

# Algorithmic development for digital image processing and real time application in gears

Freedy Sotelo Valer, Doctor, Hugo Gamarra Chinchay, Doctor. Jorge López Córdova, Maestro  
Universidad Nacional de Ingeniería, Perú, fresov@uni.edu.pe, Universidad Ricardo Palma, Perú,  
jorgeluislopezc@hotmail.com Universidad Nacional de Ingeniería, Perú, hugogamarrach@gmail.com

*Abstract -- In this work some algorithmic techniques of frequent use were developed in different processes carried out using digital image processing techniques. Previously said algorithms were verified using the matlab software, then once the validity and operation of said algorithms was verified; We proceeded to the implementation of a system of verification of parameters of quality of paint, sizing and forms in a sample of gears in real time. The System was implemented using the Visual C++ programming platform, due to its robustness.*

*Keywords – artificial vision, digital image processing, algorithm, Matlab, VC++.*

Digital Object Identifier (DOI):  
<http://dx.doi.org/10.18687/LACCEI2019.1.1.396>  
ISBN: 978-0-9993443-6-1 ISSN: 2414-6390

# Desarrollo algorítmico para tratamiento digital de imágenes y aplicación en tiempo real en engranajes

Freedy Sotelo Valer, Doctor, Hugo Gamarra Chinchay, Doctor. Jorge López Córdova, Maestro  
 Universidad Nacional de Ingeniería, Perú, fresov@uni.edu.pe, Universidad Ricardo Palma, Perú, jorgeluislopezc@hotmail.com  
 Universidad Nacional de Ingeniería, Perú, hugogamarrach@gmail.com

**Resumen**– Este trabajo desarrolla e implementa algunas técnicas algorítmicas que son de uso frecuente y de mucha utilidad en procesos donde se utiliza el tratamiento digital de imágenes. La validez de dichos algoritmos fue verificada utilizando imágenes a color mediante el programa Matlab, posteriormente a la verificación de dichos algoritmos; se procedió a la implementación de un sistema de verificación de parámetros de calidad de pintado, dimensionamiento y formas, utilizando una muestra de engranajes de diferentes colores y con algunas fallas en su fabricación. Finalmente, el sistema se implementó para que trabaje en tiempo real, utilizando la plataforma de programación Visual C++, esto debido a la existencia de librerías y la robustez de la programación en dicha plataforma.

**Palabras Claves**– visión artificial, tratamiento digital de imágenes, algoritmo, matlab, tiempo real.

## I. INTRODUCCIÓN

En nuestros días en diversos campos de las diferentes actividades humanas y la industria, se puede observar que de la captación y evaluación de imágenes mediante la identificación sus características y parámetros particulares de las mismas, se puede obtener mucha información. Esta área denominada tratamiento digital de imágenes [1], con el apoyo de algoritmos diversos y apoyados además por técnicas inteligencia artificial [2], pueden servir para la implementación de diversos sistemas, que solucionen un gran número de problemas.

Entre las muchas aplicaciones que se pueden implementar podemos mencionar: reconocimiento de formas y patrones, control de calidad, sistemas de seguridad, entre otros.

En la actualidad, existen una gran cantidad de sistemas comerciales que aplican técnicas de procesamiento digital de imágenes en el mundo y en menor grado en nuestro medio. Pero cabe resaltar que estos, se caracterizan por ser de aplicación específica y tener costos elevados.

Frente a dicha realidad se hace necesario desarrollar e implementar sistemas que resuelvan problemas específicos de nuestro entorno y a un costo razonable.

En este trabajo se desarrolla e implementa una estación de control de calidad de engranajes mediante visión artificial, para ello se captan imágenes utilizando una cámara de video digital y luego se procesa en tiempo real [3].

## II. DESARROLLO DE ALGORITMOS

### A. Descripción de los algoritmos

Aquí se muestra los algoritmos y verificación de su funcionamiento utilizando el programa matlab, para tal fin se utiliza el dibujo del engranaje mostrado en la Fig.1.a, para

Digital Object Identifier (DOI):

<http://dx.doi.org/10.18687/LACCEI2019.1.1.396>

ISBN: 978-0-9993443-6-1 ISSN: 2414-6390

finalmente en la implementación hacerlo con imágenes reales en plataforma Visual C++.

