

# DESARROLLO DE HERRAMIENTAS DE VISIÓN ARTIFICIAL EN MATLAB PARA INSPECCIÓN DE TARJETAS PCB

***Hugo Antonio Méndez Guzmán***

Universidad Politécnica del Bicentenario

*hmendezg@upbicentenario.edu.mx*

***Adrián Martínez González***

Universidad Politécnica del Bicentenario

*amartinezg@upbicentenario.edu.mx*

## Resumen

A través de este trabajo se presenta el desarrollo de herramientas de inspección para Matlab basadas en un sistema de visión industrial, así como una propuesta de interfaz de usuario para la configuración de un sistema de inspección en tiempo real, usando como herramienta de adquisición una webcam común para detección de defectos como ausencia o desplazamiento de elementos con respecto a componentes Thru-Hole, así como la detección de corto-circuito, soldadura excesiva, componente sin soldadura, detección de perforaciones, ausencia de componentes y conteo de circuitos integrados SMT. La implementación de este sistema se realiza en Matlab® presentándose una comparativa en funcionalidad con el software In-Sight Explorer® versión 4.9.3.

**Palabra(s) Clave(s):** control de calidad, inspección PCB, visión artificial.

## 1. Introducción

En los procesos industriales en la región del Bajío, existen empresas dedicadas a la manufactura electrónica para el área automotriz, implementando productos como tableros de control electrónicos, tarjetas electrónicas como sensores para velocidad, estado de puertas, vidrios, control de chispa de encendido, etc.

Para establecer un producto de calidad hacia el mercado, es indispensable la verificación de dichas tarjetas antes de proceder su montaje en los autos para asegurar un funcionamiento óptimo. Sin embargo en el proceso de fabricación y montaje de componentes en tarjetas electrónicas se generan defectos en el proceso ya sea en la colocación de los componentes en el proceso de SMT (Surface Mount Technology) o cualquiera de los procesos siguientes Radial, Pin Inserter, Cut & Clinch, etc [1].

La inspección de tarjetas PCB es de suma importancia dentro del proceso de fabricación y montaje, ya que una vez que los componentes han sido colocados y han pasado por diferentes etapas de soldadura, es momento de que las tarjetas sean programadas. Una vez que el firmware en componentes como microprocesadores, FPGA's o memorias es programado, se aplica el conformal haciendo imposible la programación de las mismas si hubo un fallo intermedio en los procesos anteriores.

Actualmente en empresas del área automotriz en la región del Bajío existen grandes pérdidas en scrap debido a que la maquinaria es prácticamente automática y no permite la completa manipulación del operador. En diversas ocasiones el operador no se percata de que a la tarjeta le falta algún componente, o que alguno de estos no está colocado de manera correcta. También se presentan pérdidas en el proceso de programación de circuitos integrados de montaje superficial, cuando no existe un sistema de inspección.

El presente trabajo está enfocado en la generación de un prototipo de inspección de defectos para tarjetas electrónicas a través de procesamiento digital de imágenes, ofreciendo una solución a dichas problemáticas por medio de un sistema de inspección de bajo costo.

## **2. Marco Teórico**

Un sistema de inspección industrial cuenta con múltiples herramientas para detección, conteo, análisis dimensional, etc. Entre las principales se encuentran las herramientas presencia-ausencia, cuya finalidad es determinar si una característica está presente o ausente en la imagen sobre una región de interés (ROI).

Para el fin específico de nuestra aplicación se utilizaron las herramientas de Luminosidad, Recuento de pixeles, Blob, Margen, Círculo y Recuento de patrones.

## Herramientas Presencia-Ausencia

### Luminosidad

Esta herramienta determina si una característica está presente o ausente basándose en la intensidad promedio de escala de grises (es decir, brillo). Comunica una aceptación si la puntuación del área de la región se encuentra dentro de los límites o un rechazo si está fuera de ellos. Matemáticamente el brillo de una imagen se define con ecuación 1.

$$\bar{g} = \sum_{g=0}^{L-1} \frac{g N(g)}{M} \quad (1)$$

Donde  $L$ ,  $N(g)$  y  $M$  son el número de niveles de gris contenido en una imagen, número de pixeles en el nivel de intensidad  $g$  y el número de pixeles en la imagen, respectivamente [2].

