajo
Err(Err <= Err_tolerable) = 1;
Err(Err > Err_tolerable) = 0;
featbn = logical(Err);        %convierte los valores numericos del error a logicos
featbn = bwareaopen(featbn,20000); %Remueve objetos con menos de 20000 pixeles
se = strel('disk',10);        %Elemento estructurante
featbn = imclose(featbn,se);    %Operacion Closing
featbn = bwareafilt(featbn,1,'smallest'); %Conserva el objeto mas pequeño

% Gráfica de la detección de la feature en la imagen
f2 = figure(2); set(f2,'Color',[1,1,1]);
imshow(featbn);
```

8.4 Detección 4 Esquinas en la Zona de Trabajo

```
function corners = Detect4corners(featbn)
    [nf,nc]=size(featbn);% Extraemos tamaño original de la imagen

    %Creamos matriz X con columnas idénticas y matriz Y con filas son
    %idénticas del mismo tamaño de la imagen original
    [X,Y] = meshgrid(1:nc,1:nf);

    % Multiplicacion de la matriz binaria con X y Y respectivamente
    Xm = X.*featbn;
    Ym = Y.*featbn;
    D = Xm.^2+Ym.^2; %Resultado de la suma de la matriz binaria multiplicada por X y Y

    % Buscamos esquina maxima de Izquierda a Derecha
    Cmax_ID=[X(find(D==max(max(D)))) , Y(find(D==max(max(D))))];
    D(D==0)=1000000000000;

    % Buscamos esquina minima de Izquierda a Derecha
    Cmin_ID=[X(find(D==min(min(D)))) , Y(find(D==min(min(D))))];
    D(D==1000000000000)=0;

    %%%
    Xm2 = (X-nc).*featbn;
    D2 = Xm2.^2+Ym.^2;

    % Buscamos esquina maxima de Derecha a Izquierda
    Cmax_DI=[X(find(D2==max(max(D2)))) , Y(find(D2==max(max(D2))))];
    D2(D2==0)=1000000000000;
    % Buscamos esquina minima de Derecha a Izquierda
    Cmin_DI=[X(find(D2==min(min(D2)))) , Y(find(D2==min(min(D2))))];
    D2(D2==1000000000000)=0;

    %Esquinas encontradas en la figura
    corners = [Cmin_ID;Cmin_DI;Cmax_ID;Cmax_DI];
end

%% Detección 4 esquinas en la zona de trabajo

corners = Detect4corners(featbn);
X = [corners(:,1)];
Y = [corners(:,2)];
indc1 = 0;
indc2 = 0;
X(1) = X(1) - indc1;
Y(1) = Y(1) - indc1;
X(2) = X(2) + indc1;
Y(2) = Y(2) - indc1;
X(3) = X(3) + indc2;
Y(3) = Y(3) + indc2;
X(4) = X(4) - indc2;
Y(4) = Y(4) + indc2;

% Gráfica de la imagen de la zona de trabajo con cada esquina encontrada
f3= figure(3);
imshow(featbn); hold on;
plot(X,Y, '*','Color','g');
```

8.5 Homografía 2D

```
function P = Homograph2Dfeat(X,Y,x,y)
    A=zeros(8,8);
    A(1,:)=[X(1),Y(1),1, 0, 0, 0, -1*X(1)*x(1), -1*Y(1)*x(1)];
    A(2,:)=[0, 0, 0, X(1),Y(1),1, -1*X(1)*y(1), -1*Y(1)*y(1)];
    A(3,:)=[X(2),Y(2),1, 0, 0, 0, -1*X(2)*x(2), -1*Y(2)*x(2)];
    A(4,:)=[0, 0, 0, X(2),Y(2),1, -1*X(2)*y(2), -1*Y(2)*y(2)];
    A(5,:)=[X(3),Y(3),1, 0, 0, 0, -1*X(3)*x(3), -1*Y(3)*x(3)];
    A(6,:)=[0, 0, 0, X(3),Y(3),1, -1*X(3)*y(3), -1*Y(3)*y(3)];
    A(7,:)=[X(4),Y(4),1, 0, 0, 0, -1*X(4)*x(4), -1*Y(4)*x(4)];
    A(8,:)=[0, 0, 0, X(4),Y(4),1, -1*X(4)*y(4), -1*Y(4)*y(4)];