### Formato RGB de imagen a color y en grises

La Fig.1.b, muestra la representación de una imagen en formato RGB (Red-Green-Blue) en una computadora; la cual está representada por un arreglo 3D o vector de matrices [4].

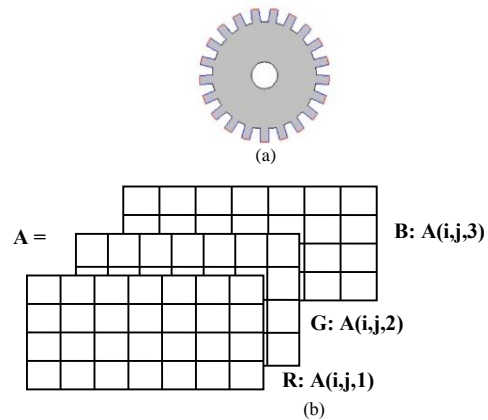


Fig. 1 Modelo utilizado para la prueba de los algoritmos, y su representación de formato RGB.

El arreglo matemático A, que representa a la imagen se muestra en (1).

$$A = \{R, G, B\} = \{A(i,j,1), A(i,j,2), A(i,j,3)\} \quad (1)$$

Una forma fácil de obtener la imagen en grises, es separar el componente Green del arreglo A. Es decir si llamamos matriz C a la imagen en grises, esta se obtiene mediante (2).

$$C(i,j) = A(i,j,2) \quad (2)$$

En base a la imagen en grises, podemos obtener el histograma de la siguiente forma [5]:

$$H(k) = 0 ; \text{ Para } k = [1, 256]$$

$$H(C(i,j)+1) = H(C(i,j)+1) + 1$$

La matriz C, es importante porque nos brinda mucha información sobre la imagen; tales como colores, formas, áreas, entre otras.

### Parámetros numéricos de la imagen

Definiendo cada valor de la matriz C, con el valor  $X_i$ , podemos obtener información útil sobre la imagen, tales como:

Media aritmética

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (3)$$

Varianza

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2 \quad (4)$$

Desviación estándar

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \quad (5)$$

Mediana

$$M_d = \begin{cases} x_{(n+1)/2} & ; \text{ si } n \text{ es impar} \\ \frac{1}{2}(x_{n/2} + x_{(n+2)/2}) & ; \text{ si } n \text{ es par} \end{cases} \quad (6)$$

*Binarización de la imagen*

Elijiendo un valor adecuado de umbralización o binarización, del histograma podemos obtener la imagen en blanco y negro con (7) [6].

$$CC(i, j) = \begin{cases} 0 & ; C(i, j) \leq 50 \\ 255 & ; C(i, j) > 50 \end{cases} \quad (7)$$

Con esto se obtiene una imagen como la mostrada en la Fig. 2, solamente en blanco y negro.



(a) Imagen en grises (b) Imagen blanco y negro  
Fig. 2 Imagen en grises y su conversión en blanco y negro.

Con la imagen en blanco y negro se puede calcular las diferentes áreas de interés, por proporcionalidad entre número de puntos de la imagen (datos de la matriz) y áreas de referencia de una imagen patrón.

*Centro de masa*

Las coordenadas  $(X_m, Y_m)$  del centro de masa de la imagen se puede obtener mediante las ecuaciones (8) y (9).

$$X_m = \frac{1}{NP} \sum_{j=1}^Q \sum_{i=1}^P i \quad ; \forall C(i, j) = 1 \quad (8)$$

$$Y_m = \frac{1}{NP} \sum_{i=1}^P \sum_{j=1}^Q j \quad ; \forall C(i, j) = 1 \quad (9)$$

donde  $NP=0$ , para luego calcular su nuevo valor con (10).

$$NP = \sum_{i=1}^P \sum_{j=1}^Q C(i, j) \quad ; \forall C(i, j) = 1 \quad (10)$$

*Contornos de la imagen*

Haciendo un barrido horizontal se captura los puntos donde  $C(i, j)=0$  y  $C(i, j+1)=1$

$$D(i, j) = \begin{cases} 1 & ; \text{ para } A(i, j) + A(i+1, j) = 1 \\ 0 & ; \text{ para } A(i, j) + A(i+1, j) \neq 1 \end{cases} \quad (11)$$