### Recuento de Pixeles

Determina si una característica está presente o ausente basándose en el número de pixeles claros u oscuros de una región; comunica una aceptación si el número de pixeles de la región está dentro de los límites o un rechazo si esta fuera de ellos. El recuento de pixeles claros y oscuros se pueden expresar matemáticamente con ecuaciones 2 y 3.

$$n_B = \sum_{i=0}^{M-1} P(i) \quad (2)$$

$$n_D = M - n_B \quad (3)$$

Con

$$P(i) = \begin{cases} 1 & \text{if } f(i) \geq T \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Como se puede apreciar en la ecuación 4,  $P(i)$  denota la operación de umbralización sobre cada uno de los  $M$  píxeles de la imagen.

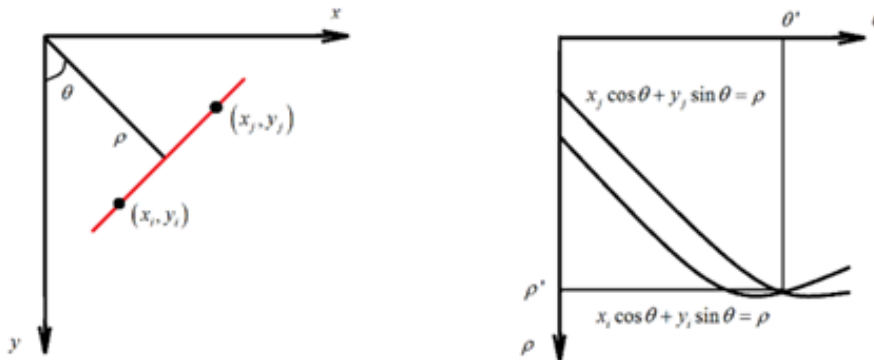
### Margen

La función de esta herramienta ayuda a determinar si hay márgenes lineales sobre nuestra pieza de inspección presentes o ausentes y comunica una aceptación si la característica de margen está dentro de los límites o mostrará un rechazo si está fuera de ellos o la característica no se encuentra sobre la pieza.

Un margen lineal se forma cuando hay un cambio brusco de brillo en una imagen, es decir un borde. Para hacer la detección de estos es necesario usar la transformada de Hough en su representación polar de una línea, ecuación 5.

$$x \cos \theta + y \sin \theta = \rho \quad (5)$$

Tal que dos puntos que pertenecen a una línea corresponden a una intersección en el plano  $\rho\theta$  [3], como se muestra en la figura 1.



a) Parametrización de líneas en el plano  $xy$ .      b) Curvas senoidales en el plano  $\rho\theta$ .

Figura 1 Correspondencia intersección punto  $(\rho', \theta')$  y línea formada por  $(x_i, y_i)$  y  $(x_j, y_j)$ .

Siendo así cada pixel de un borde tiene una representación en el plano  $\rho\theta$ , y por lo cual una línea formada por un cambio de brillo tiene múltiples intersecciones en dicho plano, y por lo tanto una línea recta en el plano  $\rho\theta$  está localizada donde hay mayor incidencia de intersecciones por cada pixel y su representación polar. Se

puede apreciar en la figura 2 como un borde está ubicado en donde los pixeles del plano  $\rho\theta$  son más brillantes.

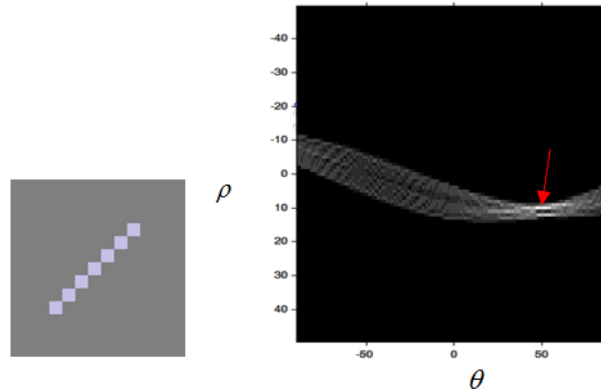


Figura 2 Transforma de Hough del borde de una imagen.

### Círculo

Determina si una característica circular está presente o ausente sobre la pieza de inspección y comunica la aceptación si la característica circular está dentro de los límites o un rechazo si está fuera de ellos o la característica no se encuentra sobre la pieza.

