



Universidad de Matanzas

Facultad de Ciencias Técnicas

Departamento de Informática

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN INFORMÁTICA

Monitoreo de parámetros de pestañado en cilindros de 10 Kg mediante inspección visual automatizada.

Autor: Aylin Morales Ortega

Tutores: M Sc. Darien Menéndez Molina

M Sc. Briseida Blanco Alfonso

A G R A D E C I M I E N T O S

DECLARACIÓN DE AUTORIDAD Y NOTA LEGAL

RESUMEN

Los botellones de gas licuado abastecen el combustible con el cual funcionan la mayoría de las cocinas cubanas. La calidad con que están fabricados es de gran importancia, debido a esto, defectos en su construcción podrían causar pérdidas materiales y de vidas humanas. Por tanto, los métodos de monitoreo de la calidad son de gran importancia y deben realizarse de forma breve y efectiva para evitar que no se vea afectado el ritmo de producción, ocasionando pérdidas de tiempo y recursos. En el presente trabajo se muestra un sistema de monitoreo para el control de la calidad del proceso de pestañado en cilindros de 10kg, utilizando técnicas de procesamiento digital de imágenes. El método se basa en el uso de un haz de luz estructurada, el cual al incidir sobre la superficie de la pestaña genera una imagen que es captada por una cámara digital. Una unidad de procesamiento ejecuta un algoritmo capaz de identificar las partes de la pestaña y obtener mediciones sobre sus dimensiones. Esta solución permite realizar la inspección visual a los cilindros con mayor precisión y rapidez que un operario humano, posibilitando la automatización del proceso en la fábrica.

ABSTRACT

The liquefied gas bottles supply the fuel with the best performance of the Cuban kitchens. The quality with which they are manufactured is of great importance, materials and human lives. Because of this, the quality monitoring methods are of great importance, which go in a brief and effective way so that the production rhythm is seen and the losses of time and resources are minimal. The work shows a monitoring system for the quality control of the process of tabs in cylinders of 10 kg, using techniques of digital image processing. The method is based on the use of a structured light, which is located on the surface of the eyelash generates an image that is captured by a digital camera. Once the image is processed, the noise is eliminated by means of a Gaussian filter, it is transformed to the hue-saturation-luminosity (HSL) space, it is segmented by the red color and the region of interest is selected, the laser is skeletonized, they are calculated. The points can be detected by transformed Hough lines. The parameters can be seen through a graphic viewer that allows a quick interpretation of the results.

Contenido

Introducción	1
DEFINICIÓN DEL PROBLEMA	2
HIPÓTESIS	2
OBJETIVO GENERAL	2
OBJETIVOS ESPECÍFICOS	2
Métodos teóricos empleados	3
Métodos empíricos empleados	3
Capítulo 1. MARCO TEÓRICO-REFERENCIAL	4
1.1 Introducción	4
1.2 Introducción a la inspección visual automatizada	4
1.2.1 Inspección visual automatizada	4
1.2.2 Procesamiento digital de imágenes	5
1.2.3 Inteligencia Artificial	5
1.2.4 Ruido en imágenes digitales	6
1.3 Componentes y elementos de un sistema de visión artificial	7
1.3.1 Iluminación	7
1.3.2 Cámaras	8
1.3.3 Unidad de procesamiento	8
1.4 Ventajas y desventajas de los sistemas de visión artificial	9
1.5 Herramientas, librerías, lenguajes informáticos y metodología de desarrollo de software a utilizar	10
1.5.1 Lenguaje informático	10

1.5.2 Librerías	10
1.5.3 Metodología de desarrollo de software.....	11
1.5.4 Herramientas de modelado	13
1.6 Conclusiones parciales	14
Capítulo 2. SOLUCIÓN TEORICA DEL PROBLEMA CIENTIFICO	15
2.1 Introducción:.....	15
2.2 Elementos básicos de ingeniería de software:.....	15
2.2.1 Modelado del negocio	15
2.2.2 Descripción de Actores y Casos de uso del Negocio	16
2.2.3 Equipo de Trabajo y Roles	19
2.2.4 Pila del Producto (ProductBacklog)	20
2.2.5 Pila de Sprint (Sprint Backlog)	21
2.2.6 Planificación del Sprint.	23
2.3 Algoritmo utilizado para la inspección visual automática	28
2.4 Software (Tutorial).....	39
2.4.1 Interfaz principal	40
2.4.2 Funcionalidad Ajustar	40
2.4.3 Funcionalidad Calibrar	40
2.4.4 Funcionalidad Inspeccionar	41
2.4.5 Funcionalidad Salir	41
2.5 Conclusiones parciales	41

Capítulo 3 Validación del sistema de monitoreo propuesto.....	42
3.1 Introducción	42
3.2 Esquema de la instalación experimental	42
3.3 Análisis de costo.....	43
3.3.1 Estimación de costo del proyecto	43
3.3.2 Coste computacional.....	48
3.4 Caso de estudio.....	49
3.4.1 Preparación	49
3.4.2 Monitoreo	50
3.5 Conclusiones parciales	53
Conclusiones	54
Recomendaciones.....	55
Bibliografía	56

Índice de Ilustraciones y Tablas

Ilustración 1: Modelado del negocio 15

Ilustración 2: Descripción de actores 16

Ilustración 3: Caso de uso realizar pestañado 16

Ilustración 4: Caso uso realizar muestreo 17

Ilustración 5: Caso de uso medir dimensión 17

Ilustración 6: Caso de uso realizar soldadura 18

Ilustración 7: Caso de uso realizar prueba de presión 18

Ilustración 8: Miembros y roles del equipo SCRUM 19

Ilustración 9: Pila del Producto (ProductBacklog) 20

Ilustración 10: Pila del Sprint (Sprint Backlog) 21

Ilustración 11: Planificación del Sprint 1 23

Ilustración 12: Planificación del Sprint 2 25

Ilustración 13: Planificación del Sprint 3 27

Ilustración 14: Diagrama de algoritmo 28

Ilustración 15: Captura de imagen 29

Ilustración 16: Filtro gaussiano 30

Ilustración 17: Espacio HSL 32

Ilustración 18: Componente Luminosidad	32
Ilustración 19: Componente Matiz	33
Ilustración 20: Componente Saturación	33
Ilustración 21: Segmentación por color	34
Ilustración 22: Extracción de la región de interés	35
Ilustración 23: Adelgazamiento de láser	36
Ilustración 24: Perfil láser	36
Ilustración 25: Identificación de los puntos de interés	38
Ilustración 26: Transformada de Hough	39
Ilustración 27: Interfaz principal	40
Ilustración 28: Prototipo de dispositivo	42
Ilustración 29: Ventana de inspección con información sobre altura	50
Ilustración 30: Ventana de inspección con información sobre el ancho	51

INTRODUCCIÓN

La automatización industrial es un fenómeno que ha revolucionado la industria moderna. Pretende controlar maquinarias o procesos industriales, basándose en el uso de sistemas o elementos computarizados y electromecánicos. Comprende la instrumentación industrial que incluye los sensores, sistemas de control y supervisión, sistemas de transmisión y recolección de datos y aplicaciones de software en tiempo real, las cuales son muy útiles en la supervisión y control de los procesos industriales.

El uso de la automatización ha permitido alcanzar en la industria un mejor control de la calidad, mayor eficiencia, aumento de la productividad y reducción del trabajo manual. Por tales motivos, desde su aparición en la industria hasta la actualidad, su ritmo de adopción ha ido en aumento, y es de especial interés por sectores, propietarios e inversores.

Desde hace años el perfeccionamiento empresarial es un proceso que han tomado un gran interés en Cuba. Mayores niveles de productividad con mejores niveles de calidad son los objetivos de este proceso nacional. Una de las industrias que está siendo evaluada y reestructurada es la de conformación de metales (CORFORMAT), actualmente encargada de la producción de cilindros de gas licuados.

La Empresa de Conformación de Matanzas "Noel Fernández" en lo adelante CONFORMAT, situada en el municipio Playa de la ciudad de Matanzas, Cuba, tiene como objetivo fundamental producir y comercializar de forma mayorista, recipientes a presión y sus componentes, artículos de uso doméstico e industrial a partir de la conformación de metales tanto para el mercado nacional como para la exportación.

Los botellones de gas licuado abastecen el combustible con el cual funcionan la mayoría de las cocinas cubanas. La calidad con que están fabricados es de gran importancia, puesto que defectos en su construcción podrían causar pérdidas materiales y de vidas humanas.

El control a la calidad se efectúa bajo la base de métodos estadísticos, donde se selecciona una pequeña muestra de la línea de producción que representa la totalidad producida. Este método es empleado para evitar el agotamiento de los inspectores de la calidad, los cuales les es imposible manualmente revisar la totalidad de los botellones durante la fabricación de cada una de sus partes, así como seguir el veloz ritmo que demanda la producción, que quedaría retrasada si se llevase a cabo una inspección exhaustiva. Al final de la línea de producción cada botellón pasa por una prueba de presión. Esta prueba verifica que soporte las condiciones a las que será sometido una vez que se comience su uso, pero debido a su baja revisión durante el proceso de producción llegan a esta etapa botellones que no sobrepasan la prueba y deben ser enviados a otra empresa para su reciclaje. Entre los parámetros de calidad a controlar durante

el proceso se encuentran: las dimensiones y el espesor de los discos, las dimensiones resultantes del proceso de embutido a los discos, las dimensiones del cordón de soldadura, las dimensiones resultantes de la etapa de pestañado, por citar solo algunas, la automatización de esta última es el motivo de la presente investigación.

DEFINICIÓN DEL PROBLEMA

¿Cómo automatizar la detención de defectos en el proceso de pestañado durante la elaboración del botellón de gas, de manera que no afecte el ritmo de producción?

HIPÓTESIS

Si se introduce en la línea de producción un sistema de monitoreo en tiempo real, capaz de realizar la inspección de forma automatizada, es posible supervisar los parámetros de calidad requeridos a la totalidad de los cilindros producidos, en una etapa temprana que permita la rápida corrección del producto y el ahorro de materiales en etapas posteriores del proceso.

OBJETIVO GENERAL

Como objetivo general se plantea alcanzar mayores niveles de calidad en la producción de cilindros de gas licuado con un mayor ahorro de recursos, mediante la identificación temprana de defectos en el producto.

OBJETIVOS ESPECÍFICOS

La disminución de la cantidad de botellones de gas licuado con defecto, al final de la línea de producción.

Valor económico: al automatizar la inspección al pestañado se identificarán imperfecciones en una etapa temprana de la producción, esto permitirá que sea posible una rápida corrección, de manera que no solo no ralentice el proceso de producción, sino que permita una aceleración del mismo. Además, la identificación a tiempo de desperfectos permite el ahorro de materiales que de otra manera serían empleados inútilmente en cilindros defectuosos.

Carácter social: la automatización de este proceso mejorará las condiciones de trabajo para el inspector de calidad. Un aumento de la calidad del producto final, aseguraría una mejor protección al consumidor.

Valor práctico: pues el resultado final de la investigación, un dispositivo de gran eficacia que podrá posteriormente ser empleado por empresas en condiciones similares que pretendan automatizar el control de calidad de sus productos.

Métodos teóricos empleados

Analítico-Sintético: permitirá como método, analizar la situación problemática y buscar la información necesaria para establecer el problema y la hipótesis de la investigación.

Hipotético-deductivo: Permitirá, a partir de la hipótesis asumida, elaborar conclusiones partiendo del análisis del conjunto de datos y de principios más generales, los resultados de la investigación.

Métodos empíricos empleados

Revisión de documentos: Se consultaron libros, trabajos, artículos relacionados con el tema.

Entrevista: Se realizaron entrevistas a especialistas del tema para identificar problemas, posibles soluciones y requerimientos a tener en cuenta.

CAPÍTULO 1. MARCO TEÓRICO-REFERENCIAL.

1.1 Introducción

En este capítulo se realiza un estudio del estado del arte relativo a los sistemas de inspección visual automatizados, la relación existente con la disciplina de inteligencia artificial, los componentes que conforman un sistema de su tipo, ventajas y desventajas que poseen, así como la definición de los conceptos fundamentales para una mejor comprensión de la temática seleccionada.