Luego con un barrido vertical se captura los puntos donde  $C(i, j)=0$  y  $C(i+1, j)=1$ .

$$D(i, j) = \begin{cases} 1 & ; \text{ para } A(i, j) + A(i, j+1) = 1 \\ 0 & ; \text{ para } A(i, j) + A(i, j+1) \neq 1 \end{cases} \quad (12)$$

En la matriz E, se almacenan los bordes mediante (13), previa inicialización de la matriz.

$$E(i, j) = 0 \quad ; \text{ Para } i = [1, P], j = [1, Q]$$

$$E(k, 1) = i \quad ; \text{ para } C(i, j) = 1 \\ E(k, 2) = j \quad ; \text{ para } C(i, j) = 1 \quad (13)$$

Donde:  $k = [1, T]$

$T = \#$  de puntos del perímetro

T almacena los puntos que están en el contorno exterior.

*Firma de la imagen*

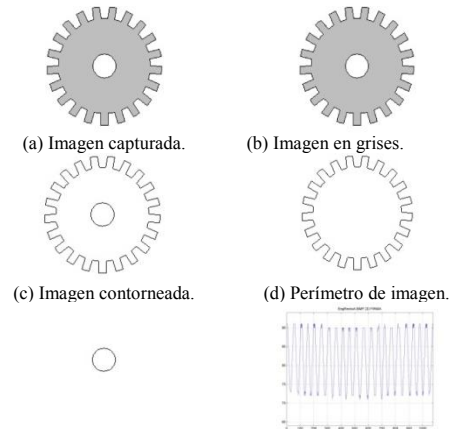
Halla las distancias euclidianas del centro de masa  $(X_m, Y_m)$  a los puntos del borde de la imagen.

$$Vr(i) = \sqrt{(VE(i, 1) - X_m)^2 + (VE(i, 2) - Y_m)^2} \quad (14)$$

Ubica como referencia la menor distancia entre el borde y el centro de masa.

*B. Ejecución de algoritmos desarrollados*

En la Fig. 3, se observan los resultados gráficos obtenidos, al implementar los algoritmos planteados en la sección anterior [7][8].



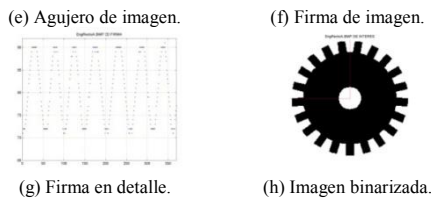


Fig. 3 Resultados gráficos al ejecutar los algoritmos.

**Número de dientes**

Se obtiene contando el número de transiciones de blanco a negro, con un barrido circular, en un radio del centro del engranaje a un valor intermedio de los dientes.

$$\begin{aligned}
 & \text{dientes} = 0 \\
 & \text{luego :} \\
 & \text{dientes} = \text{dientes} + 1 \Leftrightarrow (vcc(i) = 72) \cap (vcc(i+1) = 90)
 \end{aligned}$$

**Cálculo de áreas en la imagen**

Cambiando el interior del engranaje a un nivel de gris intermedio, se tiene la referencia de 0 para el área de la pieza y 255 para el exterior. Fig.4.

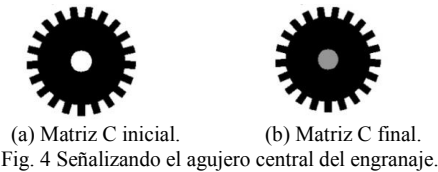


Fig. 4 Señalizando el agujero central del engranaje.

Luego, se hallan las áreas de la figura, de acuerdo a los niveles de gris, usando la proporción  $Kp = \pi \times R^2$ .

$$\text{Area} = Kp \times \frac{Na}{Np} \tag{15}$$

donde:

- Kp = Constante de proporcionalidad.
- Na = Número de puntos del área a calcular.
- Np = Número de puntos del agujero.

En este caso,  $R=0,55$ ,  $\pi=3,1416$ ; entonces  $Kp= 0,950 \text{ cm}^2$   
 Área engranaje = Ap, número de puntos.  
 Área exterior = Al, número de puntos.  
 Área interior = Aa, número de puntos.

**C. Implementación del sistema de control de calidad**

La implementación del sistema de control de calidad de engranajes en tiempo real, se realizó utilizando la plataforma de programación Visual C++ [9][10].