De forma similar que un margen, un círculo es formado por un cambio brusco de brillo pero con un patrón circular. Por lo que para su detección se puede usar una modificación al algoritmo de Hough. Partiendo de la ecuación de un círculo con centro en  $(\bar{x}, \bar{y})$ , ecuación 6.

$$(x - \bar{x})^2 + (y - \bar{y})^2 = \rho^2 \quad (6)$$

Y representando en coordenadas polares un punto en la circunferencia  $(u, v)$  con referencia al plano  $(\bar{x}, \bar{y})$  tal y como se muestra en la figura 3.

$$\begin{aligned} u &= \bar{x} + \rho \cos \theta \\ v &= \bar{y} + \rho \sin \theta \end{aligned} \quad (7)$$

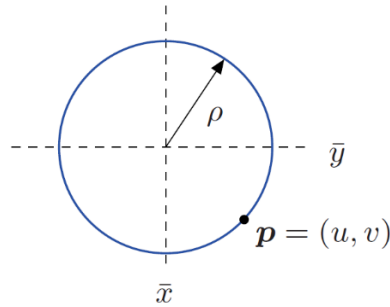


Figura 3 Representación de círculos en 2D.

De tal forma que es posible trazar un círculo de radio  $\rho$  por cada punto  $(u, v)$  en la circunferencia formada por los bordes de la imagen, y por lo tanto la localización del centro de un círculo de la imagen será en aquel lugar donde haya mayor incidencia de intersecciones por cada pixel y su representación polar [4]. Se puede apreciar en la figura 2 como un borde está ubicado en donde los pixeles del plano  $\rho\theta$  son más brillantes. En la figura 4 se detalla la localización del centro de un círculo, a través de la traza y acumulación de intersecciones en la traza de círculos por cada punto  $(u, v)$  sobre la circunferencia original.

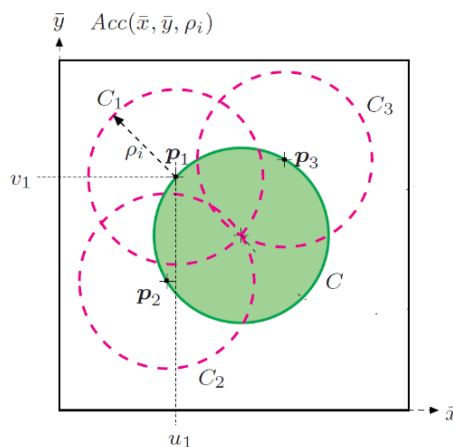


Figura 4 Transforma de Hough para círculos.

### Blob

Determina si está presente o ausente un grupo de píxeles claros u oscuros conectados, denominado Blob. Comunica una aceptación si la característica de Blob está dentro de los límites, o un rechazo si está fuera.

Esta herramienta es muy similar a la de recuento de pixeles, sin embargo, la herramienta de blob una vez que ha umbralizado la imagen elige de entre los objetos localizados (segmentos en blanco dentro de la región de interés ROI verde mostrado en la figura 5) el objeto más relevante de la región. El objeto más relevante de una imagen puede ser elegido por criterios de integridad de contorno o área mayor en pixeles, para nuestro caso por área.

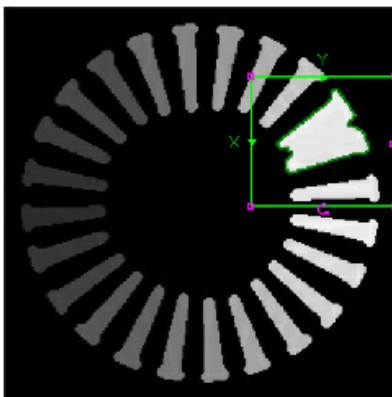


Figura 5 Segmentación por Blob.

### Recuento de patrones

Cuenta el número de características de patrón entrenadas. Comunica el número de patrones encontrados y una aceptación si el número de patrones encontrados está dentro de los límites especificados, o un rechazo si esta fuera de los límites o los patrones no se encuentran.

Para reconocer o clasificar un objeto en una imagen se deben extraer características de interés, para posteriormente usarlas en un clasificador de patrones ya sea para ser contabilizados o clasificarlos. Entre las características que podemos distinguir en una región de una imagen están forma, contorno, textura, tamaño, intensidades, propiedades estadísticas, etc [5].