1.2 Introducción a la inspección visual automatizada

1.2.1 Inspección visual automatizada

La inspección visual automatizada surge por la necesidad de asistir o sustituir a los operarios en las actividades de inspección dentro de las líneas de montaje. Las máquinas constituyen una propuesta ideal en este aspecto, puesto que siempre producen el resultado para lo cual fueron programadas.

La inspección visual automatizada está basada en técnicas de visión artificial y procesamiento digital de imágenes. La visión artificial, pretende simular el funcionamiento de la visión humana. La adquisición de imágenes por parte de los ojos le corresponden a una o varias cámaras, y su interpretación por parte del cerebro son simulados por algoritmos computacionales que se encargan de extraer la información, analizarla y tomar decisiones. El procesamiento de imágenes se compone de un conjunto de técnicas (algoritmos computacionales) que se aplican sobre imágenes digitales con el fin de producir un efecto sobre ellas que mejore en algún aspecto la visualización o extracción de información contenida.

Uno de los beneficios que posee este tipo de inspección, es que no realiza contacto con el objeto a inspeccionar. Esta cualidad es muy útil en la industria alimenticia, semiconductores, manufactura a altas temperaturas o expuestas a corrientes eléctricas, donde es una prioridad preservar las características del objeto o la salud de un operario.

La inspección visual automatizada puede ser definida como la revisión visual de un amplio rango de productos por una máquina sin la supervisión de un humano. Esta técnica depende para su funcionamiento de la captura de imágenes por una cámara digital.

Los sistemas basados en captura de imágenes se pueden clasificar en dos tipos: sistemas de detección de anomalías y en sistemas basados en medición. Ambos tienen un objetivo común, el control del producto bajo ciertos parámetros de calidad, no todos operan de igual forma. Los sistemas basados en detección de forma, textura, intensidad, color, pertenecen a la primera clasificación. Ejemplos de tales sistemas lo podemos encontrar en los trabajos: (1) —(2) (3) (4) (5) mientras que los basados en coordenadas y métricas de distancias pertenecen a la segunda clasificación, un ejemplo de ellos los constituyen los trabajos: (6) (7).

1.2.2 Procesamiento digital de imágenes

Con el objetivo de mejorar la calidad o facilitar la búsqueda de información en imágenes digitales se emplean un conjunto de técnicas llamadas procesamiento digital de imágenes.

Para obtener a partir de una imagen inicial, otra final cuyo resultado sea más adecuado para una aplicación específica, mejorando ciertas características de la imagen que posibilite efectuar operaciones de procesado sobre ella, se utiliza las técnicas de filtrado.

La aplicación de filtros permite reducir la cantidad de variaciones de intensidad entre píxeles vecinos, eliminar aquellos píxeles cuyo nivel de intensidad es muy diferente al de sus vecinos y cuyo origen puede estar tanto en el proceso de adquisición de la imagen como en el de transmisión, destacar los bordes que se localizan en una imagen, detectar los píxeles donde se produce un cambio brusco en la función intensidad. Por tanto, se consideran los filtros como operaciones que se aplican a los píxeles de una imagen digital para optimizarla, enfatizar cierta información o conseguir un efecto especial en ella.

1.2.3 Inteligencia Artificial

Para la obtención de resultados más eficientes en la inspección visual, la inteligencia artificial ha jugado un destacado papel, consiguiendo en muchos casos, la búsqueda de parámetros óptimos para los algoritmos empleados, ofreciendo soluciones que de otra manera sería difícil de obtener.

Con el objetivo de aprender a reconocer los defectos en los productos inspeccionados se emplean un grupo de algoritmos estándares de aprendizaje supervisado. El artículo presentado en (8), muestra ejemplos de los algoritmos más empleados: arboles de decisiones, clasificadores estadísticos, redes neuronales y máquinas de soporte de vectores.

Son muchas las técnicas de inteligencia artificial usadas en procesos de inspección visual en la industria. (9) muestra un ejemplo del empleo del método de recocido simulado, para la identificación de defectos en la industria textil. (10) presenta un modelo empleando algoritmos evolutivos, utilizando la variante CoDE para la búsqueda de parámetros óptimos a emplear con filtro de Gabor para la detección de defectos. Los resultados obtenidos mejoran los indicadores de efectividad y rendimientos que los algoritmos anteriores aplicados en su especialidad, un ejemplo es el propuesto por (9) (11), que aplican otras técnicas de inteligencia artificial.

Junto con métodos de inspección visual se han empleado sistemas de expertos. Un ejemplo de ello lo es (12), cuya propuesta proporciona al operador una base de conocimientos para consulta en la inspección de la tela. Para mejorar la efectividad del sistema, los autores proponen una innovadora combinación con un sistema de diagnóstico basado en la teoría de conjuntos difusos.

En (13) se muestra un ejemplo de la aplicación de las Redes Neuronales a la inspección automatizada, donde se construye una red neuronal multicapa para identificar los defectos en imágenes de temperatura. Debido a las bajas tasas de errores obtenidas se considera este método satisfactorio para emplearse en inspección. En (14) se realiza una revisión bibliográfica exhaustiva de los sistemas de planificación de la inspección para máquinas de medición coordinadas (CMM). Se muestra una clasificación atendiendo a las tecnologías utilizadas para asistir la planificación de las cuales cabe destacar la influencia de Sistemas Expertos, específicamente los basados en conocimientos (KB); las Redes Neuronales; los Algoritmos Genéticos y los sistemas basados en Lógica Fuzzy.

Según (15), la Inteligencia Artificial ofrece un gran potencial no solo para los procesos de inspección sino para el diseño ingenieril manufacturado en general.

1.2.4 Ruido en imágenes digitales

El ruido digital es conocido en la literatura como la variación aleatoria (que no se corresponde con la realidad) del brillo o el color en las imágenes digitales. En el tipo de ruido conocido como ruido sal y pimienta los píxeles de la imagen son muy diferentes en color o intensidad a los píxeles circundantes. El hecho que define este tipo de ruido es que el pixel ruidoso en cuestión no tiene relación alguna con los píxeles circundantes. Este tipo de ruido se adquiere principalmente en el proceso de captura. (16)

En el ruido de tipo gaussiano, todos y cada uno de los píxeles que componen la imagen cambian su valor, de acuerdo con una distribución normal o gaussiana. Se podrían aplicar otro tipo de distribuciones, pero la gaussiana o normal se toma como el modelo al que más se aproxima, debido al teorema central del límite, que dice que la suma de los diferentes ruidos tiende a aproximarse a una distribución normal o gaussiana.

El ruido de luminancia aparece como variaciones de luminosidad en la imagen, se nota sobre todo en superficies que deberían tener un color uniforme. Las imágenes digitales son susceptibles a una gran cantidad de ruidos, de ahí la importancia de emplear filtros para su eliminación en correspondencia con su tipo.

1.3 Componentes y elementos de un sistema de visión artificial.

Como plantea (17) La visión artificial es un procedimiento de adquisición de imágenes, sin contacto y mediante sistemas ópticos, donde se realiza un análisis automático de las imágenes. No se clasifica como una técnica óptica de medición, pues la visión artificial es el conjunto de los sistemas de formación de imágenes y el sistema de procesamiento de estas.

Los sistemas de visión artificial están conformados de cinco elementos básicos, como plantea (18). Iluminación, cámaras, tarjeta de captura de imagen, hardware de computadora y software.

1.3.1 Iluminación

Es el componente encargado de proveer las condiciones adecuadas de luz en la escena, para una correcta captura de las imágenes del objeto a inspeccionar, su objetivo es asistir en una mejor captura de las imágenes a procesar. Una inadecuada iluminación puede introducir defectos en las imágenes adquiridas y por consiguiente un mal procesamiento y resultado final.

El tipo de iluminación puede variar desde rayos X, pasando por el espectro ultravioleta, luz visible, e infrarrojo. El espectro infrarrojo es útil para el trabajo con imágenes de temperatura como en (13), o para el diagnóstico de objetos que trabajan específicamente en este rango de frecuencia como los filtros IR-CUT en (19). En (20) se pueden identificar más ejemplos incluso sobre otros espectros de trabajo.

El empleo de luminarias de bajo costo a la hora del montaje probablemente requerirán de ajustes para paliar cualquier inconveniente, como plantea (4). Una tendencia es el uso de luminaria LED (21),

tecnología que abarata los costos de montaje y mantenimiento, aunque (22) señala su sensibilidad al brillo; sin embargo (7) posee un criterio diferente.

1.3.2 Cámaras

Las cámaras son imprescindibles en los sistemas de inspección visual, pues pretenden simular la visión humana. Su objetivo es capturar la escena real y enviar en forma de imagen digital los datos para su análisis por la unidad de procesamiento.

Existen muchos tipos de cámaras que operan en diferentes espectros de visión, las más empleadas según (7), son las de tipo CCD (Charge Coupled Device) y las de tipo CMOS (Complementary Metal Oxide Semiconductor). El mismo autor indica las de tipo CCD como las más usadas por sus características, para la visión por computador, pero las de tipo CMOS por su rapidez, para inspección de las líneas de producción. (20) presenta una descripción del funcionamiento interno de las cámaras CCD. (23) muestra cómo se recomienda el uso de iluminación LED junto con cámaras CMOS para evitar errores de medición.

1.3.3 Unidad de procesamiento

Para la ejecución de los algoritmos seleccionados que procesarán las imágenes capturadas, se requiere de una unidad de procesamiento. En la mayoría de los casos se emplean CPUs (ejemplo: (2) (4) (7)).

Paralelizar los algoritmos es una de las opciones, de modo que se aproveche mejor las capacidades que incluyen los micro-procesadores modernos de incluir varios núcleos en un solo integrado, o de emplear varios ordenadores distribuyendo la carga de trabajo, como en (4). Esta alternativa tiene como objetivo mejorar los tiempos de ejecución de los algoritmos. Esta técnica adiciona un nivel mayor de complejidad de diseño en la confección del sistema de inspección, aunque muchas veces no logra establecer mejoras significativas, como se muestra en (24).

(3) (25) (26), proponen el empleo de FPGA para lograr tiempos de ejecución más reducidos en comparación con los microprocesadores. Las FPGA permiten ser programadas para ejecutar tareas complejas en pocos ciclos de reloj, lo que garantiza una velocidad de ejecución alta, como la que exigen un sistema de inspección visual en las cadenas de montajes. La forma en que están diseñadas las hace ideales para el procesamiento de imágenes y señales, aunque tienen sus límites en cuanto a los tipos de

algoritmos que se pueden ejecutar sobre ellas. Una variante es emplearlas de conjunto con un microprocesador, como en (26).

En la última década el empleo de la programación gráfica GPU para realización de cálculos ha ganado en popularidad, pues se emplea las potencialidades que poseen estas unidades para ejecutar tareas en paralelo. Un estudio que aprovecha dichas ventajas lo presentan (27), abordando la necesidad de obtener rápidos resultados con el aumento cada vez mayor de la calidad de las imágenes para diagnóstico.

La plataforma ARM es una variante de hardware prometedora para configuraciones de sistemas de inspección visual de bajo costo. Aunque no posee un gran potencial de cálculo, si lo es en portabilidad y prestaciones para inspecciones automáticas sencillas. Además, en los últimos años se ha experimentado un incremento en su desarrollo debido a la popularidad de los dispositivos móviles y el fenómeno de la Internet de las Cosas. Actualmente se realizan numerosos estudios con resultados positivos usando esta plataforma, específicamente sobre placas reducidas con la Raspberry Pi. Un ejemplo es (28), (29).

1.4 Ventajas y desventajas de los sistemas de visión artificial.

Estos sistemas, tienen como ventaja que obtienen las medidas sin entrar en contacto directo con el objeto, esto es de gran utilidad cuando el objeto a inspeccionar es de difícil acceso, ya sea por las condiciones del medio o por las características del objeto, un ejemplo se muestra en (30). Además de que puede ser integrado a un sistema de control del proceso de manufactura que vaya clasificando los objetos según sus parámetros de calidad.

Los elementos que lo integran son de fácil adquisición, principalmente una cámara, iluminación y una unidad de procesamiento. Con respecto a otros sistemas, como: pruebas de ultrasonido, pruebas radiográficas, mediante partículas magnéticas o con líquidos penetrantes, suele ser más económica de implementar.