En la Fig. 5 se muestra el diagrama de flujo del sistema de procesamiento del engranaje, luego de haber sido previamente captada la imagen.

En la Fig. 6 se puede apreciar el diagrama de flujo que muestra el funcionamiento del sistema de control completo.



Fig. 5 Diagrama de flujo del proceso de evaluación.

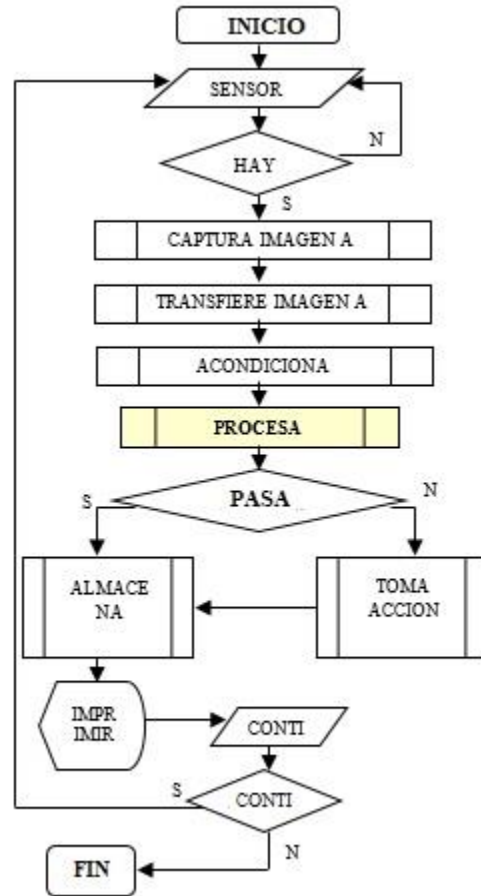


Fig. 6 Diagrama de flujo del sistema completo.

En la Fig. 7 se muestra el diagrama de bloques del sistema de control de calidad de engranajes en tiempo real. Se puede apreciar los diferentes módulos con los que cuenta el sistema.

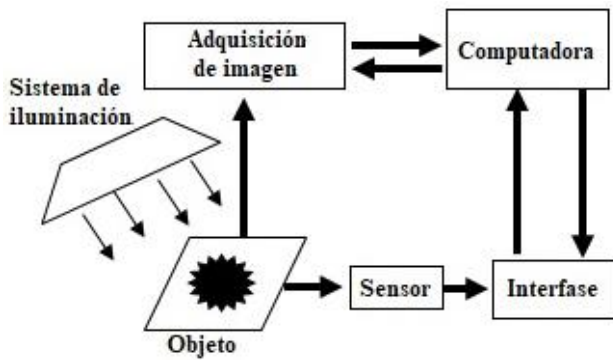


Fig. 7 Diagrama de bloques del sistema.

La Fig. 7 muestra los elementos del sistema, donde cada uno de ellos realiza funciones específicas dentro del sistema.

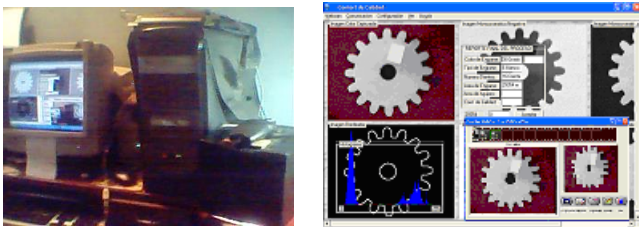
- 1) *Adquisición de imagen*, se encarga del sensado y captura de la imagen.
- 2) *Sistema de iluminación*, proporciona la cantidad y calidad de luz adecuada.
- 3) *Computadora*, ejecuta los algoritmos desarrollados e implementados en los programas.
- 4) *Programa*, codificación de los algoritmos desarrollados.
- 5) *Sensor*, detecta presencia del objeto para iniciar el proceso de control de calidad.
- 6) *Interfase*, tarjeta electrónica que recibe la señal del sensor y ordena la captura de la imagen.
- 7) *Objeto*, material de análisis del sistema.

La Fig. 8 muestra algunas imágenes obtenidas por diferentes cámaras y bajo diferentes condiciones de luz.



Fig. 8 Imágenes obtenidas bajo diferentes condiciones iluminación.

