

UNIVERSIDAD NACIONAL DE INGENIERIA
FACULTAD DE INGENIERIA ELECTRICA Y ELECTRONICA



**RECONOCIMIENTO Y CONTROL DE CALIDAD DE
ENGRANAJES UTILIZANDO TÉCNICAS ALGORÍTMICAS DE
TRATAMIENTO DIGITAL DE IMÁGENES**

TESIS

**Para optar el Grado de Maestro en Ciencias
Mención: Automática e Instrumentación**

**PRESENTADA POR:
FREEDY SOTELO VALER**

**LIMA – PERU
2005**

UNIVERSIDAD NACIONAL DE INGENIERIA

Facultad de Ingeniería Eléctrica y Electrónica

Reconocimiento y Control de Calidad de Engranajes Utilizando Técnicas Algorítmicas de Tratamiento Digital de Imágenes

TESIS

Para optar el Grado de Maestro en Ciencias

Mención: Automática e Instrumentación

Presentada por:

Freedy Sotelo Valer

Lima – Perú

Extracto

La presente tesis esta constituida por el desarrollo, simulación e implementación de técnicas de procesamiento digital de imágenes por visión artificial, mediante computadora y su aplicación en el control de calidad de engranajes.

El trabajo, fundamentalmente se ha realizado en tres etapas:

1. Se desarrollaron los algoritmos para procesos puntuales necesarios en el sistema de control de calidad.
2. Los algoritmos desarrollados en la etapa inicial fueron verificados utilizando como soporte de programación el MatLab.
3. Se implementó el sistema completo en plataforma de programación VC++.

El sistema de control de calidad ha sido implementado en su totalidad, para lo cual se construyó una cámara de visión con iluminación propia, una WebCam para adquirir las imágenes, una computadora para realizar el proceso, un circuito electrónico de interfase entre el sistema de adquisición y la computadora, y además; el sistema de software que realiza el proceso de control de calidad.

Los resultados de la implementación del sistema demuestran que es perfectamente factible seguir desarrollando sistemas de visión artificial para la solución de problemas específicos.

INDICE GENERAL

CAPITULO 1

Introducción	1
1.1 Formulación del problema	2
1.2 Perspectiva histórica	3
1.3 Organización de la tesis	3

CAPITULO 2

Visión Artificial. Aplicaciones	5
2.1 Estación de visión artificial del CIM (Computer Integrated Manufacturing) de la URP (Universidad Ricardo Palma)	5
2.1.1 Información general	7
2.1.2 Descripción del sistema	9
2.1.3 Control del circuito impreso	10
2.2 MAPS INFORMATICA & CONTROL	11
2.2.1 Sistema de inspección industrial VISIOMAPS	11
2.2.2 Control y evaluación de parámetros de tráfico	13
2.3 HEUFT SYSTEMTECHNIK	14
2.3.1 Sistemas de control de productos llenos	14
2.3.2 Inspector de cajas con botellas vacías	16
2.3.3 Inspector de botellas vacías	16
2.3.4 Clasificación de envases	17
2.4 Otras aplicaciones	17
2.4.1 La Compañía Coca Cola – Perú	17
2.4.2 Universidad Nacional de ingeniería	17
2.4.3 Sistema de seguridad en transporte terrestre	17

CAPITULO 3

Procesamiento Digital de Imágenes	19
3.1 Introducción al procesamiento digital de imágenes	19
3.1.1 Un modelo de imagen	19
3.1.2 Muestreo y cuantificación	21
3.2 Etapas fundamentales de la visión por computador	24
3.2.1 Aplicaciones	27
3.3 Fundamentos de la visión por computador	28
3.3.1 Conceptos generales	28
3.3.2 Geometría de la formación de imágenes	32