Una de las desventajas de estos sistemas es que la selección de los algoritmos, se hacen apoyado en la experiencia de expertos, y el ajuste de sus parámetros se obtienen tras muchos ensayos. Estos algoritmos son sensibles a condiciones ambientales, dígame iluminación, calidad de las imágenes capturadas, ruido. (21) realiza una propuesta para abordar este problema. Otra desventaja es que, este tipo de prueba, solamente se realiza en la superficie visible.

1.5 Herramientas, librerías, lenguajes informáticos y metodología de desarrollo de software a utilizar.

1.5.1 Lenguaje informático

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas. (31)

1.5.2 Librerías

OpenCV Es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicaciones de control de procesos donde se requiere reconocimiento de objetos. Esto se debe a que su publicación se da bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.

Open CV es multiplataforma, existiendo versiones para GNU/Linux, Mac OS X y Windows. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estérea y visión robótica.

1.5.3 Metodología de desarrollo de software.

La metodología de desarrollo del software constituye un proceso donde se definen técnicas y procedimientos para llevar a cabo el desarrollo de software en el que se van indicando paso a paso todas las actividades a realizar para lograr el producto deseado. Además, indican los roles que deben participar en el desarrollo de las actividades. Por otra parte, detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

(32) Plantea que las metodologías de software tienen que ver con el proceso de creación de software, no tanto el aspecto técnico, pero sí los aspectos organizativos. Una metodología debe ser adaptable para poder aplicarse en distintos proyectos, sencilla para que no resulte muy engorrosa su utilización y a la vez sea completa como para que su utilización por parte de un equipo sea provechosa.

(33) establecen que las metodologías de desarrollo de software proveen una dirección a seguir para la correcta aplicación de los demás elementos (34) explica que las Metodologías de Desarrollo surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto de software.

Las metodologías de desarrollo se pueden encontrar en dos grandes grupos: metodologías tradicionales (o también denominadas Metodologías Pesadas) y metodologías ágiles.

Metodologías ágiles

La metodología de software ágil combina una filosofía y un conjunto de directrices de desarrollo. La filosofía busca la satisfacción del cliente y la entrega temprana de software. Las directrices de desarrollo resaltan la entrega sobre el análisis y diseño, y la comunicación activa y continua entre los desarrolladores y clientes.

Metodologías tradicionales

Llevan un control estricto del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Centran su atención en llevar una documentación exhaustiva de todo el proyecto y en cumplir con un plan de proyecto.

Las metodologías tradicionales establecen durante todo el proceso de desarrollo un mayor énfasis en la planificación y control del proyecto, y en la especificación precisa de los requisitos y el modelado. Por otro lado, las ágiles están más orientadas a la generación de código con ciclos muy cortos de desarrollo,

enfocándose a equipos de desarrollo pequeños, haciendo especial hincapié en aspectos humanos asociados al trabajo en equipo e involucrando activamente al cliente en el proceso.

Se seleccionan las metodologías ágiles porque se adecuan más para guiar el desarrollo del software propuesto, principalmente porque el proyecto no es altamente complejo, se cuenta con pocos roles y no se dispone de mucho tiempo para su producción, además permite adaptarse a las exigencias y condiciones del cliente en cuanto a ahorro de tiempo y entrega de la solución con calidad.

Metodología de desarrollo de software a utilizar

Scrum

Scrum es un método de desarrollo ágil de software concebido por Jeff Sutherland y su equipo de desarrollo. Los principios Scrum son congruentes con el manifiesto ágil y se utilizan para guiar actividades de desarrollo dentro de un proceso de análisis que incorpora las siguientes actividades estructurales: requerimientos, análisis, diseño, evolución y entrega. Dentro de cada actividad estructural, las tareas del trabajo ocurren con un patrón del proceso llamado sprint. El trabajo realizado dentro de un sprint (el número de éstos que requiere cada actividad estructural variará en función de la complejidad y tamaño del producto) se adapta al problema en cuestión y se define y con frecuencia se modifica en tiempo real por parte del equipo Scrum.

Sprint

Un Sprint es el procedimiento de adaptación de las cambiantes variables del entorno (requerimientos, tiempo, recursos, conocimiento, tecnología). Son ciclos iterativos en los cuales se desarrolla o mejora una funcionalidad para producir nuevos incrementos. Durante un Sprint el producto es diseñado, codificado y probado. Y su arquitectura y diseño evolucionan durante el desarrollo. Un Sprint tiene una duración planificada de entre una semana y un mes.

Artefactos de la metodología.

- **Product Backlog (Pila del Producto)**

El *Product Backlog* es un inventario que contiene cualquier tipo de trabajo que haya que hacer en el producto. Es una lista en cualquier formato que contiene todos los requerimientos que necesitamos implementar en el producto. Esta lista es el resultado del trabajo del *Product Owner* con los distintos *Stakeholders* de la organización, y refleja el estado real del trabajo pendiente de implementar en un producto.

La pila del producto no es un documento de requisitos, sino una herramienta de información para el equipo. Si se emplea formato de lista, la información mínima que se suele incluir para cada historia de usuario es: descripción de la funcionalidad/requisito, denominado "historia de usuario", prioridad, pre-estimación del esfuerzo necesario.

- ***Sprint Backlog (Pila del Sprint)***

El Sprint Backlog contiene todo el trabajo que hay que hacer durante un sprint. Este proporciona una visión del trabajo a realizar durante el Sprint actual. Está gestionado por el equipo de desarrollo, que se encarga de mantenerlo actualizado y transparente durante toda la iteración, especialmente a través de los *Daily Scrum*. La pila del sprint descompone las historias de usuario en unidades de tamaño adecuado para monitorizar el avance a diario, poder identificar riesgos y problemas.

- Incremento

El Incremento es la suma de todas las tareas, casos de uso, historias de usuarios y cualquier elemento que se haya desarrollado durante el Sprint y que será puesto a disposición del usuario final en forma de software.

Un incremento es el resultado del Sprint. Es una pieza de Software, acorde con los elementos seleccionados durante el *Sprint Planning* del *Sprint Backlog* que aporta un valor de negocio al producto que se está desarrollando.

Justificación para su uso en el proyecto:

Se ha decidido utilizar para el desarrollo de este proyecto como metodología de desarrollo Scrum ya que es una de las metodologías ágiles más populares y usadas en proyectos de software además de ser un marco de trabajo para el desarrollo y mantenimiento de productos complejos o sencillos, acepta el cambio de requisitos fácilmente y es muy sencillo de entender. Posibilita el cambio en el orden de prioridades en el caso que se requieran y se trabaja de forma individual.

1.5.4 Herramientas de modelado

Visual Paradigm for UML 8 es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Este software de modelado ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor

coste. Permite dibujar todos los tipos de diagramas de clases, generar código desde diagramas, generar documentación, así como la generación de código inverso.

Posee varias características importantes como son (multiplataforma, interoperabilidad, modelamiento de los requisitos, colaboración de equipo, generación de documentos, editor de detalles de casos de uso, ingeniería de código, modelado de procesos del negocio y modelamiento de bases de datos. (35)

1.6 Conclusiones parciales

Durante este capítulo se hace un estudio del estado del arte referente a los sistemas de inspección visual automatizada, el cual permitió la profundización de los conocimientos sobre el procesamiento digital de imágenes, la inteligencia artificial y ruido en imágenes digitales. También se investigó sobre cuáles son los componentes imprescindibles de un sistema de visión artificial, así como las ventajas y desventajas de estos sistemas y la correcta selección de las herramientas necesarias para su confección.

CAPÍTULO 2. SOLUCIÓN TEORICA DEL PROBLEMA CIENTIFICO.

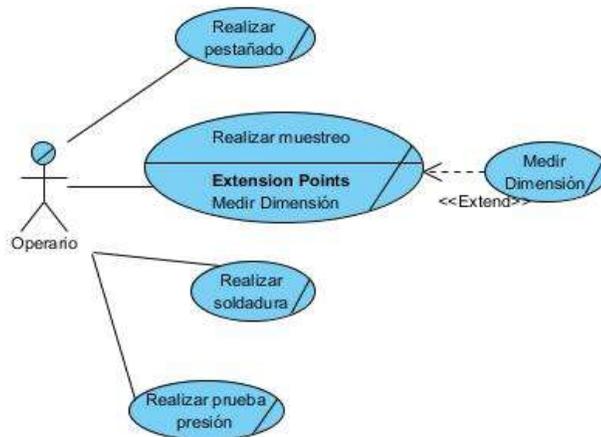
2.1 Introducción:

En el presente capítulo se realiza la descripción de la propuesta de solución basada en los principios y reglas de la metodología de desarrollo de software SCRUM. Se propone un sistema de monitoreo para el control de calidad durante el proceso de pestañado en cilindros de 10 kg y se describen las etapas del algoritmo implementado.

2.2 Elementos básicos de ingeniería de software:

2.2.1 Modelado del negocio

Ilustración 1: Modelado del negocio



2.2.2 Descripción de Actores y Casos de uso del Negocio

Ilustración 2: Descripción de actores

Actor del negocio	Descripción
<i>Operario</i>	El Operario es aquella persona que inicializa el negocio (Realizando el proceso de pestañado)

Casos de Uso del Negocio

Realizar pestañado

Realizar muestreo

Medir dimensión

Realizar soldadura

Realizar prueba presión

Descripción del Caso de Uso

Ilustración 3: Caso de uso realizar pestañado

Caso de uso del negocio	< Realizar pestañado >
Actores	<i>Operarios</i>
Resumen	<i><El Caso de Uso comienza cuando el Operario realiza el proceso de pestañado.></i>
Casos de uso asociados	
Acción del actor	Respuesta del proceso de negocio
<i>1. El Operario realiza el proceso de pestañado.</i>	
Otras secciones	
Cursos Alternos:	

Ilustración 4: Caso uso realizar muestreo

Caso de uso del negocio	<Realizar muestreo>
Actores	Operario
Resumen	<El Caso de Uso comienza cuando el operario revisa la pestaña y procede a realizar un muestreo. >
Casos de uso asociados	
Acción del actor	Respuesta del proceso de negocio
<ol style="list-style-type: none"> 1. Revisa la pestaña. 2. Realiza un muestreo de la pestaña. 	
Otras secciones	
Cursos Alternos:	

Ilustración 5: Caso de uso medir dimensión

Caso de uso del negocio	< Medir dimensión>
Actores	Operario
Resumen	<El Caso de Uso comienza cuando el operario mide las dimensiones del muestreo >
Casos de uso asociados	
Acción del actor	Respuesta del proceso de negocio
<ol style="list-style-type: none"> 1. Mide las dimensiones del muestreo. 	
Otras secciones	
Cursos Alternos:	
1.1 Si está mal se recicla	

Ilustración 6: Caso de uso realizar soldadura

Caso de uso del negocio	< Realizar soldadura >
<u>Actores</u>	<u>Operario</u>
Resumen	<El Caso de Uso comienza cuando el operario realiza el proceso de soldadura. >
<u>Casos de uso asociados</u>	
Acción del actor	Respuesta del proceso de negocio
1. El operario de realiza la soldadura.	
Otras secciones	
Cursos Alternos:	

Ilustración 7: Caso de uso realizar prueba de presión

Caso de uso del negocio	< Realizar prueba de presión >
<u>Actores</u>	<u>Operario</u>
Resumen	<El Caso de Uso comienza cuando el operario después del proceso de soldadura realiza la prueba de presión.>
<u>Casos de uso asociados</u>	
Acción del actor	Respuesta del proceso de negocio
1. Realiza pruebas de presión.	
Otras secciones	
Cursos Alternos:	
1.1 Si está mal se recicla	

2.2.3 Equipo de Trabajo y Roles

La base del éxito del desarrollo de un proyecto depende en gran medida, en el valor del equipo de trabajo que lo lleva a cabo. Los integrantes del equipo que realizaron esta investigación fueron:

Ilustración 8: Miembros y roles del equipo SCRUM

Miembros	Roles de Scrum
COMFORMAT	ProductOwner
MSc. Darien Menendez	Scrum Master
Est. Aylin Morales	Equipo de desarrollo, tester

A partir de entrevistas realizadas al cliente, donde se explica claramente cuáles son las principales necesidades del cliente, y la especificación de las reglas de negocio que se deben tener en cuenta para que el sistema se ajuste perfectamente a las solicitudes de la empresa, se obtuvieron los siguientes requerimientos funcionales iniciales:

- RF 1. Ajustar foto.
- RF 2 Calibrar foto.
- RF 3 Inspeccionar cilindro.