En la Fig. 9 se muestran las imágenes del sistema, la interface de software del usuario, la cámara utilizada y la interface electrónica.



(a) Sistema de control de calidad.

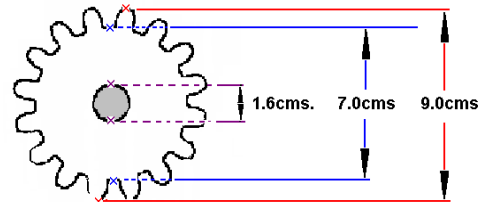
(b) Interface de usuario.



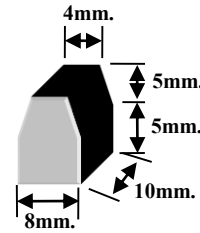
(c) Cámara de video. (d) Interface de hardware.  
Fig. 9 Imágenes del sistema y sus componentes.

#### D. Engranaje patrón

En la Fig. 10 se muestran los diámetros del engranaje patrón, que fueron utilizados para el cálculo de áreas.



(a) diámetros del engranaje patrón.



(b) Dimensiones de diente.

Fig. 10 Dimensiones del engranaje patrón.

Áreas del engranaje patrón:

$$Ad = Base \times h - 2 \times A\Delta = 0.8 \times 1 - 2 \left( \frac{0.5 \times 0.2}{2} \right) = 0.7 \text{cms.}^2$$

$$Ac = \pi(Rm - Rc)^2 = \pi(3.5 - 0.8)^2 = 22.90 \text{cms.}^2$$

$$Aa = \pi \times Rc^2 = 3.14159 \times 0.8^2 = 2.01 \text{cms.}^2$$

$$Ae = Ac + (Nd \times Ad) = 22.09 + 16 \times 0.7 = 33.29 \text{cms.}^2$$

Donde:  $Ae$  = área del engranaje patrón

$Aa$  = área del agujero

$Ad$  = área de cada diente

$Ac$  = área del cuerpo

$Nd$  = Número de dientes del engranaje

De las dimensiones de los diferentes parámetros del engranaje patrón y el cálculo de áreas, podemos resumir en la tabla I.

TABLA I  
DIMENSIONES DEL ENGRANAJE PATRÓN EN CMS. Y PÍXELES

Parámetro engranaje	Valor
Diámetro mayor	9 cms.
Diámetro menor	7 cms.
Diámetro de agujero	1.6 cms.
Área del engranaje	33.29 cms. <sup>2</sup>
Área del agujero	2.01 cms. <sup>2</sup>
Número de dientes	16

De los parámetros medidos y los obtenidos en el sistema, se determinaron los parámetros característicos del engranaje en pixeles, como se muestra en la tabla II.

TABLA II  
DIMENSIONES DEL ENGRANAJE PATRÓN EN CMS. Y PÍXELES

Parámetro	Por medición	Del sistema
Diámetro mayor	9 cms.	218 pixeles
Diámetro menor	7 cms.	180 pixeles
Diámetro de agujero	1.6 cms.	50 pixeles
Área engranaje	33.29 cms. <sup>2</sup>	30000 pixeles
Área agujero	2.01 cms. <sup>2</sup>	1380 pixeles

Para el diámetro menor en pixeles se realizó un barrido de puntos para  $\alpha = [0^\circ-20^\circ]$  usando la variable  $r$ , que detecte el cambio de gris y detectando el borde del engranaje.

En la tabla III se muestran los valores obtenidos por el programa para detectar el número de dientes, utilizando el engranaje patrón.

TABLA III  
UBICANDO DIENTES EN EL PRIMER CUADRANTE DEL ENGRANAJE

A [°]	Pe <sub>1</sub> [(xe <sub>1</sub> , ye <sub>1</sub> )]	Pe <sub>2</sub> [(xe <sub>2</sub> , ye <sub>2</sub> )]	Pe <sub>1</sub> ¿? Pe <sub>2</sub>	Dientes
0	116, 278	107, 277	diferentes	1
5	107, 277	98, 276	iguales	
15	89, 274	81, 271	diferentes	2
20	81, 271	72, 268	iguales	
25	72, 268	65, 264	diferentes	3
30	65, 264	57, 259	iguales	
35	57, 259	50, 254	diferentes	4
40	50, 254	43, 248	iguales	
45	43, 248	37, 241	diferentes	5
50	37, 241	32, 234	iguales	
60	27, 227	25, 219	diferentes	6
65	23, 219	20, 210	iguales	
70	20, 210	17, 202	diferentes	7
75	17, 202	15, 193	iguales	
80	15, 193	14, 184	diferentes	8
85	14, 184	14, 176	iguales	

Basta la verificación en el primer cuadrante, debido a la simetría de los engranajes.