    % Se crea un vector columna para almacenar los valores de X y Y para cada par de
    coordenadas cartesianas
    v=[x(1);y(1);x(2);y(2);x(3);y(3);x(4);y(4)];
    % Solucionamos en sistema de ecuaciones
    ec = A\v;
    e = reshape([ec;1],3,3)';
    % Se escoge projective2d y se ingresa la matriz de transformación "e"
    % transpuesta para transformar la proyectiva
    P=projective2d(e');
end

%% Homografía

% Se crean 2 nuevos vectores columna con los coeficientes de Transformación
x=[1;1000;1000;1];
y=[1;1;1000;1000];

% Hacemos la Homografía 2-D con los datos de los Coeficientes de Transformación
P = Homograph2Dfeat(X,Y,x,y);
xWorldLimits = [1 1000];
yWorldLimits = [1 1000];
tic
RGB_H = imwarp(RGB,P,'FillValues',255,'OutputView',imref2d([1000 1000 3],xWorldLimits,
yWorldLimits));

% Gráfica del la imagen de la zona de trabajo con la homografía aplicada
f5=figure(5);, set(f5,'Color',[1,1,1]);
imshow(RGB_H);
```

8.6 Inspección

```
%%                               Inspección                               %%

% Deteccion de la feature
RGBF = double(RGB_H); % Imagen RGB de la zona de trabajo en la imagen
colorF = [31, 70, 138]; % Valor de pixeles de la zona de trabajo en la imagen

% Se obtiene el error cuadratico medio
Err_li = ( (RGBF(:, :, 1) - colorF(1)).^2 + (RGBF(:, :, 2) - colorF(2)).^2 ...
           + (RGBF(:, :, 3) - colorF(3)).^2 ).^0.5;

% Condiciones para el procesamiento de la imagen
Err_tolerableli = 85; % Valor del Error tolerable
Err_li(Err_li <= Err_tolerableli) = 1;
Err_li(Err_li > Err_tolerableli) = 0;
patron = logical(Err_li); %convierte los valores numericos del error a logicos
patron = bwareaopen(patron, 100000); %Remueve objetos con menos de 200000 pixeles
patron_lin = bwareafilt(patron, 1, 'largest'); %Conserva el objeto mas grande
patron_lin = bwareaopen(not(patron_lin), 700); %Remueve objetos con menos de 1000
pixeles en la imagen negada
patron_lin = bwareafilt(patron_lin, 1, 'largest'); %Conserva el objeto mas grande
patron_lin = not(patron_lin); %Niega la imagen
% Grafica de la feature patron para perpendicularidad y paralelismo
f6 = figure(6); set(f6, 'Color', [1, 1, 1]);
imshow(patron_lin);

% Clasifica elementos segun sus limites
[B, L] = bwboundaries(patron_lin, 'noholes');
hold on
for i = 1:length(B);
    elemento = B{i};
    plot(elemento(:, 2), elemento(:, 1), 'r', 'LineWidth', 1)
end

% Puntos en los limites de la feature
P1 = 50;
P2 = 500;
P3 = 550;
P4 = 1000;
P5 = 1075;
P6 = 1500;
P7 = 1575;
P8 = 2025;

% Puntos para etiqueta de las lineas
PL1 = 225;
PL2 = 775;
PL3 = 1288;
PL4 = 1800;

% Grafica de los puntos tomados en la feature
figure(5);
hold on;
plot(elemento(P1, 2), elemento(P1, 1), '+m', 'MarkerSize', 20, 'LineWidth', 2);
plot(elemento(P2, 2), elemento(P2, 1), '+m', 'MarkerSize', 20, 'LineWidth', 2);
plot(elemento(P3, 2), elemento(P3, 1), '+m', 'MarkerSize', 20, 'LineWidth', 2);
plot(elemento(P4, 2), elemento(P4, 1), '+m', 'MarkerSize', 20, 'LineWidth', 2);
plot(elemento(P5, 2), elemento(P5, 1), '+m', 'MarkerSize', 20, 'LineWidth', 2);
plot(elemento(P6, 2), elemento(P6, 1), '+m', 'MarkerSize', 20, 'LineWidth', 2);
plot(elemento(P7, 2), elemento(P7, 1), '+m', 'MarkerSize', 20, 'LineWidth', 2);
plot(elemento(P8, 2), elemento(P8, 1), '+m', 'MarkerSize', 20, 'LineWidth', 2);
legend off;
```