Para el sistema fueron capturados los siguientes requisitos no funcionales que son sumamente importantes para el diseño de la arquitectura a desarrollar:

- RNF 1: El sistema deberá ser utilizado por personas que no tengan un conocimiento básico en el manejo de las computadoras.
- RNF 2: La interfaz debe ser sencilla y fácil de usar.
- RNF 3: Los tiempos de respuesta deben ser mínimos.
- RNF 4: El sistema debe ser lo más exacto posible.

2.2.4 Pila del Producto (ProductBacklog)

La Pila del Producto, llamada también Lista de Historias de Usuarios, registra los requisitos vistos desde el punto de vista del cliente.

La Pila del Producto es la fuente principal de información sobre el producto donde son recopilados todos los requerimientos a realizar que son necesarias para el desarrollo del mismo. Siendo este un artefacto dinámico en constante cambio y evolución para identificar lo que el producto necesita para ser adecuado, competitivo y útil, se presenta a continuación la Pila del Producto final elaborada y ordenada por el ProductOwner.

Ilustración 9:Pila del Producto (ProductBacklog)

Pila del Producto			
Prioridad	Descripción	Estimados (Horas)	Sprint
1	Gestionar Clientes	4	1
2	Gestionar Recursos	8	
3	Preparación del laboratorio de prueba	4	
4	Simulación de proceso real	2	
5	Calcular las dimensiones del pestañado del cilindro	2	
6	Obtención de datos	24	
7	Clasificación de datos	4	
8	Confección de algoritmos (etapa 1)	40	2
9	Prueba de algoritmos (etapa1)	24	
10	Generar Gráfica (etapa 1)	1	

11	Análisis de datos obtenidos (etapa1)	1	
12	Confección de algoritmos (etapa 2)	40	3
13	Prueba de algoritmos (etapa2)	24	
14	Generar Gráfica (etapa2)	1	
15	Análisis de datos obtenidos (etapa2)	1	
16	Diseño de la interfaz de usuario	2	

2.2.5 Pila de Sprint (Sprint Backlog)

La pila del sprint refleja los requisitos vistos desde el punto de vista del equipo de desarrollo. Partiendo de la planificación de cada sprint se crea la Pila de Sprint que es una lista de trabajos seleccionados de la Pila del Producto que debe realizar el equipo durante el sprint, para generar el incremento previsto. Es una herramienta para la comunicación visual del equipo la cual permite monitorizar el avance a diario, e identificar riesgos y problemas sin necesidad de procesos de gestión complejos.

A continuación, se relaciona cada Sprint con las tareas realizadas durante el desarrollo de la propuesta.

Ilustración 10: Pila del Sprint (Sprint Backlog)

Iteración (Sprint)	Tareas de la Pila del Producto	Encargado	Fecha Inicial	Fecha Final	Horas estimadas totales
1	Gestionar Clientes	Aylin Morales	4/2/2019	11/2/2019	48
	Gestionar Recursos				
	Preparación del laboratorio de prueba				
	Simulación de proceso real				

	Calcular las dimensiones del pestañado del cilindro				
	Obtención de datos				
	Clasificación de datos				
2	Confección de algoritmos (etapa 1)	Aylin Morales	12/2/2019	25/2/2019	66
	Prueba de algoritmos (etapa 1)				
	Generar Gráfica (etapa 1)				
	Análisis de datos obtenidos (etapa 1)				
3	Confección de algoritmos (etapa 2)	Aylin Morales	26/2/2019	5/3/2019	68
	Prueba de algoritmos (etapa 2)				
	Generar Gráfica (etapa 2)				
	Análisis de datos obtenidos (etapa 2)				
	Diseño de la interfaz de usuario				

2.2.6 Planificación del Sprint.

La reunión de planificación de las tareas es conducida por el responsable del funcionamiento del marco Scrum (*Scrum Master*) a la que deben asistir el propietario del producto y el equipo completo, y a la que también pueden asistir otros implicados en el proyecto.

En la planificación del sprint se toma como base las prioridades y necesidades de negocio del cliente, y se determinan cuáles y cómo van a ser las funcionalidades que se incorporarán al producto en el siguiente sprint.

Esta etapa de planificación consta de dos partes:

Primera: El cliente muestra al equipo de trabajo la lista de requisitos del producto ordenados. El equipo revisa la lista y añade más funcionalidades, pero además pregunta al cliente las dudas que le surgen, el mismo escoge los objetivos de más prioridad en el ProductBacklog y aquí es cuando el equipo de desarrollo se compromete a entregar una parte del producto una vez que finalice la iteración(sprint). Cada requisito principal se convierte en una historia de usuario.

Segunda: El equipo planifica la iteración y elabora el modo de actuar que le permitirá conseguir el mejor resultado posible con el mínimo esfuerzo, en el menor tiempo posible. En esta actividad el equipo es el responsable de organizar su trabajo porque es quien mejor conoce cómo hacerlo. El equipo desglosa cada funcionalidad en tareas, y estima el esfuerzo para cada una de ellas en horas, descomponiendo así las tareas que forman la pila del sprint. En este desglose, el equipo tiene en cuenta los elementos de diseño y arquitectura que deberá incorporar el sistema.

El papel del propietario del producto es atender a dudas, comprobar que el equipo comprende y comparte su objetivo. El Scrum Master actúa de moderador de la reunión.

A continuación, se muestra la planificación de los Sprint de forma independiente y detallada, con todos los elementos de la pila del producto descompuesto en tareas.

Ilustración 11: Planificación del Sprint 1

Sprint 1				
Pila del Producto	Tareas	Encargado	Estimación inicial	Estimación Total

Gestionar Clientes	Entrevistar cliente	Aylin Morales	4	6 Días
	Anotar requisitos		0.50	
	Negociar el proyecto		0.50	
Gestionar Recursos	Conseguir materiales de la fábrica	Aylin Morales	2	
	Transportar los materiales		1	
Preparación del laboratorio de prueba	Acondicionar local	Aylin Morales	4	
	Montar la plataforma de soporte		5	
	Instalar los materiales de los cilindros		7	
	Montaje de la cámara		8	
	Montar las herramientas de medición		5	
	Montar la unidad de procesamiento		5	
Simulación de proceso real	Capturar las imágenes	Aylin Morales	2	
Calcular las dimensiones del pestañado del cilindro	Medir la de altura de pestañado con pie de rey	Aylin Morales	1	
	Medir la profundidad de pestañado con pie de rey		1	

Obtención de datos	Ingresar datos en hojas de cálculo	Aylin Morales	1	
Clasificación de datos	Comparar con la norma de calidad la altura obtenida.	Aylin Morales	1	
	Comparar con la norma de calidad la profundidad obtenida.			

Ilustración 12: Planificación del Sprint 2

Sprint 2				
Pila del Producto	Tareas	Encargado	Estimación inicial	Estimación Total
Confección de algoritmos (etapa 1)	Realizar estudio del estado del arte	Aylin Morales	20	8 Días
	Revisar literatura		20	
	Implementar algoritmos		20	
Prueba de algoritmos (etapa 1)	Correr programa	Aylin Morales	2	
Generar Gráfica (etapa 1)	Implementar métodos de la librería matplotlib	Aylin Morales	2	
Análisis de datos obtenidos (etapa 1)	Confrontar resultados obtenidos por el programa con las mediciones realizadas.	Aylin Morales	2	

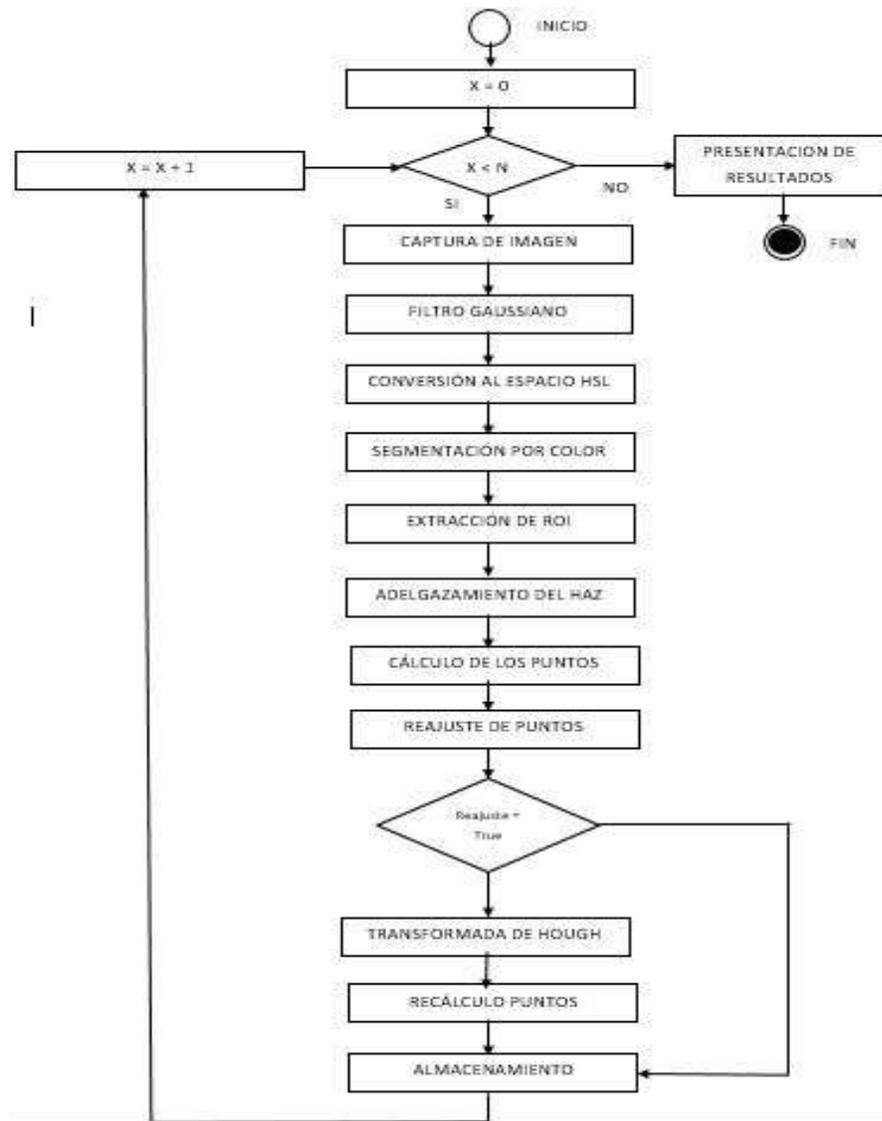
Ilustración 13: Planificación del Sprint 3

Sprint 3				
Pila del Producto	Tareas	Encargado	Estimación inicial	Estimación Total
Confección de algoritmos (etapa 2)	Revisar literatura	Aylin	30	8 Días
	Implementar algoritmos	Morales	30	
Prueba de algoritmos (etapa 2)	Correr programa	Aylin Morales	2	
Generar Gráfica (etapa 2)	Actualizar gráfica de la librería matplotlib	Aylin Morales	1	
Análisis de datos obtenidos (etapa 2)	Confrontar resultados obtenidos con las normas de calidad establecidas.	Aylin	1	
	Confrontar resultados obtenidos con los valores reales	Morales	1	
Diseño de la interfaz de usuario	Crear ventana	Aylin Morales	1	
	Implementar funcionalidades		2	

2.3 Algoritmo utilizado para la inspección visual automática

A continuación, se muestra un diagrama de bloques del algoritmo propuesto y como se relacionan cada una de sus etapas.