### III. RESULTADOS

En la Fig. 11, se pueden observar las imágenes obtenidas por el sistema de control de calidad en tiempo real, utilizando un grupo de engranajes reales [11].

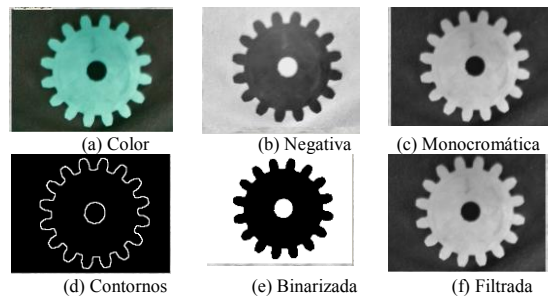


Fig. 11 Gráficas mostradas por el sistema.

También se puede observar los resultados obtenidos después del procesado de un engranaje con el sistema. Fig. 12.

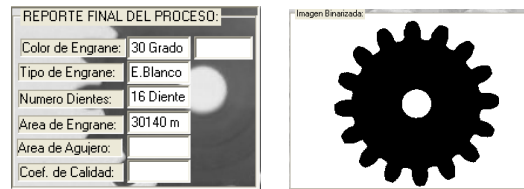


Fig. 12 Reporte de resultados.

#### Análisis de color en el sistema

En la imagen monocromática, se ve que tanto el fondo como el engranaje adquirirán niveles de gris entre  $[0-255]$ . Debido a la relación de áreas entre el fondo y el engranaje, se tendrá un histograma similar a la mostrada en la Fig. 13.

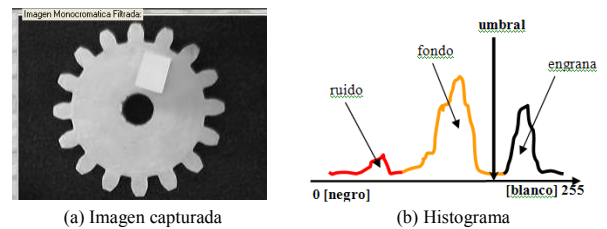


Fig. 13 Imagen capturada y su histograma.

En la Fig. 13.b, el segundo mayor área del histograma corresponde al engranaje, la mayor área corresponde al fondo y la menor área al ruido generado en la etapa de captación.

Lo descrito, se puede verificar observando el histograma obtenido por el sistema. Ver Fig. 13.

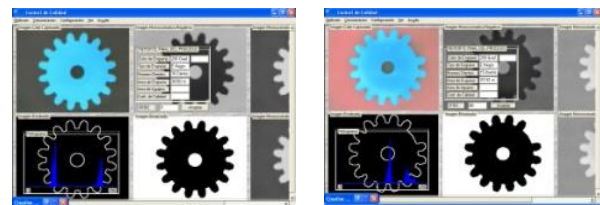


Fig. 14 Resultados gráficos de aplicar el algoritmo del promedio local.

La validez de los resultados se puede corroborar con los valores de la tabla IV. Ver Fig. 14.

TABLA IV  
RESULTADOS DE APLICAR EL ALGORITMO DEL PROMEDIO LOCAL

Engranaje	Fondo	Umbral	Color	Imagen	Fig.
Azul	negro	80	200	ok	a
Azul	rojo	100	194	ok	b

En los histogramas de la Fig. 15, se puede observar como debe ser la forma del resultado si es válido o no.

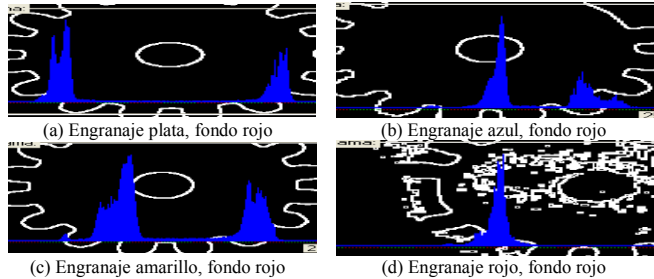


Fig. 15 Histograma de los diferentes engranajes.