```

% Extraccion de puntos en X e Y
PLx2 = elemento(P1:P2,:);
PLy2 = elemento(P3:P4,:);
PLx1 = elemento(P5:P6,:);
PLy1 = elemento(P7:P8,:);

% Asignacion de coordenadas de los puntos
Ly1X= PLy1(:,2);
Ly1Y= PLy1(:,1);
Lx2X= PLx2(:,2);
Lx2Y= PLx2(:,1);
Ly2X= PLy2(:,2);
Ly2Y= PLy2(:,1);
Lx1X= PLx1(:,2);
Lx1Y= PLx1(:,1);

% Regresion lineal de los punto obtenidos
fitLy1 = fit(Ly1X,Ly1Y,'poly1'); %Regresion lineal para Ly1
fitLx2 = fit(Lx2X,Lx2Y,'poly1'); %Regresion lineal para Lx2
fitLy2 = fit(Ly2X,Ly2Y,'poly1'); %Regresion lineal para Ly2
fitLx1 = fit(Lx1X,Lx1Y,'poly1'); %Regresion lineal para Lx1

% Grafica de lineas resultantes de la regresión lineal
figure(5);
plot(fitLy1,'k',Ly1X,Ly1Y); %dibuja regresion lineal de Ly1 y la data Ly1X y Ly
plot(fitLx2,'r',Lx2X,Lx2Y); %dibuja regresion lineal de Lx2 y la data Lx2X y Lx
plot(fitLy2,'g',Ly2X,Ly2Y); %dibuja regresion lineal de Ly2 y la data Ly2X y Ly
plot(fitLx1,'c',Lx1X,Lx1Y); %dibuja regresion lineal de Lx1 y la data Lx1X y Lx

% etiquetas de lineas
text(elemento(PL1,2),elemento(PL1,1)-40,'Lx2', 'Color','r', 'FontSize',16)
text(elemento(PL2,2)+20,elemento(PL2,1),'Ly2', 'Color','r', 'FontSize',16)
text(elemento(PL3,2),elemento(PL3,1)+30,'Lx1', 'Color','r', 'FontSize',16)
text(elemento(PL4,2)-80,elemento(PL4,1),'Ly1', 'Color','r', 'FontSize',16)

% hallamos el valor de la pendiente de cada regresion ineal
PenLy1 = coeffvalues(fitLy1); % Obtiene los valor de los coeficientes
PenLy1 = PenLy1(1); % toma el primer coeficiente(p1) o pendiente
PenLx2 = coeffvalues(fitLx2); % Obtiene los valor de los coeficientes
PenLx2 = PenLx2(1); % toma el primer coeficiente(p2) o pendiente
PenLy2 = coeffvalues(fitLy2); % Obtiene los valor de los coeficientes
PenLy2 = PenLy2(1); % toma el primer coeficiente(p1) o pendiente
PenLx1 = coeffvalues(fitLx1); % Obtiene los valor de los coeficientes
PenLx1 = PenLx1(1); % toma el primer coeficiente(p2) o pendiente

% Convertimos a grados el valor de las pendientes
GPLy1 = atan(PenLy1)*(180/pi);
GPLx2 = atan(PenLx2)*(180/pi);
GPLy2 = atan(PenLy2)*(180/pi);
GPLx1 = atan(PenLx1)*(180/pi);

% Valor absoluto para los angulos en grados
GPLy1a = abs(GPLy1);
GPLx2a = abs(GPLx2);
GPLy2a = abs(GPLy2);
GPLx1a = abs(GPLx1);

```

```

%% Paralelismo

% Rotacion de data Lx2 respecto a linea de refenecia Lx1
AnRPLx1 = -GPLx1;
MrotLx1 = [cosd(AnRPLx1) -sind(AnRPLx1); sind(AnRPLx1) cosd(AnRPLx1)];
rotLx2 = MrotLx1 * PLx2';
rotLx2 = rotLx2';
min_Lx2Y = min(rotLx2(:,1));
max_Lx2Y = max(rotLx2(:,1));
P_errorLx2 = max_Lx2Y - min_Lx2Y;