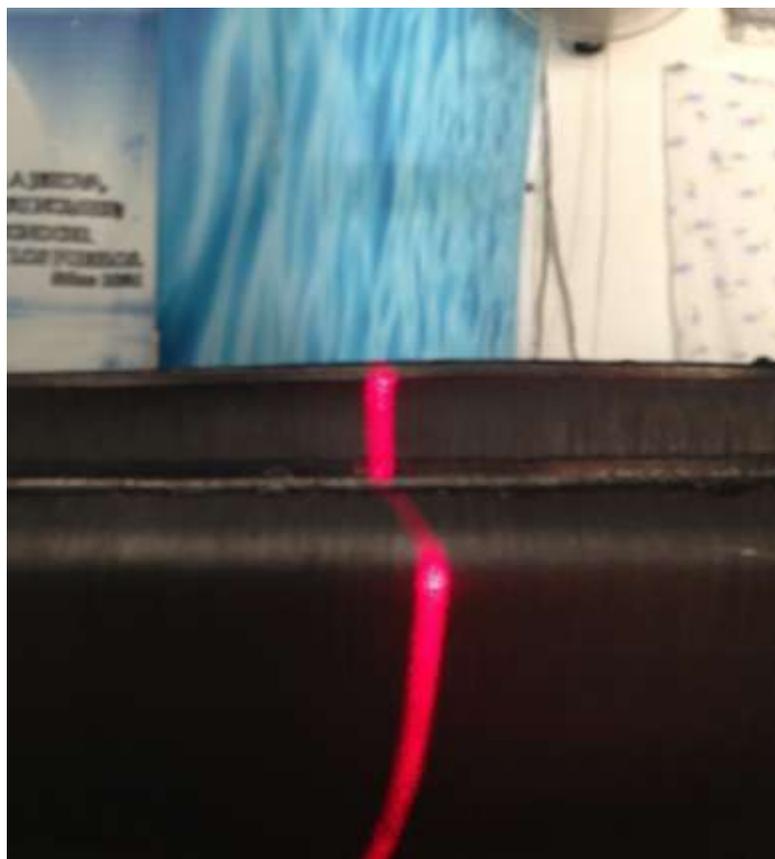
Ilustración 14: Diagrama de algoritmo



Captura de imagen

La unidad de procesamiento es conectada a la cámara. Las imágenes son capturadas mediante instrucciones a la cámara digital, y en formato RGB a diferencia de (36) y (6) que realizan sus propuestas empleando la escala de gris. Un ejemplo de captura se muestra en la Figura 15.

Ilustración 15: Captura de imagen



Filtro Gaussiano

Este procedimiento elimina ruidos no deseados en la imagen, procedentes del proceso de captura. La presencia de ruido en la imagen en muchas ocasiones afecta la detección de patrones. La operación queda definida matemáticamente por la ecuación, que representa una operación de convolución en el espacio de las frecuencias (37).

$$g(i, j) = \sum_{m=0}^M \sum_{n=0}^N f(i, j) h(i - m, j - n)$$

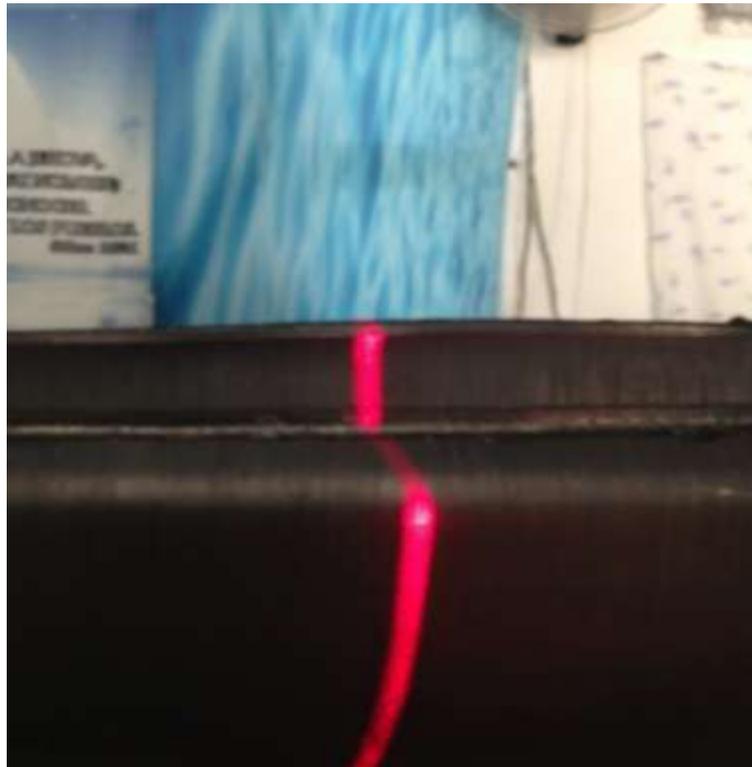
donde:

g -> imagen de salida

f -> imagen de entrada

h -> núcleo de convolución que responde a una distribución gaussiana de dimensiones 5 x 5.

Ilustración 16: Filtro gaussiano



Conversión al espacio HSL (Matiz-Saturación-Luminosidad)

Se procede a transformar la imagen en el modelo de colores HSL (Matiz, Saturación, Luminosidad). Este modelo de colores responde a una transformación no lineal del espacio de color RGB, y es más adecuado e intuitivo para la identificación de colores en imágenes, como plantea (16). Este método al trabajar con colores y no con valores de intensidad hace menos sensible el dispositivo a cambios de iluminación en la escena. La transformación de la imagen está definida por las ecuaciones (38).

Para: r, g, b -> valores de rojo, verde y azul de un pixel

$$M = \text{MAX}(r, g, b) \quad m = \text{MIN}(r, g, b)$$

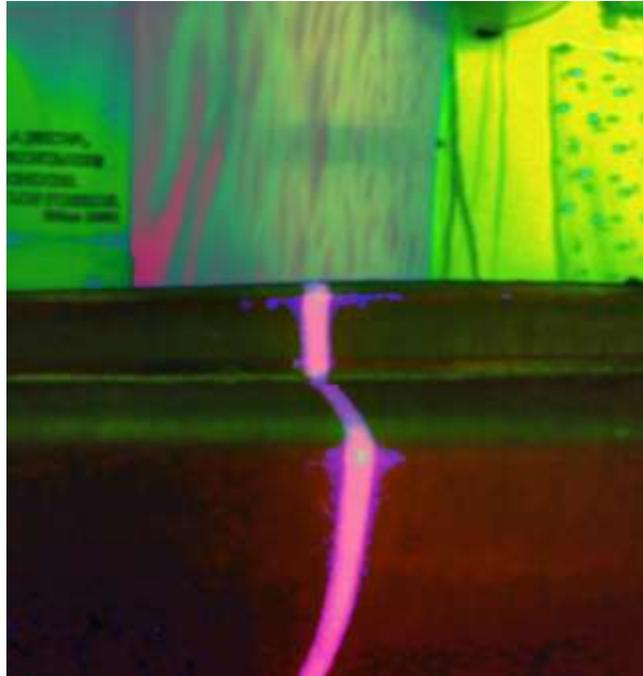
Matiz:

$$H = \begin{cases} 0, & \text{si } M = m \\ \left(60 \times \frac{g - b}{M - m} + 360 \right) \text{ mod } 360, & \text{si } M = r \\ 60 \times \frac{b - r}{M - m} + 120, & \text{si } M = g \\ 60 \times \frac{M - m}{r - g} + 240, & \text{si } M = b \end{cases}$$

$$\text{Saturación: } S = \begin{cases} \frac{0}{M - m}, & \text{si } M = m \\ \frac{M - m}{M + m}, & \text{si } L < 0.5 \\ \frac{M - m}{2 - (M + m)}, & \text{si } L \geq 0.5 \end{cases}$$

$$\text{Luminosidad: } L = \frac{1}{2}(M + m)$$

Ilustración 17: Espacio HSL



La figura 17 representa la transformación de la imagen capturada al espacio HSL. Una mejor identificación del haz láser se observa al analizar los componentes de matiz y luminosidad, los cuales serán usados por el algoritmo para la extracción de información de haz láser, por su color rojo y luz intensa.

Ilustración 18: Componente Luminosidad



Ilustración 19: Componente Matiz

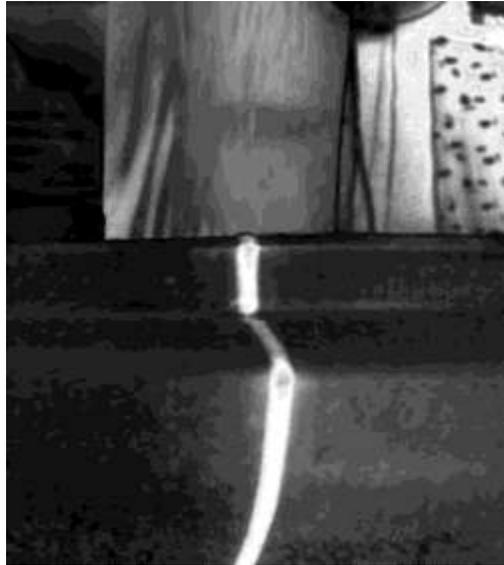
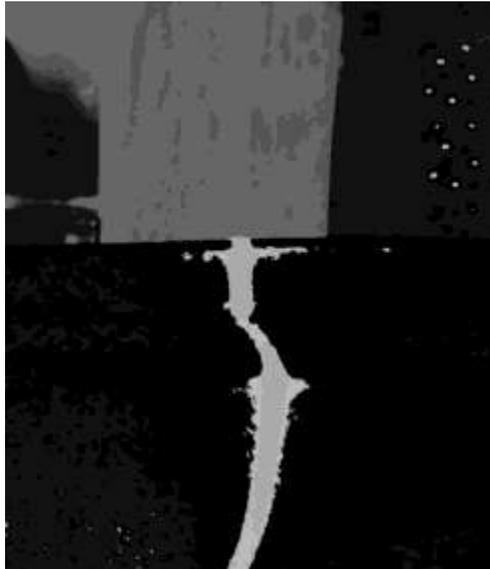


Ilustración 20: Componente Saturación



Segmentación por color

Esta técnica permite separar en la imagen, el objeto que se desea analizar del resto de la escena. El resto de los valores de la imagen son desechados al sustituirlo por el color negro como se representa en la ecuación. La Figura 22 muestra el resultado de aplicar una segmentación a la imagen anterior.

$$g(i, j) = \begin{cases} 0 & \text{si } f(i, j) < [220, 0, 80] \\ 1 & \text{si } f(i, j) \geq [220, 0, 80] \end{cases}$$

donde:

g(i, j) -> imagen de salida

f(i, j) -> imagen de entrada

[H, S, L] -> vector que representa un valor en el espacio HSL

Ilustración 21: Segmentación por color



Extracción de la región de interés (ROI)

Esta región comprende solamente la imagen en el área donde hay diferencia entre los valores de píxeles. De esta manera se evita hacer procesamiento innecesario y aumenta la rapidez del algoritmo. La imagen seleccionada es un subconjunto de la imagen segmentada, tal y como representa la ecuación.

$$g(m, n) = f(i, j) \quad \text{para } m = i, n = j, \quad i, j \in Q$$

$$m, n \in P \quad \text{donde } P \leq Q$$

Ilustración 22: Extracción de la región de interés



Adelgazamiento del haz láser

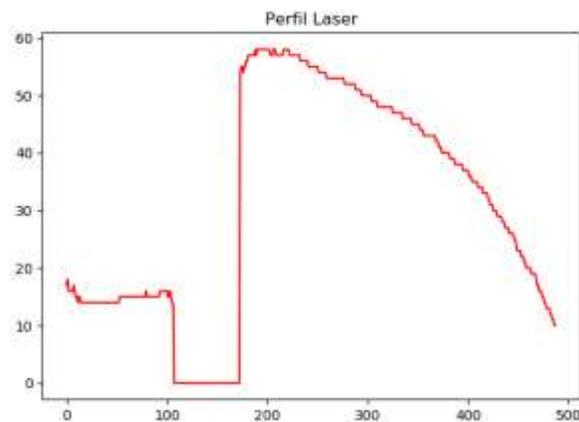
A pesar que el láser proyecta una luz intensa y concentrada su ancho todavía no es adecuado para detectar los puntos de interés. Se necesita hacer adelgazar con el ancho mínimo posible, se propone a 3 píxeles en la imagen para la detección de líneas en la etapa siguiente. El efecto de adelgazar el haz se obtiene al extraer los píxeles centrales en el haz y desechar el resto, la Figura 23 muestra el resultado final.