### Interface de software en diferentes tipos de adquisición

La calidad de la imagen está vinculada la cámara, su interface de adquisición y su configuración. En la Fig. 16 se observan algunas interfaces utilizadas en el desarrollo del presente trabajo [12].



Fig. 16 Interface de tres tipos de cámaras distintas.

### Engranajes de diferente color en fondos distintos

El color del fondo de la imagen debe ser elegido de tal forma que puede contrastar con el engranaje, Fig. 17.

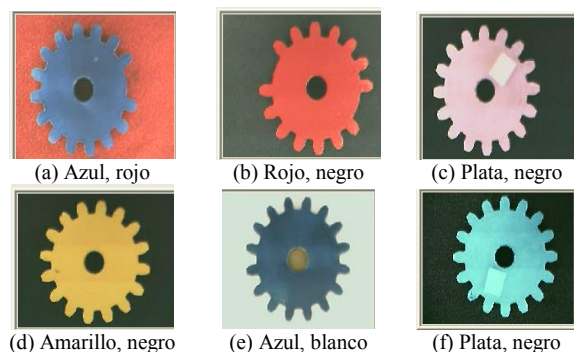


Fig. 17 Diferentes colores de engranaje y de fondo.

En estos casos se podrá evaluar en el sistema con relativa facilidad.

### Efecto de movimiento del engranaje

Otra característica importante en el procesamiento es el efecto de movimiento de la imagen que se desea capturar. Mientras más rápido se mueva la imagen, mayor será la distorsión. Dicho efecto se puede observar en la Fig. 18.



(a) Mano en reposo (b) Velocidad media (c) Velocidad mayor  
Fig. 18 Efecto del movimiento en la captura de la imagen.

En la Fig. 18.c, se puede apreciar que la imagen de la mano prácticamente no aparece en la fotografía, debido a la velocidad de su movimiento.

### Efecto de la iluminación

Otro parámetro importante es la iluminación en la etapa de adquisición de la imagen. En la Fig. 19, se puede apreciar diferentes condiciones de iluminación y el efecto en el resultado del análisis del engranaje [13].

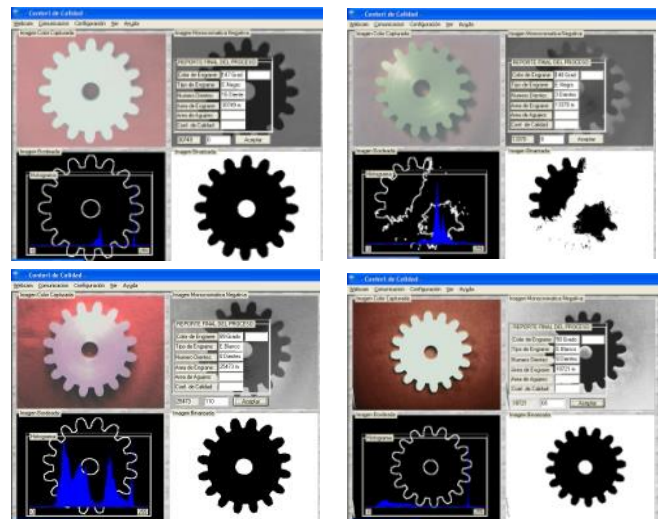


Fig. 19 Efecto del sistema de iluminación.

### Estadísticas

En la tabla V se muestra un resumen de resultados obtenidos probando el sistema.

TABLA V  
RESUMEN DE LAS DIFERENTES PRUEBAS REALIZADAS

Color	Fondo	Número Pruebas	Validos Color [%]	Validos Dientes [%]
Aluminio	Rojo	30	99	99
Aluminio	Amarillo	30	99	99
Aluminio	Negro	30	99	99
Aluminio	Azul	30	99	99