En el caso específico de los circuitos impresos podemos contabilizar componentes como circuitos integrados (CI) SMT basados en su baja reflectancia, ya que como muestra la figura 6b hay un gran contenido de pixeles alojados en niveles muy bajos de la escala de grises, permitiendo una segmentación de estos aplicando umbralización en estas zonas y por ende la forma geométrica de los CI.

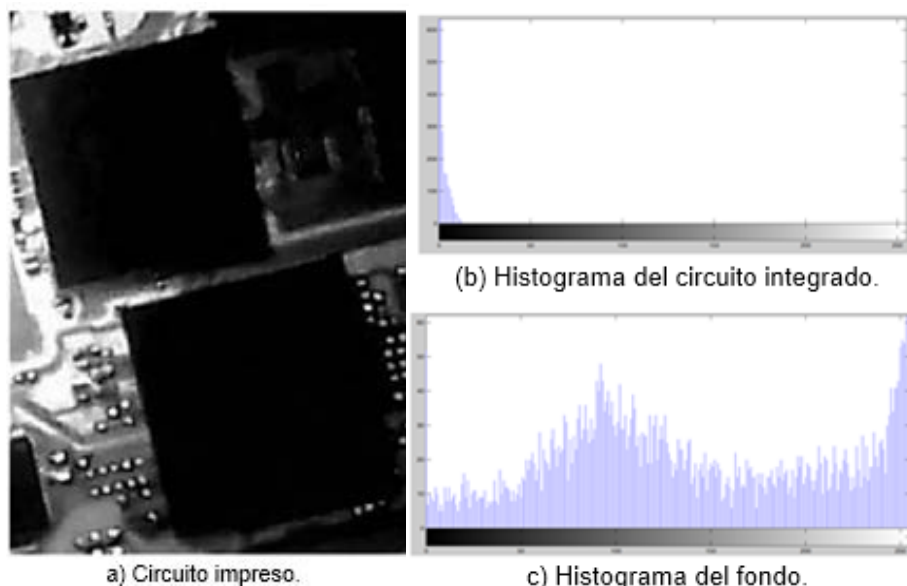


Figura 6 Reflectancia de circuito integrado SMT ante un sistema de iluminación coaxial.

Por otra parte los CI pueden estar en diferentes orientaciones y tamaños en el circuito impreso, por lo que para contabilizarlos es necesario encontrar su forma geométrica tomando en cuenta parámetros invariantes a escala y orientación.

Para encontrar su forma geométrica es necesario encontrar características que sean invariantes a la rotación y a la escala tal como los momentos de Hu.

Los momentos geométricos se definen como:

$$\mu_{pq} = \iint (x - \bar{x})^p (y - \bar{y})^q f(x, y) dydx \quad (8)$$

Donde  $\mu_{pq}$  es el momento geométrico de orden  $(p + q)$ ,  $f(x, y)$  es el valor del pixel en la posición  $(x, y)$  de la imagen y  $(x - \bar{x})$  es el centroide de la misma. Partiendo de estos momentos podemos obtener  $\mu_{pq}$ , un momento de orden  $(p+q)$  que es invariante a la escala [6].

$$\mu_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\frac{p+q}{2}}} \quad (9)$$

Sus siete momentos invariantes son:



$$\begin{aligned}\phi_1 &= \mu_{20} + \mu_{02} \\ \phi_2 &= (\mu_{20} + \mu_{02})^2 + 4\mu_{11}^2 \\ \phi_3 &= (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2 \\ \phi_4 &= (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2 \\ \phi_5 &= (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2) + (3\mu_{21} - \mu_{03})(\mu_{21} - \mu_{03})(3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2) \\ \phi_6 &= (\mu_{20} - \mu_{02})((\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2) + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}) \\ \phi_7 &= (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2) - (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})(3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2)\end{aligned}$$

Tal que un CI sin importar su tamaño y orientación tendrá momentos invariantes muy similares.

Finalmente dado que la segmentación puede producir objetos con geometrías distintas, se pueden discriminar objetos en la imagen a través de la comparación de forma geométrica por medio del cálculo de la distancia Euclidea [7], pero usando en lugar de los objetos, la comparación entre sus momentos invariantes tal que:

$$d_E^2(\phi_x, \phi_y) = \sum_{k=1}^7 (\phi_x^k - \phi_y^k)^2 \quad (11)$$

Donde  $\phi_x$ ,  $\phi_y$  son los momentos invariantes del patrón y un objeto analizado en la imagen, de esta forma para aquellos objetos que presenten la menor distancia con respecto al patrón entrenado serán reconocido y contabilizados.