Ilustración 23: Adelgazamiento de láser



En esta etapa del algoritmo, es posible convertir los píxeles centrales en datos que representan el haz láser, creando así, un perfil del láser que puede ser usado como una función representativa. La figura 24 muestra cómo quedaría la función si se graficase.

Ilustración 24: Perfil láser



Identificación de puntos de interés

Una de las etapas cruciales para la medición es la correcta identificación de los puntos de interés. Estos puntos definen los extremos del alto y borde de la pestaña. Tales puntos se nombran como *A*, *B* y *C*, y los segmentos *AB* y *BC* determinan la altura y el ancho respectivamente.

El punto A, puede definirse matemáticamente por la siguiente ecuación y corresponde con el intercepto en eje de las y:

$$A = (0, f(0))$$

Para:

$x \in N$; f : función que representa el haz laser

El punto B: Puede definirse como el punto más alejado del punto A antes de llegar a C mientras que pertenezca a una recta de pendiente aproximadamente a 0 que incluya a ambos puntos.

El punto C: Puede definirse matemáticamente por la siguiente ecuación y corresponde con el punto máximo de la función que representa el haz láser.

$$C = MAX(x, f(x))$$

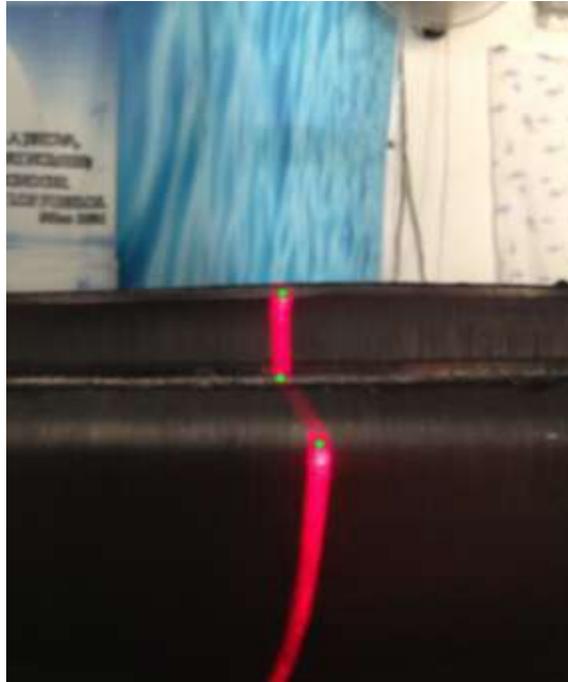
Para:

$x \in N$; f : función que representa el haz laser

MAX : función que determina el máximo; si para $(x, y) \in S$, para todo par $(a, b) \in$

S , entonces $y \geq b$

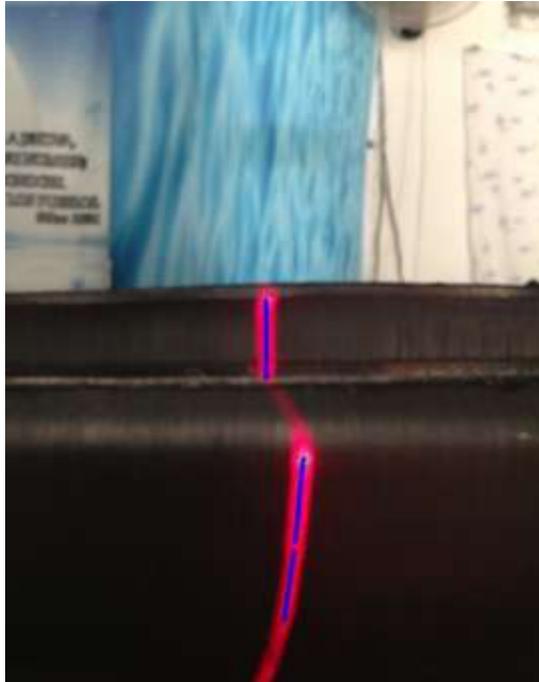
Ilustración 25: Identificación de los puntos de interés



Transformada de Hough

La transformada de Hough es empleada como método de corrección en la identificación de puntos. Este algoritmo permite identificar patrones de líneas en la imagen esqueletizada del láser. Al aplicar una transformada de Hough, similar a como plantea (36), se detectan rectas de la forma $y = mx + n$ o más bien de la forma $\rho = x * \cos \theta + y \sin \theta$ en coordenadas polares, espacio empleado por el algoritmo. Las líneas perpendiculares corresponden con el segmento de altura de pestaña y las oblicuas a la barriga de la tapa. Los puntos de tales líneas, coinciden también con los puntos de interés, como se puede ver en la figura 26.

Ilustración 26: Transformada de Hough



2.4 Software (Tutorial)

Para la implementación del algoritmo se utilizó el lenguaje de programación Python versión 3, esta elección se hizo basándose en que este lenguaje cuenta con una vasta disponibilidad de librerías para proyectos científicos y que es multiplataforma, posibilitando su ejecución en sistemas operativos Windows, Linux o Mac. La ventaja de ser multiplataforma es que beneficia su despliegue, independientemente de la tecnología que posea la empresa.

Se utilizó la librería OpenCV versión 3.2 para la implementación de los algoritmos de procesamiento digital de imágenes. Una de las ventajas de esta librería es que es de código abierto e implementada con gran eficiencia, debido a que hace uso de instrucciones nativas del procesador y la mayoría de sus métodos son acelerados mediante OpenCL, posibilitando obtener tiempos de ejecución inferiores a 1 nanosegundo de robustos algoritmos en modestos procesadores.

2.4.1 Interfaz principal

El diseño de la ventana principal de la aplicación corresponde con las necesidades del cliente al ser pequeña, sencilla, con 4 funcionalidades o botones, características que la hacen compatible con sistemas de pantallas para placas reducidas de 5 pulgadas y táctiles.

Ilustración 27: Interfaz principal



2.4.2 Funcionalidad Ajustar

El botón **Ajustar**, muestra una nueva ventana con el contenido capturado por la cámara. se podrá rectificar la correcta posición del dispositivo sobre la pestaña. Esta funcionalidad es imprescindible durante la etapa de instalación o reajuste del dispositivo. La funcionalidad termina cuando es centrado el dispositivo y cerrada la ventana de ajuste por el operario.

2.4.3 Funcionalidad Calibrar

El botón **Calibrar** ejecuta un proceso de calibración sobre una pestaña patrón. Este proceso consiste en ejecutar un conjunto de mediciones sobre una pestaña patrón. El proceso combina el resultado de cada una de las mediciones y es comparada con el valor real de la pestaña patrón, la cual es conocida anteriormente y configurada en el software. De esta manera se calibran las mediciones obtenidas

empleando la cámara durante este proceso, con los valores reales de la pestaña. Los parámetros obtenidos por este proceso serán empleados en las posteriores mediciones.

2.4.4 Funcionalidad Inspeccionar

El botón **Inspeccionar**, ejecuta el proceso de monitoreo a los parámetros de la pestaña insertado frente a la cámara del dispositivo de monitoreo. Durante este proceso se ejecutan 20 mediciones a lo largo del cordón, y se verifica que esté sobre los límites permitidos según el estándar de calidad requerido por la industria. La cantidad de mediciones que se llevan a cabo durante este proceso son configuradas en el software vía archivo de configuración, así también como los límites permitidos en los parámetros de pestañado. Al finalizar el proceso de inspección la información es mostrada en gráficas. En este momento el operario solo tendrá que cerrar la ventana mostrada.

2.4.5 Funcionalidad Salir

El botón **Salir** finaliza con la ejecución de la aplicación. Esta opción permite facilitar el acceso a dicha funcionalidad en pantallas táctiles pequeñas, como las seleccionadas para integrar con el dispositivo de monitoreo que ejecuta el software.

2.5 Conclusiones parciales

Durante este capítulo se seleccionaron dispositivos de bajo costo y fácil obtención para la construcción del sistema de monitoreo propuesto. Se seleccionaron las técnicas de procesamiento digital de imágenes y visión artificial para la extracción y monitoreo de los parámetros resultantes de proceso de pestañado. Se diseñó un software con una interfaz de usuario con el fin de ser usada por operarios que no siempre cuentan con altos niveles de capacitación o experiencia.

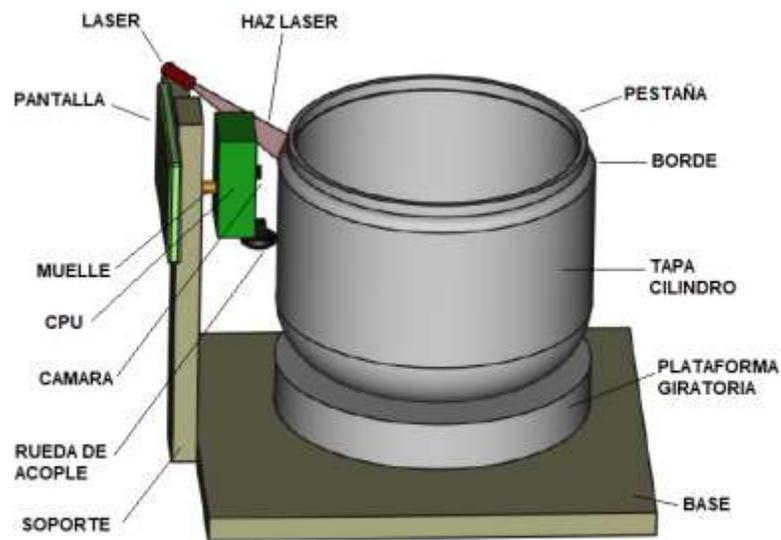
CAPÍTULO 3 VALIDACIÓN DEL SISTEMA DE MONITOREO PROPUESTO.

3.1 Introducción

En este capítulo se presenta el esquema de instalación del dispositivo de monitoreo, el análisis de costo según la metodología empleada. Se muestra el resultado de ejecutar el proceso haciendo uso de la propuesta sobre una tapa de cilindro defectuosa, comparándola con los datos obtenidos del proceso de medición manual.

3.2 Esquema de la instalación experimental

Ilustración 28: Prototipo de dispositivo



La figura 28 muestra cada una de las partes que componen el prototipo del dispositivo de monitoreo propuesto, las cuales son: base, plataforma giratoria, soporte, rueda de acople, unidad de procesamiento,

pantalla, muelle de ajuste y laser. Este dispositivo permite realizar la inspección a la pestaña de la tapa de cilindro de una manera práctica y sencilla. Se coloca la tapa de cilindro sobre la superficie giratoria y se ajusta junto a la rueda de acople. Al situar la tapa de cilindro de esta manera, el muelle ubicado después de la unidad de procesamiento hace presión manteniendo siempre la misma distancia entre la cámara y el cilindro. La cámara captura la imagen que es generada al incidir el haz de luz láser sobre la pestaña. La imagen es procesada por la CPU y los resultados son mostrados en una interfaz gráfica que muestra la pantalla.

La distancia entre la cámara y la superficie de la pestaña es de 10 cm. Esta distancia seleccionada obtiene una captura con adecuado enfoque y detalles de la escena. El ángulo entre la cámara y la luz estructurada emitida por el dispositivo láser es de 60° a diferencia del propuesto por (6), pues un mayor ángulo produce una mayor desviación del láser al entrar en contacto con la pestaña y por consiguiente una mejor identificación de la curva y mayor precisión en el monitoreo de la pestaña como indica (39).

3.3 Análisis de costo.

3.3.1 Estimación de costo del proyecto

Principales Conceptos en el Método por Puntos de Función:

- EI: Procesos en los que se introducen datos y que suponen la actualización de cualquier interno.
- EO: Procesos en los que se envía datos al exterior de la aplicación.
- EQ: Procesos consistentes en la combinación de una entrada y una salida en el que la entrada no produce ningún cambio en ningún archivo y la salida no contiene información derivada.
- ILF: Grupos de datos relacionados entre sí, internos al sistema.
- EIF: Grupos de datos que se mantienen externamente.
- PFTe: Total Puntos de Función para las entradas del sistema.
- PFTo: Total Puntos de Función para las salidas del sistema.