% Rotacion de data Ly2 respecto a linea de refenecia Ly1
AnRPLy1 = 90-GPLy1;
MrotLy1 = [cosd(AnRPLy1) -sind(AnRPLy1); sind(AnRPLy1) cosd(AnRPLy1)];
rotLy2 = MrotLy1 * PLy2';
rotLy2 = rotLy2';
min_Ly2X = min(rotLy2(:,2));
max_Ly2X = max(rotLy2(:,2));
P_errorLy2 = max_Ly2X - min_Ly2X;

%% Perpendicularidad

% Rotacion de data Ly1 respecto a linea de refenecia Lx1
AnRPLx1pe = -GPLx1;
MrotLx1pe = [cosd(AnRPLx1pe) -sind(AnRPLx1pe); sind(AnRPLx1pe) cosd(AnRPLx1pe)];
rotLy1Lx1pe = MrotLx1pe * PLy1';
rotLy1Lx1pe = rotLy1Lx1pe';
min_Ly1Lx1Xpe = min(rotLy1Lx1pe(:,2));
max_Ly1Lx1Xpe = max(rotLy1Lx1pe(:,2));
P_errorLy1Lx1pe = max_Ly1Lx1Xpe - min_Ly1Lx1Xpe;

% Rotacion de data Ly2 respecto a linea de refenecia Lx1
rotLy2Lx1pe = MrotLx1pe * PLy2';
rotLy2Lx1pe = rotLy2Lx1pe';
min_Ly2Lx1Xpe = min(rotLy2Lx1pe(:,2));
max_Ly2Lx1Xpe = max(rotLy2Lx1pe(:,2));
P_errorLy2Lx1pe = max_Ly2Lx1Xpe - min_Ly2Lx1Xpe;

% Rotacion de data Ly1 respecto a linea de refenecia Lx2
AnRPLx2pe = -GPLx2;
MrotLx2pe = [cosd(AnRPLx2pe) -sind(AnRPLx2pe); sind(AnRPLx2pe) cosd(AnRPLx2pe)];
rotLy1Lx2pe = MrotLx2pe * PLy1';
rotLy1Lx2pe = rotLy1Lx2pe';
min_Ly1Lx2Xpe = min(rotLy1Lx2pe(:,2));
max_Ly1Lx2Xpe = max(rotLy1Lx2pe(:,2));
P_errorLy1Lx2pe = max_Ly1Lx2Xpe - min_Ly1Lx2Xpe;

% Rotacion de data Ly2 respecto a linea de refenecia Lx2
rotLy2Lx2pe = MrotLx2pe * PLy2';
rotLy2Lx2pe = rotLy2Lx2pe';
min_Ly2Lx2Xpe = min(rotLy2Lx2pe(:,2));
max_Ly2Lx2Xpe = max(rotLy2Lx2pe(:,2));
P_errorLy2Lx2pe = max_Ly2Lx2Xpe - min_Ly2Lx2Xpe;

```

```

%% calculo de angulos respecto a rotaciones

% % Regresion lineal de los puntos obtenidos
fitLx2N = fit(rotLx2(:,2),rotLx2(:,1),'poly1'); %Regresion lineal para Lx2
fitLy2N = fit(rotLy2(:,2),rotLy2(:,1),'poly1'); %Regresion lineal para Ly2
fitLy2M = fit(rotLy1Lx1pe(:,2),rotLy1Lx1pe(:,1),'poly1'); %Regresion lineal para Ly2
fitLy1M = fit(rotLy2Lx1pe(:,2),rotLy2Lx1pe(:,1),'poly1'); %Regresion lineal para Ly1
fitLy1p = fit(rotLy1Lx2pe(:,2),rotLy1Lx2pe(:,1),'poly1'); %Regresion lineal para Ly1
fitLy2p = fit(rotLy2Lx2pe(:,2),rotLy2Lx2pe(:,1),'poly1'); %Regresion lineal para Ly2