### 3. Desarrollo

El funcionamiento de un sistema de verificación por visión puede representarse gráficamente como una secuencia ordenada de pasos, desde la calibración (configuración de herramientas), hasta el análisis ON-LINE y visualización de resultados (figura 7).

El proceso comienza con la captura de una imagen de prueba, asumiendo que esta es una pieza OK y con la cual se ajustan las herramientas necesarias para detectar cada uno de los fallos en tarjetas que serán verificadas posteriormente. Para cada falla se debe asignar un ROI y la herramienta adecuada para su análisis, tal que

para la verificación de múltiples errores solo será necesario agregar más herramientas como una lista ordenada de parámetros en la tarea de inspección.

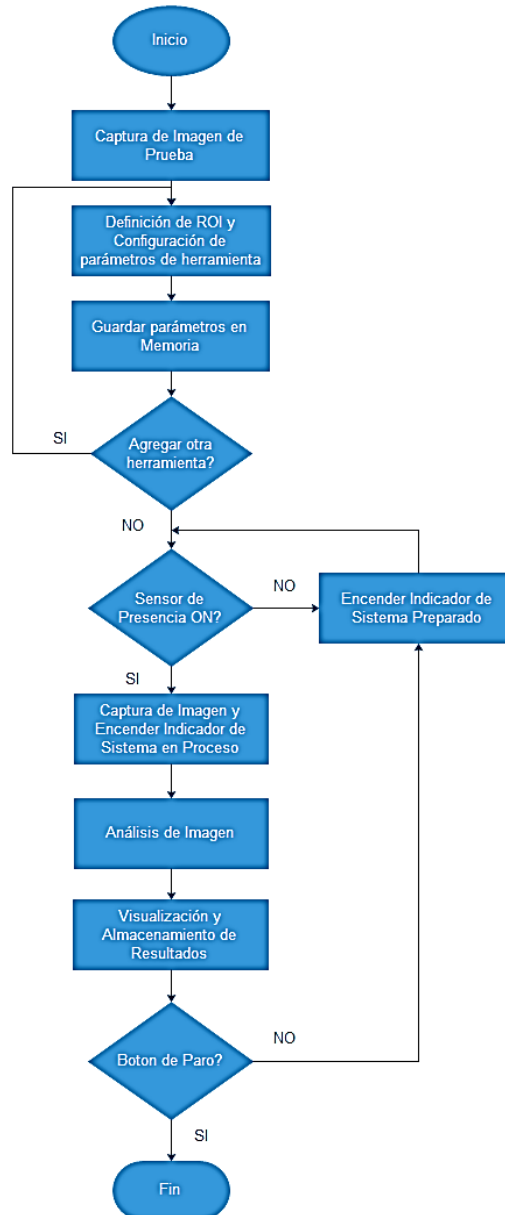


Figura 7 Proceso de verificación por inspección visual.

Una vez asignada la lista de verificaciones el sistema entra en marcha por la detección de una pieza a través de un sensor de presencia, el cual envía una señal digital al sistema de adquisición para indicar el inicio de la etapa de análisis de la

imagen. Este proceso continua de forma indefinida inspeccionando cada una de las piezas que se introduzcan en el sistema y mostrando en pantalla los resultados de cada inspección, que a través de la comparación de características físicas con la imagen patrón, el sistema puede generar una señal digital usada como elemento activador de un sistema mecánico de rechazo ante la detección de una pieza NO OK.

#### 4. Resultados

##### **Detección de defectos en tarjetas PCB para componentes Thru-hole y SMT en base a herramientas Presencia-Ausencia**

Acorde a las características antes descritas en el marco teórico podemos definir entonces los criterios para la búsqueda de defectos en tarjetas PCB.

Entre las características más distinguibles en cuanto a la presencia o ausencia de un componente se encuentran regiones con cambio de luminosidad drástico, de oscuro a claro o viceversa alrededor de una zona específica (contorno); estas regiones pueden ser detectadas por la búsqueda de círculos o márgenes, tal y como se muestra en la figura 8.