- PFTq: Total Puntos de Función para las consultas del sistema.
- PFTif: Total Puntos de Función para los archivos internos del sistema.
- PFTef: Total Puntos de Función para los archivos externos del sistema

Tablas de Valores:

E I- E Q

CLASIFICACION DE SALIDAS	1-5 Atributos	6-19 Atributos	Más de 19 Atributos
0 o 1 ficheros accedidos	BAJA 4	BAJA 4	MEDIA 5
2 o 3 ficheros accedidos	BAJA 4	MEDIA 5	ALTA 7
Más de 3 ficheros accedidos	MEDIA 5	ALTA 7	ALTA 7

Tabla 2

E O

CLASIFICACION DE ENTRADAS Y CONSULTAS	1-4 Atributos	5-15 Atributos	Más de 15 Atributos
0 o 1 ficheros accedidos	BAJA 3	BAJA 3	MEDIA 4
2 ficheros accedidos	BAJA 3	MEDIA 4	ALTA 6
Más de 2 ficheros accedidos	MEDIA 4	ALTA 6	ALTA 6

Tabla 1

IL F

FICHEROS LÓGICOS INTERNOS	1-19 Atributos	20-50 Atributos	Más de 50 Atributos
1 Entidad o registro lógico	BAJA 7	BAJA 7	MEDIA 10
2 - 5 Entidades o registros lógicos	BAJA 7	MEDIA 10	ALTA 15
Más de 5 Entidades o registros lógicos	MEDIA 10	ALTA 15	ALTA 15

Tabla 3

E I F

FICHEROS LÓGICOS EXTERNOS	1-19 Atributos	20-50 Atributos	Más de 50 Atributos
1 Entidad o registro lógico	BAJA 5	BAJA 5	MEDIA 7
2 - 5 Entidades o registros lógicos	BAJA 5	MEDIA 7	ALTA 10
Más de 5 Entidades o registros lógicos	MEDIA 7	ALTA 10	ALTA 10

Tabla 4

La información que resulte de estas tablas se pondera en una tabla general la cual describe el componente y el nivel en el que se encuentra. Determinando el peso de cada componente :

Componente	Bajo	Medio	Alto	Total
E I	$3 * 3 = 9$	$2 * 4 = 8$	$6 * 6 = 36$	PFTe= 8
E O	$2 * 4 = 8$	$0m * 5 = 0$	$0a * 7 = 0$	PFTo= 12
E Q	$2 * 3 = 6$	$4 * 4 = 16$	$0a * 6 = 0$	PFTq= 6
ILF	$2 * 7 = 14$	$IFm * 10 = 0$	$IFa * 15 = 0$	PFTif= 14
E I F	$1 * 5 = 5$	$EFm * 7 = 0$	$EFa * 10 = 0$	PFTef= 5
				PFS A = 45

$$PFS A = PFTe + PFTo + PFTq + PFTif + PFTef$$

$$PFS A = 8+12+6+14+5$$

P F S A = 45

Luego de obtener los puntos de función sin ajustar, debemos calificar cada uno de los factores de valor de ajuste, utilizando una escala del 0 al 5 con el siguiente desglose:

0 → sin influencia

1 → influencia incidental

2 → influencia moderada

3 → influencia media

4 → influencia significativa

5 → fuertes influencian en toda la aplicación

Calificamos cada uno de los 14 elementos y sumamos los grados de influencia (T D I) para obtener el factor de complejidad técnica (F C T):

Elementos	T D I
1. Comunicación de datos	5
2. Proceso distribuido de datos	2
3. Desempeño	5
4. Configuración	3
5. Volumen de transacciones	4
6. Captura de datos en Línea	3
7. Eficiencia al usuario final	4

8. Actualización de Datos en Línea	2
9. Complejidad	4
10. Reusabilidad	5
11. Facilidad de instalación	4
12. Facilidad de operación	4
13. Instalación Múltiple	3
14. Facilidad de cambio	4
FCT	52

Puntos de Función Ajustados (PFA)

$$PFA = PFSA * [0,65 + (0,01 * FCT)]$$

$$PFA = 45 * [0.65 + (0.01 * 52)]$$

$$PFA = 45 * [0.65 + 0.52]$$

$$PFA = 45 * 1.17$$

$$PFA = 52.65$$

Líneas de código (LC)

$$LC = PFA * (Líneas * PF)$$

$$LC = 52.65 * 100$$

$$LC = 5265$$

Esfuerzo hora/persona

$$E = PFA / (1/8 \text{ persona/hora})$$

$$E = PFA / (1/5) \quad 1 \text{ persona trabaja } 5h$$

$$E = 52.65 / 0.2$$

$E = 263.25$ horas/persona

Tomando 24 días laborables en el mes y 8 horas productivas al día, obtenemos 192 horas laborables al mes.

Duración del proyecto en meses

$263.25 \text{ horas/persona} / 1 \text{ persona} = 263.25$ horas

$DM = 263.25 \text{ horas} / 192 \text{ horas/mes}$

$DM = 1.37$ aproximadamente 1.5 meses + 1 mes y 1 semana por el margen de error de la estimación por puntos de función y 1 mes para la fase de prueba

$DM =$ aproximadamente 3.6 meses

Costo total del proyecto

$CT = \text{sueldo de 1 persona} / \text{cant de personas} * DM$

$CT = (500/1) * 3.6$

$CT = 1800$

3.3.2 Coste computacional.

Pasos	E t a p a	T i e m p o (s)
1	Captura	46 m s
2	Eliminar ruido	< 1 n s
3	Convertir a H S L	< 1 n s
4	Segmentación	< 1 n s
5	Extracción de R O I	< 1 n s
6	Esqueletizado	95 m s
7	Identificación de puntos	< 1 n s

8	Cálculo de dimensiones	< 1 ns
9	Transformada Hough	< 1 ns
10	Reajuste de puntos	< 1 ns
	Total	141 ms

La tabla anterior muestra los tiempos de ejecución de las etapas del algoritmo propuesto. En la mayoría de los casos, los tiempos de ejecución son sumamente pequeños, menor que 1 nanosegundo. Esto evidencia la eficiencia de la implementación. El algoritmo en su totalidad consume un tiempo menor de 150 ms, empleando un procesador Intel i3-5015U a una velocidad de 2.10GHz. Este tiempo es suficiente para realizar 7 mediciones en 1 segundo y aproximadamente 20 veces más rápido que un operario entrenado.

3.4 Caso de estudio.

Para comprobar el algoritmo propuesto se realizó el monitoreo a los parámetros de pestañado de varios cilindros de 10Kg. Los datos resultantes mostrados a continuación pertenecen específicamente a un botellón defectuoso desechado por la fábrica. Las etapas de ajuste y calibración se mostrarán en el ejemplo a continuación, pero en un entorno real solamente serán realizadas por el ingeniero a cargo del dispositivo de monitoreo.

3.4.1 Preparación

Se sitúa el cilindro patrón sobre la plataforma giratoria. Se selecciona la opción *Ajuste* de la pantalla principal, se muestra la ventana de ajuste y se procede a centrar el dispositivo de monitoreo sobre la pestaña. Una vez confirmada la correcta posición se cierra la ventana.

El proceso de calibración requiere previamente de la correcta entrada de los datos de alto y ancho de una pestaña modelo, cuyas dimensiones se conocen previamente. Se introducen en la configuración del software para la fábrica. Se selecciona la opción *Calibración* de la pantalla principal.

3.4.2 Monitoreo

Se inicia el proceso de rotación de la plataforma giratoria que contiene el cilindro y se selecciona la opción *Inspeccionar* de la pantalla principal. Luego se muestran por pantalla el resultado del proceso de inspección.

Al ejecutar esta función, se muestra una ventana de monitoreo, donde se observa un video con el desplazamiento del dispositivo sobre la pestaña. Al finalizar las mediciones se muestra la información en forma de gráficas para su fácil comprensión por parte del operario (Ver figura 29 y 30).

Ilustración 29: Ventana de inspección con información sobre altura

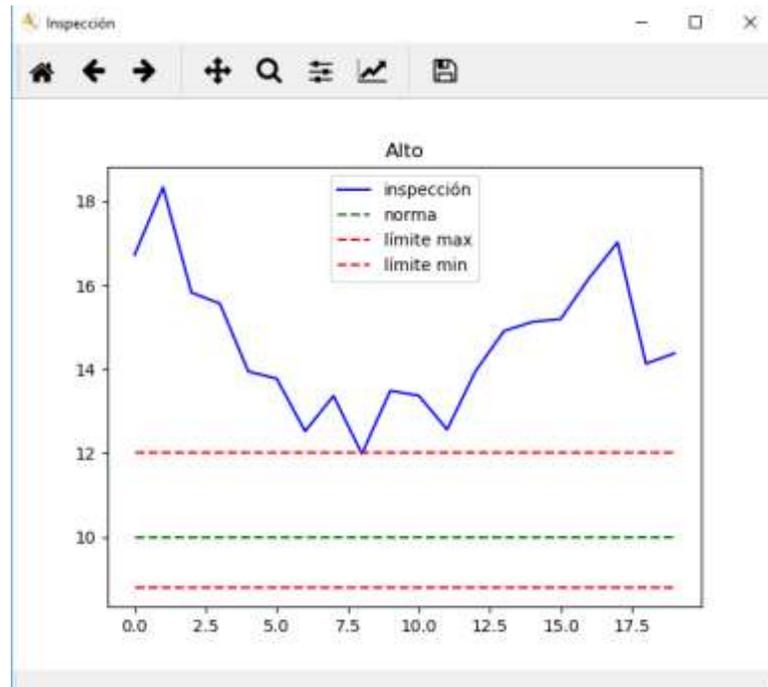
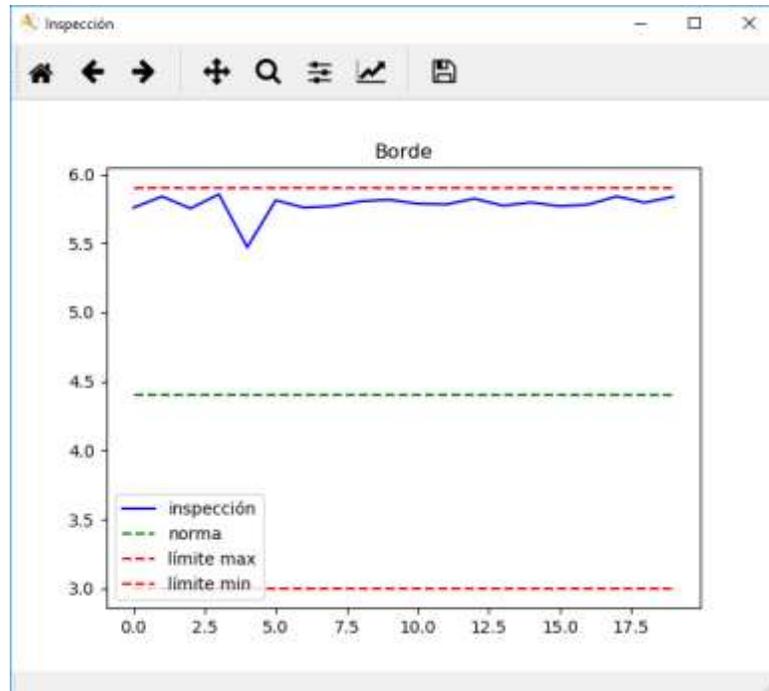


Ilustración 30: Ventana de inspección con información sobre el ancho



En las figuras 28 y 29 se muestra el resultado de 20 mediciones a los parámetros de alto y profundidad de la pestaña durante el proceso de inspección. La norma (40) establece, que la altura de la pestaña debe ser de 10mm, tomándose como altura mínima 8 y máxima 12; en cuanto a la profundidad se establece en 4,5mm, y el rango aceptable entre 6 y 3. Como se evidencia por los resultados mostrados gráficamente durante la inspección a la pestaña, la dimensión de la altura sobrepasa el límite máximo permitido, coincidiendo con los valores que clasificaron el cilindro como defectuoso. La profundidad calculada por el algoritmo se encuentra entre los límites. Los valores obtenidos por el sistema concuerdan con las mediciones que manualmente se realizaron con un pie de rey de 0,05mm de precisión. La diferencia entre las mediciones obtenidas entre el método tradicional y el automático es de tan solo 0.16mm, aproximadamente 3 veces menos la precisión del pie de rey, pero el 1.6% de la medición real, un valor muy pequeño de error, y suficiente para confiar en el método propuesto. La siguiente tabla muestra los valores obtenidos por cada uno de los métodos.