% Hallar las pendientes de las regresiones lineales
PenLx2N = coeffvalues(fitLx2N); % Obtiene los valor de los coeficientes
PenLx2N = PenLx2N(1); % toma el primer coeficiente(p2) o pendiente
PenLy2N = coeffvalues(fitLy2N); % Obtiene los valor de los coeficientes
PenLy2N = PenLy2N(1); % toma el primer coeficiente(p2) o pendiente
PenLy1M = coeffvalues(fitLy1M); % Obtiene los valor de los coeficientes
PenLy1M = PenLy1M(1); % toma el primer coeficiente(p2) o pendiente
PenLy2M = coeffvalues(fitLy2M); % Obtiene los valor de los coeficientes
PenLy2M = PenLy2M(1); % toma el primer coeficiente(p2) o pendiente
PenLy1p = coeffvalues(fitLy1p); % Obtiene los valor de los coeficientes
PenLy1p = PenLy1p(1); % toma el primer coeficiente(p2) o pendiente
PenLy2p = coeffvalues(fitLy2p); % Obtiene los valor de los coeficientes
PenLy2p = PenLy2p(1); % toma el primer coeficiente(p2) o pendiente

% Convertimos a grados el valor de las pendientes
GPLx2N = atan(PenLx2N)*(180/pi);
GPLy2N = atan(PenLy2N)*(180/pi);
GPLy1M = atan(PenLy1M)*(180/pi);
GPLy2M = atan(PenLy2M)*(180/pi);
GPLy1p = atan(PenLy1p)*(180/pi);
GPLy2p = atan(PenLy2p)*(180/pi);

%% Detección de circularidad en la feature

patron_cir = not(patron); % Niega la imagen
patron_cir = bwareaopen(patron_cir,4000); % Remueve objetos con menos de 100 pixeles
patron_cir = and(patron_lin,patron_cir); % Operación AND para sumar las imagenes
% Grafica de la imagen patron para circularidad
f7 = figure(7);
imshow(patron_cir); % Muestra la imagen

% Clasifica elementos segun sus limites
[C,CI] = bwboundaries(patron_cir,'noholes');
hold on
for j=1:length(C);
    elemento_cir = C{j};
    plot(elemento_cir(:,2),elemento_cir(:,1),'r','LineWidth',1)
end

REGIONES = bwlabel(patron_cir); % Etiqueta las regiones
ESTADISTICAS = regionprops(REGIONES,'all'); %Estadisticas
EJEX=ESTADISTICAS(1).Centroid(2);
EJEY=ESTADISTICAS(1).Centroid(1);
RGB_H(fix(EJEX),:,:)=255;
RGB_H(:,fix(EJEY),:)=255;

```

```

% Propiedades circulo
Circ = ESTADISTICAS.Circularity;
center = ESTADISTICAS.Centroid;
diameter = mean([ESTADISTICAS.MajorAxisLength ESTADISTICAS.MinorAxisLength],2);
DPC = ESTADISTICAS.MajorAxisLength - ESTADISTICAS.MinorAxisLength;
plot(center(:,1),center(:,2),'+b')

%% Conversion a pixeles

Rlin = bwlabel(patron_lin); % Etiqueta las regiones
EstadLin = regionprops(Rlin,'all'); %Estadisticas
DPX = EstadLin.MaxFeretCoordinates(1,1) - EstadLin.MaxFeretCoordinates(1,2); %✓
Diferencia Pixeles en X
DPY = EstadLin.MaxFeretCoordinates(2,1) - EstadLin.MaxFeretCoordinates(2,2); %✓
Diferencia Pixeles en Y
DPX = abs(DPX);
DPY = abs(DPY);
DPXY = mean([DPX DPY]); %Diferencia Pixeles en X e Y
VR = 70; %valor real de la pieza
VP = VR / DPXY; %Valor del Pixel en mm
err_Pa_Lx2 = P_errorLx2*VP; %Error paralelismo entre Lx1 y Lx2
err_Pa_Ly2 = P_errorLy2*VP; %Error paralelismo entre Ly1 y Ly2
err_Pe_Ly1Lx1 = P_errorLy1Lx1pe*VP; %Error perpendicularidad entre Lx1 y Ly1
err_Pe_Ly2Lx1 = P_errorLy2Lx1pe*VP; %Error perpendicularidad entre Lx1 y Ly2
err_Pe_Ly1Lx2 = P_errorLy1Lx2pe*VP; %Error perpendicularidad entre Lx2 y Ly1
err_Pe_Ly2Lx2 = P_errorLy2Lx2pe*VP; %Error perpendicularidad entre Lx2 y Ly1
err_C = DPC*VP; %Error circularidad en C