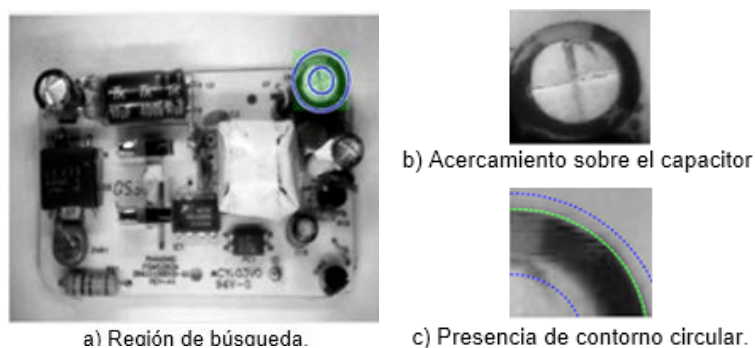
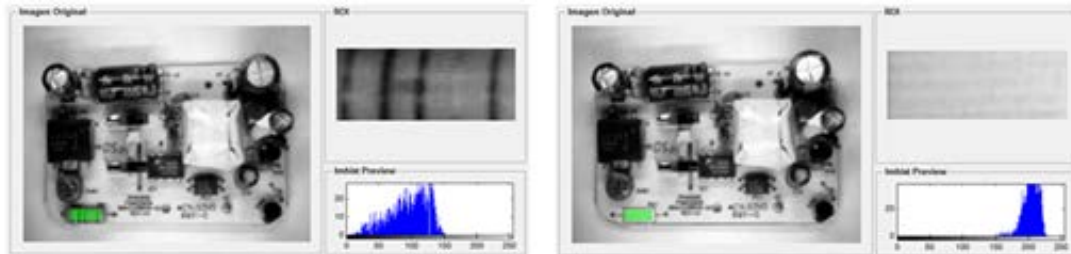


Figura 8 Detección de capacitores en base a presencia de contornos circulares.

Así mismo existen cambios en la luminosidad en una región que modifica la distribución de niveles de intensidad sobre la imagen, modificaciones presentes en defectos como desplazamiento total o parcial de componentes, ausencia de

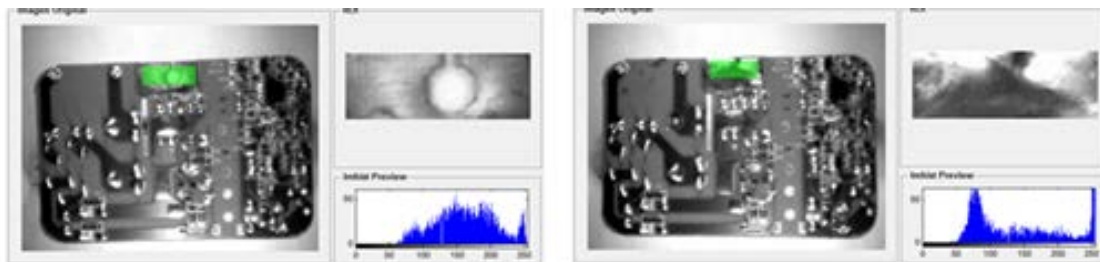
resistores (figura 9), relevadores, puentes rectificadores, transformadores, etc., todos ellos componentes Thru-Hole.



a) Intensidad sobre un resistor.      b) Cambios en distribución de histograma.

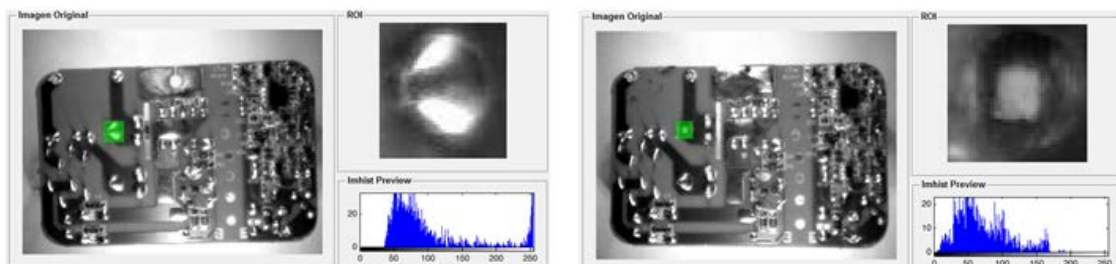
Figura 9 Ausencia de resistores.