	Altura Real (mm)	Profundidad Real (mm)	Altura Programa (mm)	Profundidad Programa (mm)
1	16.8	5.8	16.72	5.75
2	18.3	5.8	18.12	5.75
3	15.9	5.8	16.19	5.75
4	15.5	5.8	15.11	5.75
5	14.8	5.8	13.94	5.75
6	13.8	5.8	14.17	5.75
7	12.5	5.8	12.51	5.75
8	13.4	5.8	13.36	5.75
9	12	5.8	11.97	5.75
10	13.5	5.8	13.48	5.75

11	13.4	5,8	13.26	5.75
12	12.6	5,8	12.83	5.75
13	13.9	5,8	13.94	5.75
14	14.9	5,8	14.91	5.75
15	15.1	5,8	14.9	5.75
16	15.2	5,8	14.96	5.75
17	16.2	5,8	16.17	5.75
18	17	5,8	17.02	5.75
19	14.16	5,8	14.13	5.75
20	14.4	5,8	14.37	5.75

3.5 Conclusiones parciales

En este capítulo se demostró que es posible construir un sistema de inspección visual automatizado, de fácil manipulación para ser insertado en una línea de producción, utilizando componentes de bajo costo, que pueda simplificar el trabajo de los operarios y a una mayor velocidad de lo que estos lo pueden realizar.

CONCLUSIONES

Tras los resultados obtenidos con el presente trabajo, se puede arribar a las siguientes conclusiones:

La inspección visual automatizada es de gran importancia para la industria moderna, debido a que permite la revisión de la calidad de los productos en una línea de montaje.

El empleo de técnicas de procesamiento digital de imágenes, visión artificial e inteligencia artificial permite crear un algoritmo capaz de controlar la calidad de los parámetros de pestañado en cilindros de 10 kg. Este sistema contribuye a una aceleración del proceso de producción, esto se debe a que el sistema es capaz de realizar las mediciones a una velocidad mayor que la de un operario. También disminuye la cantidad de cilindros defectuosos al final de la línea de producción, puesto que permite la inspección en tiempo real y detectar tempranamente defectos durante el proceso de pestañado.

Se demuestra la factibilidad de confeccionar un dispositivo sencillo de monitoreo empleando bajos recursos por la fábrica. Su implementación puede ser ejecutada empleando componentes de bajos costos.

RECOMENDACIONES

Realizar una prueba más exhaustiva del dispositivo prototipo y poner a prueba el algoritmo en todas las situaciones posibles, con el fin de garantizar su funcionamiento bajo cualquier circunstancia del entorno de la fábrica.

BIBLIOGRAFÍA

1. WANG, Y., CUI, Y. and HUANG, G. Q. *Study on Fruit Quality Inspection Based on Its Surface Color in Produce Logistics. International Conference on Manufacturing Automation. IEEE Computer Society.* 2010.
2. NASHAT, S., et al. Support vector machine approach to real-time inspection of biscuits on moving conveyor belt. *Computers and Electronics in Agriculture.* 2011, 75, pp. 147–158.
3. HOCENSKI, Z., ALEKSI, I. and MIJAKOVIC, R. *Ceramic Tiles Failure Detection Based on FPGA Image Processing.* Seoul : IEEE International Symposium on Industrial Electronics (ISIE), 2009.
4. ADAMO, F., et al. A low-cost inspection system for online defects assessment in satin glass. *Measurement.* 2009. 42, pp. 1304–1311.
5. CHETVERIKOV, D. and HANBURY, A. Finding defects in texture using regularity and local orientation. 2002. 35, pp. 2165-2180.
6. XU, M., ZHAO, M. and ZOU, Y. On-Line Visual Inspection System for Backside Weld of Tailored Blanks Laser Welding. *IEEE.* 2010. pp. 525-529.
7. ZHANG, H. and LI, D. Applications of computer vision techniques to cotton foreign matter inspection: A review. 109 *Computers and Electronics in Agriculture.* 2014. pp. 59–70.
8. MERA, C., et al. Automatic visual inspection: An approach with multi-instance learning. *Computers in Industry,* 83. 2016. pp. 46-54.
9. HU, G. H. Optimal ring Gabor filter design for texture defect detection using a simulated annealing algorithm. *International Conference on Information Science, Electronics and Electrical Engineering (ISEEE).* 2014. 2, pp. 860-864.
10. TONG, L., WONG, W. K. and KWONG, C. K. Differential evolution-based optimal Gabor filter model for fabric inspection. *Neurocomputing.* 2016. pp. 1386–1401.
11. HU, G. H., WANG, Q. H and & ZHANG, G. H. Unsupervised defect detection in textiles based on Fourier analysis and wavelet shrinkage. *Appl. Opt.* 54 2015. pp. 2963-2980.
12. LIN, J.-J., LIN, C.-H. and TSAI, I.-S. Applying Expert System and Fuzzy Logic to an Intelligent Diagnosis System for Fabric Inspection. *Textile Research Journal,* 65. 1995. pp. 697-709.
13. TRÉTOUT, H., et al. An Evaluation of Artificial Neural Networks Applied to Infrared Thermography Inspection of Composite Aerospace Structures. *In: THOMPSON, D. O. & CHIMENTI, D. E. (eds.).* 1995.
14. ANAGNOSTAKIS, D., et al. Knowledge Capture in CMM Inspection Planning: Barriers and Challenges. s.l. : *Procedia CIRP,* 2016. 52, pp. 216-221.
15. FAN, J. Applications of artificial intelligence in fabric and garment engineering. *Engineering Apparel Fabrics and Garments.* s.l. : Woodhead Publishing., 2009. 14.

16. **GONZALEZ, R. C. and WOODS, R. E.** Digital Image Processing, New Jersey, Pearson Education Inc. 2008.
17. **Ruiz Cabeza, H.** Visión Artificial Aplicada a la Industria. 2011.
18. **BROSNAN, T. and SUN, D.-W.** Improving quality inspection of food products by computer vision—a review. 2004. 61, pp. 3-16.
19. **LIU, Y. and YU, F.** Automatic inspection system of surface defects on optical IR-CUT filter based on machine vision. *Optics and Lasers in Engineering*, 55. 2014. pp. 243–257.
20. **GONZALEZ, R. C. and WOODS, R. E.** Digital Image Processing. New Jersey : Pearson Education Inc., 2008.
21. **DILEO, G., et al.** A vision system for the online quality monitoring of industrial manufacturing. s.l. : Optics and Lasers in Engineering, 89, 2017. pp. 162-168.
22. **SILVA, D.** Estudio de viabilidad de un sistema basado en Raspberry Pi para aplicaciones de Inspección Industrial por Visión Artificial. *Máster en Ingeniería de Automatización e Informática Industrial, Universidad de Oviedo*. 2015.
23. **PRIBULA, O., JANOSEK, M. and FISCHER, J.** Optical Position Sensor Based on Digital Image Processing: Magnetic Field Mapping Improvement. *RADIOENGINEERING*, 20. 2011. pp. 55-60.
24. **SUN, D.** Inspecting pizza topping percentage and distribution by a computer vision method. *Journal of Food Engineering*. 2000. Vol. 55, pp. 245–249.
25. **HAJIMOWLANA, S. H., et al.** Efficient Pre-processing Algorithms for an FPGA based In-Camera Video-Stream Processing System for Industry Inspection. s.l. : IEEE CCECE, 1997. pp. 835-838.
26. **GOLAMHOSSEINI, H. and HU, S.** A High Speed Vision System for Robots Using FPGA Technology. Auckland, New-Zealand : s.n., 2008.
27. **SMISTAD, E., et al.** Medical image segmentation on GPUs – A comprehensive review. *Medical Image Analysis*, 20. 2015. pp. 1-18.
28. **HUAMANTINCO CISNEROS, M. A., et al.** Beach carrying capacity assessment through image processing tools for coastal management. *Ocean & Coastal Management*, 130 2016. pp. 138-147.
29. **AZAEZ, S., et al.** Towards Flexibility in Future Industrial Manufacturing: A Global Framework for Sel. 2016.
30. **SZYDLOWSKIA, M., et al.** Machine vision micro-milling tool wear inspection by image reconstruction and light reflectance. *Precision Engineering*. 2016.
31. **Rossum, Guido van.** *El tutorial de Python*. 2009.
32. **Phillips, Andrew.** 2010.
33. **Amaro and Valdeverde.** 2007.

34. C coica. 2006.
35. U M L, V isual Paradigm for. 2019.
36. L I, Y., et al. On-line V isual M easurement and Inspection of W eld Bead U sing Structured Light. *IEEE Instrumentation and Measurement Technology Conference*. M ay 12-15, 2008. pp. 2038-2043.
37. A CHARYA, T and RAY, A . K. Image processing: principles and applications, John Wiley & Sons. 2005.
38. O PENC V. O pen Source Com puter V ision [O nline]. 2018.
39. M enendez, D . *MONITOREO DE LAS DIMENSIONES DEL CORDÓN DE SOLDADURA MEDIANTE PROCESAMIENTO DE IMÁGENES DIGITALES*. 2019.
40. C O V E N I N, N C 364:2004. 649:97.
41. H O C E N S K I, Z., A L E K S I, I. and M I J A K O V I C, R. Ceramic Tiles Failure Detection Based on FPGA Image Processing. *IEEE International Symposium on Industrial Electronics (ISIE)*. Seoul, Korea : s.n., 2009.
42. L E C C E S E, F. C., et al. A smart city application: A fully controlled street lighting isle based on Raspberry -Pi card, a ZigBee sensor network and W iM A X. 2014. Vols. *Sensors*, 14, pp. 24408-24424.
43. L I, Y., et al. On-line V isual M easurement and Inspection of W eld Bead U sing Structured Light. 12 -15 M ay 2008 2008 *IEEE Instrumentation and Measurement Technology Conference*. pp. 2038-2043.
44. M A K S I M O V I C, M . and A L., E T. Raspberry Pi as Internet of things hardware: performances and constraints. 3, 8. 2014.
45. M A L M I R, S and S H A L C H I A N, M . Design and FPGA implem entation of dual-stage lane detection, based on Hough transform and localized stripe features. *Microprocessors and Microsystems*, 64. 2019. pp. 12-22.
46. N A S H A T, S., et al. Support vector machine approach to real-time inspection of biscuits on moving conveyor belt. . *Computers and Electronics in Agriculture*. 2011. 75, pp. 147–158.
47. [O nline]. O PENC V. 2018. O pen Source Com puter V ision .
48. S H E N, J., et al. X-ray inspection of TSV defects with self-organizing map network and O tsu algorithm . *Microelectronics Reliability*, 67. 2016. pp. 129–134.
49. V U J O V I C, V . M . and M I R J A N A . Raspberry Pi as a Sensor Web node for home automation. 44 *Computers & Electrical Engineering*. 2015. pp. 153-171.
50. W A N G, Y., C U I, Y. and H U A N G, G . Q. Study on Fruit Quality Inspection Based on Its Surface Color in Produce Logistics. *International Conference on Manufacturing Automation. IEEE Computer Society*. 2010.

51. WEIMER, D., SCHOLZ-REITER, B. and SHPITALNI, M. Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. CIRP Annals - Manufacturing Technology. 2016.
52. AGRAWAL, N. S. and SMITA. Smart drip irrigation system using raspberry pi and arduino. 2015.
53. ANAGNOSTAKIS, D., et al. Knowledge Capture in CMM Inspection Planning: Barriers and Challenges. s.l. : Procedia CIRP, 52,, 2016. pp. 216-221.
54. SZYDŁOWSKIA, M., et al. Machine vision micro-milling tool wear inspection by image reconstruction and light reflectance. Precision Engineering. 2016.