%% Tolerancias

TPaLx2Lx1 = 0.1;
TPaLy2Ly1 = 0.1;
TPeLy1Lx1 = 0.1;
TPeLy2Lx1 = 0.1;
TPeLy1Lx2 = 0.1;
TPeLy2Lx2 = 0.1;
Tc = 0.2;

% diferencia entre el error y la tolerancia

DPaLx2Lx1 = TPaLx2Lx1 - err_Pa_Lx2;
DPaLy2Ly1 = TPaLy2Ly1 - err_Pa_Ly2;
DPeLy1Lx1 = TPeLy1Lx1 - err_Pe_Ly1Lx1;
DPeLy2Lx1 = TPeLy2Lx1 - err_Pe_Ly2Lx1;
DPeLy1Lx2 = TPeLy1Lx2 - err_Pe_Ly1Lx2;
DPeLy2Lx2 = TPeLy2Lx2 - err_Pe_Ly2Lx2;
DCirC = Tc - err_C;

% Condición para el cumplimiento de la tolerancia

if err_Pa_Lx2 < TPaLx2Lx1
    varpaLx1Lx2 = '*****';
else
    varpaLx1Lx2 = abs(DPaLx2Lx1);
    varpaLx1Lx2 = num2str(varpaLx1Lx2);
end

if err_Pa_Ly2 < TPaLy2Ly1
    varpaLy1Ly2 = '*****';
else
    varpaLy1Ly2 = abs(DPaLy2Ly1);
    varpaLy1Ly2 = num2str(varpaLy1Ly2);
end
end

```

```

if err_Pe_Ly1Lx1 < TPeLy1Lx1
    varpeLy1Lx1 = '*****';
else
    varpeLy1Lx1 = abs(DPeLy1Lx1);
    varpeLy1Lx1 = num2str(varpeLy1Lx1);
end

if err_Pe_Ly2Lx1 < TPeLy2Lx1
    varpeLy2Lx1 = '*****';
else
    varpeLy2Lx1 = abs(DPeLy2Lx1);
    varpeLy2Lx1 = num2str(varpeLy2Lx1);
end

if err_Pe_Ly1Lx2 < TPeLy1Lx2
    varpeLy1Lx2 = '*****';
else
    varpeLy1Lx2 = abs(DPeLy1Lx2);
    varpeLy1Lx2 = num2str(varpeLy1Lx2);
end

if err_Pe_Ly2Lx2 < TPeLy2Lx2
    varpeLy2Lx2 = '*****';
else
    varpeLy2Lx2 = abs(DPeLy2Lx2);
    varpeLy2Lx2 = num2str(varpeLy2Lx2);
end

if err_C < Tc
    varcirC = '*****';
else
    varcirC = abs(DCircC);
    varcirC = num2str(varcirC);
end

%% Reporte de Inspección

Tipos_de_Tolerancia = {'Paralelismo';'Paralelismo'; 'Perpendicularidad';...
    'Perpendicularidad';'Perpendicularidad';'Perpendicularidad';'Circularidad'};
Elementos = {'Lx1 - Lx2';'Ly1 - Ly2';'Lx1 - Ly1';...
    'Lx1 - Ly2';'Lx2 - Ly1';'Lx2 - Ly2'; 'C'};
Tolerancia = [TPaLx2Lx1; TPaLy2Ly1; TPeLy1Lx1; TPeLy2Lx1;...
    TPeLy1Lx2; TPeLy2Lx2; Tc];
Error = [err_Pa_Lx2; err_Pa_Ly2; err_Pe_Ly1Lx1; err_Pe_Ly2Lx1;...
    err_Pe_Ly1Lx2; err_Pe_Ly2Lx2; err_C];
Diferencia = {varpaLx1Lx2; varpaLy1Ly2; varpeLy1Lx1; varpeLy2Lx1;...
    varpeLy1Lx2; varpeLy2Lx2; varcirC};
T = table(Tipos_de_Tolerancia, Elementos, Tolerancia, Error, Diferencia)

writetable(T, 'Reporte de Inspección.xls', 'Sheet', 1, 'Range', 'A1');

```