En cuanto a la detección de defectos en soldaduras, los cambios de luminosidad pueden describir defectos como exceso de soldadura (figura 10), componente desoldado (figura 11), soldadura insuficiente y ausencia de componentes SMT entre otros.



a) Perforación necesaria para montaje en case.      b) Perforación cubierta con soldadura.

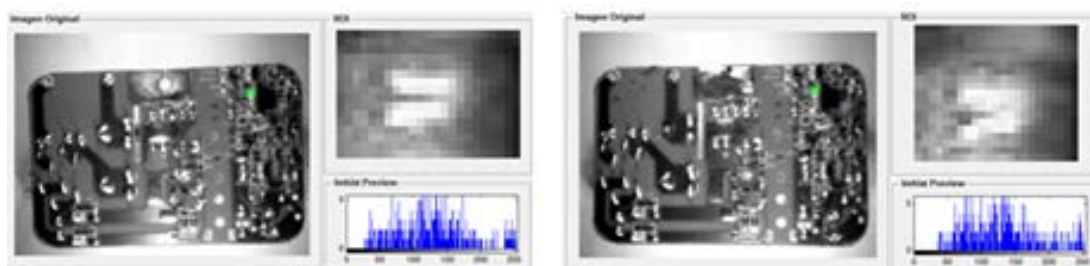
Figura 10 Detección de soldadura excesiva sobre una región.



a) Bien soldado y histograma de intensidad.      b) No soldado y variaciones en su histograma.

Figura 11 Componentes con histograma de intensidad.

En distinción a los defectos previos, la detección de corto circuito en componentes SMT no se realizaría de forma adecuada con la herramienta de luminosidad, ya que sus niveles de intensidad y distribución en la imagen son muy similares (figura 12). Sin embargo como se aprecia en la figura 12b aparece una unión entre pixeles claros cuando está presente un corto circuito en un circuito integrado SMT, por lo que la herramienta de Blob es más adecuada y nos proporciona una diferencia evidente entra la cantidad de pixeles claros cuando existe o no un corto-circuito.



a) Pines soldados adecuadamente. b) Unión pixeles claros por corto-circuito en pines.

Figura 12 Detección de corto circuito en circuito integrado SMT.

Por otra parte, para en la contabilización de CI SMT puede describirse umbralizando la imagen basados en la baja reflectancia de su cuerpo, produciendo una imagen binaria tal como se muestra en la figura 13 y tal que a través de sus momentos invariante podemos localizarlos y contabilizarlos.



a) Circuito iluminación coaxial. b) Segmentación por reflectancia. c) Ubicación y conteo de SMT.

Figura 13 Conteo de patrones para contabilización de CI.

## Implementación de prototipo e interfaz de usuario

El prototipo consiste en una cámara cerrada, donde internamente está conformada por una webcam común y un sistema de tira de leds que proporcionan una configuración de iluminación frontal y lateral (figura 14).

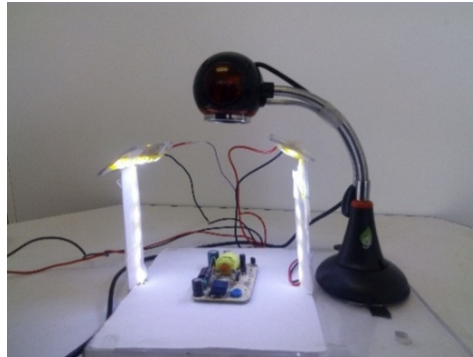


Figura 14 Sistema de adquisición de imagen y configuración de iluminación.

Para llevar el control del sistema de iluminación, así como la adquisición de imagen, configuración de herramientas de detección, análisis de imagen y visualización de resultados de inspección, se implementó un GUI en Matlab (figura 15).

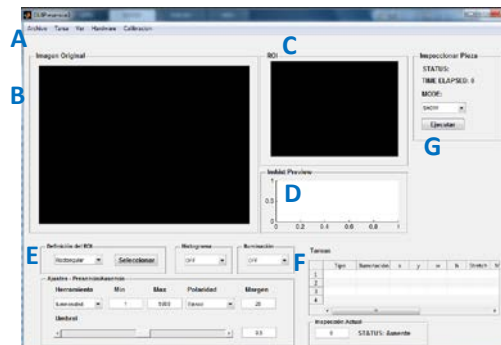


Figura 15 Interfaz de usuario para inspección de PCB.