Aluminio	Verde	30	99	99
Aluminio	Blanco	-----	-----	-----
Azul	Rojo	30	98	98
Azul	Amarillo	30	90	90
Azul	Negro	-----	-----	-----
Azul	Azul	-----	-----	-----
Azul	Verde	30	90	90
Azul	Blanco	30	95	95
Amarillo	Rojo	30	98	98
Amarillo	Amarillo	-----	-----	-----
Amarillo	Negro	30	95	95
Amarillo	Azul	30	95	95
Amarillo	Verde	30	95	95
Amarillo	Blanco	-----	-----	-----
Rojo	Rojo	-----	-----	-----
Rojo	Amarillo	30	50	40
Rojo	Negro	30	50	40
Rojo	Azul	30	50	40
Rojo	Verde	30	50	40
Rojo	Blanco	30	99	99

### CONCLUSIONES

- 1) La calidad de la imagen adquirida dependerá directamente de:
  - 1.1) La iluminación del sistema.
  - 1.2) La cámara de video o fotográfica.
  - 1.3) El hardware del sistema.
  - 1.4) Las técnicas de procesamiento.
  - 1.5) Los algoritmos utilizados.
  - 1.6) Los drivers y su programa monitor.
- 2) El ingreso de parámetros de referencia se pueden detectar en forma automática enseñándole al sistema mediante técnicas de aprendizaje.
- 3) Un sistema comercial de control de botellas procesa aproximadamente 72000 botellas/hr., es decir; 20 imágenes/seg. Se podría implementar un sistema así, eligiendo adecuadamente:
  - 3.1) El dispositivo de adquisición de imágenes.
  - 3.2) Los algoritmos adecuados.
  - 3.3) El hardware de soporte.
- 4) En el caso del trabajo desarrollado, se busca que la intensidad de luz sea homogénea en la superficie y que no se produzca sombra.

### RECOMENDACIONES

- 1) Se debe trabajar en el desarrollo de algorítmicos que aumenten la eficiencia de los procesos de tratamiento digital de imágenes.
- 2) Aprovechar la aplicación la capacidad de la algoritmia en tratamiento de imágenes utilizando equipos de adquisición de imágenes que cada día son más baratos y con muchas prestaciones técnicas de calidad.
- 3) Desarrollar sistemas de iluminación adecuados para su aplicación en el tratamiento de imágenes por tratarse de una etapa crucial en el procesamiento.
- 4) Explorar nuevos campos de adquisición de imágenes en ambientes hostiles y especiales, tales como

oscuridad, presencia de radiación, ambientes opacos, entre otros.

### REFERENCIAS

- [1] Rafael C. González, Richard E. Woods, Steven L. “Digital image processing using Matlab”, Eddins, 2009.
- [2] Shi, Zhongzhi, “Advanced artificial intelligence”, Singapur, World Scientific, 2011.
- [3] Sharad R. Laxpati, Vladimir Goncharoff . “Practical Signal Processing and Its Applications: With Solved Homework Problems”, Publisher World Scientific Publishing Co., 2018.
- [4] Kueppers teoría del color.  
<https://web.archive.org/web/20080209023216/http://www.ipsi.fraunhofer.de/Kueppersfarbe/es/index.html>
- [5] Winser Alexander, Cranos Williams, “Digital Signal Processing: Principles, Algorithms and System Design”, Publisher Academic Press, 2017.
- [6] Maurice Charbit, Gérard Blanchet, “Digital Signal and Image Processing using MATLAB”, *Volume 1*, Publisher ISTE Ltd, 2015.
- [7] Irfan Turk, “MATLAB Programming for Beginners and Professionals”, Publisher Createspace, 2018.
- [8] Cuevas,Erik Diaz Cortes,Margarita A. Camarena Méndez, Pedro, “Tratamiento de imágenes con matlab”, Ed. AlfaOmega, 2017.
- [9] Microsoft, “Visual C++ Guía de aprendizaje versión 6. 0”, Ed. McGraw-Hill Interamericana. 1999.
- [10] Francisco Jav. Ceballos. “Microsoft Visual C++. Aplicaciones para Win32. 2ª Ed.”, Ra-Ma, 1999.
- [11] Francisco Charte , “Visual basic 2012”, *Manual imprescindible*, Ed. Anaya Multimedia, 2013.
- [12] Webcam logitech.  
<https://www.fnac.es/Teclados-ratones-y-webcam/Webcam-Logitech/n28930>
- [13]Gilberto Enriquez Harper. “Manual Práctico de Alumbrado”, Ed. Limusa Noriega Editores, 2003.