La interfaz de usuario está conformada por las siguientes secciones: A, sección donde el usuario puede cargar una imagen, guardar y cargar tareas desde disco duro; B, sección de visualización de imágenes adquiridas; C, sección que permite ver con mayor acercamiento una región de interés establecida en A; D, visualización del histograma de la región mostrada en C; E, sección de

configuración de parámetros en las herramientas; F tabla de memorización de atributos configurados en las herramientas; G, botón de ejecución para verificación de todas las características programadas en F.

## 5. Discusión

Para la verificación de la funcionalidad del prototipo se inspeccionaron seis defectos en soldaduras, cada herramienta configurada a tolerancias mínima y máxima para su detección, tal y como se muestra en la tabla 1.

Tabla 1 Asignación de inspecciones para prueba de prototipo.

Defecto	Herramienta	Min	Max	Polaridad	Umbral
<b>Exceso de Soldadura</b>	Luminosidad	150	170		
<b>Soldadura Insuficiente</b>	Luminosidad	85	95		
<b>Componente desoldado</b>	Luminosidad	100	110		
<b>Corto circuito</b>	Blob	160	180	Blanco	
<b>Perforación desplazada</b>	Circular	8	23	Blanco	0.77
	Recuento de Pixeles	250	300	Blanco	
<b>Ausencia de componente SMT</b>	Luminosidad	110	130		

La implementación del programa en Matlab fue realizada sobre una laptop con procesador Intel Core 2 Duo a una velocidad de 2.53 GHz bajo un sistema operativo de 32 bits y memoria RAM de 4GB. El tiempo de procesamiento obtenido fue de 1.277 segundos. En comparativa con el software In Sight Explorer tiene una desventaja de un 1000 a 1 en velocidad. Sin embargo, posee la ventaja de generar visualizaciones de resultados parciales, permitiendo una conceptualización más firme del procesamiento que implica la detección de una falla y un mejor ajuste, además de que el software generado no posee un límite en número de inspecciones.

## 6. Conclusiones

El desarrollo de algoritmos para la detección de defectos en procesos industriales a través del software Matlab permite una comprensión de la importancia de variación de parámetros en procesos como umbralización, segmentación,

operadores morfológicos, detección de líneas, etc., permitiendo una conceptualización más adecuada a través de resultados parciales y finales en procesos de inspección visual.

La interfaz de usuario generada en este trabajo permitió tener una herramienta no solo para la inspección de tarjetas de circuito impreso, sino que brinda una herramienta aplicable a más procesos, ya que se generalizó el concepto de cada herramienta de un sistema de visión industrial.

## **7. Bibliografía y Referencias**

- [1] Basic PCB Concepts. <http://www.pcb.electrosoft.cl/04-articulos-circuitos-impresos-desarrollo-sistemas/01-conceptos-circuitos-impresos/conceptos-circuitos-impresos-pcb.html>. Fecha de consulta: abril 2016.
- [2] G. Pajares Martinsanz, J. M. de la Cruz García, *Visión por Computador – Imágenes digitales y aplicaciones*. Segunda Edición. 2008. Editorial Alfaomega Ra-Ma. México. pp. 88 – 89.
- [3] R. C. Gonzalez, *Digital Image Processing Using Matlab*. Segunda Edición. 2009. Editorial Woods & Eddins. USA. pp. 381-383.
- [4] W. Burger, M.J. Burge, *Principles of Digital Image Processing*. 2009. Editorial Springer. UK. pp. 63-65.
- [5] F. Shih, *Image Processing and Pattern Recognition – Fundamentals and Techniques*. 2010. Editorial Wiley.
- [6] J. Flusser, "On the Independence of Rotation Moment Invariants". *Pattern Recognition*. Vol. 33. Número 33. 19 de Mayo de 1999. pp. 1405–1410.
- [7] L. Wang, Y. Zhang, J. Feng, "On the Euclidean Distance of Images". *IEEE transactions on pattern analysis and machine intelligence*. Vol. 27. Número 8. 2005. pp. 1334-1339.