3.3.3 El color	36
Modelo RGB	46
3.4 Adquisición y almacenamiento de imágenes	39
3.4.1 Introducción	39
3.4.2 Técnicas de iluminación	39
3.4.3 Ejemplos de Formatos de Imágenes	41
3.5 Técnicas de preprocesado	44
3.5.1 Introducción	44
3.5.2 Dominios de preprocesado	45
Dominio del espacio	45
3.5.3 Técnicas de realce	47
Negativo	47
Desplazamiento del Histograma	48
Ecuilibración del Histograma	49
Transformación Lineal de la Escala de Grises	51
Compresión de Rango Dinámico	52
Modificación del Contraste mediante funciones No Lineales	52
Fraccionamiento de los Planos de Bits	53
Realce local. Filtros realzantes	54
3.5.4 Promediado del Entorno de Vecindad o Filtro de la Media	56
3.5.5 Filtro de la Mediana	57

CAPITULO 4

Algoritmos y Simulación del Sistema	59
4.1 Introducción	59
4.2 Proceso parcial 01	60
4.2.1 Descripción del proceso	60
4.2.2 Algoritmo	61
4.2.3 Código de Programa	67
4.2.4 Ejecución del programa	67
4.2.5 Resultados gráficos	69
4.3 Proceso parcial 02	71
4.3.1 Descripción del proceso	71
4.3.2 Algoritmo	71
4.3.3 Código de Programa	73
4.3.4 Ejecución del programa	73
4.3.5 Resultados gráficos	74
4.4 Umbralización de la imagen	74
4.4.1 Proceso	74
4.4.2 Algoritmo	75
4.4.3 Código de Programa	76
4.4.4 Resultados gráficos	76
4.5 Aplicación de diferentes tipos de filtro	77
4.5.1 Proceso	77
4.5.2 Resultados gráficos	78
4.6 Manipulación de la intensidad de luz	80
4.6.1 Proceso	80
4.6.2 Resultados gráficos	80

CAPITULO 5

Descripción de Sistema Propuesto	83
5.1 Descripción del sistema	84
5.2 Parámetros del engranaje patrón	89
5.2.1 Dimensiones físicas del engranaje	89
5.2.2 Coordenadas referenciales del engranaje	90
5.3 Desarrollo del software	94
5.3.1 Estructura de la aplicación	94
5.3.2 Estructura de control de la interfase de adquisición de imágenes	95
5.3.3 Diagrama de flujo del sistema de control de calidad	96
5.3.4 Ejecución de la aplicación	97
5.3.5 Ejecución del evento WebCam - SelectSource... en aplicación	98
5.3.6 Ejecución del evento WebCam - Acquire... en aplicación	99
5.3.7 Ejecución del evento Transfer... en interfase de adquisición de imágenes	100
5.3.8 Ejecución del evento Snap Shot... en interfase de adquisición de imágenes	100
5.3.9 Ejecución del evento Exit ... en interfase de adquisición de imágenes	100
5.3.10 Modulo de procesos	100
5.4 Ejecución del sistema	103
5.4.1 Presentación de imágenes por el sistema	103
5.4.2 Determinación de áreas de referencia en el sistema	105
5.4.3 análisis del color en el sistema	106

CAPITULO 6

Resultados y Conclusiones	113
6.1 Configuración del monitor	113
6.2 Interfases de captura de imágenes	113
6.3 Engranajes de diferentes colores de sobre fondos diferentes	116
6.4 Efecto de la configuración de la interfase	117
6.5 Efecto del valor del umbral	119
6.6 Efecto de movimiento del engranaje	120
6.7 Efecto de iluminación	122
6.8 Cambio de ubicación de pieza dentro de la cámara de visión artificial (CVA)	123
6.9 Pruebas de áreas y numero de dientes	124
6.10 Tiempos de ejecución	125
6.11 Estadísticas	127
6.12 Observaciones	130
6.13 Recomendaciones	131
6.14 Conclusiones	132

APENDICE A

Código de simulación en MatLab	134
A.1 Programa: Engra01.m	134
A.2 Programa: Engra03.m	142
A.3 Programa: UAzul.m	144
A.4 Programa: Tesis01.m	146
A.5 Programa: Tesis02.m	150
A.6 Programa: Tesis03.m	154
A.7 Programa: Tesis04.m	155
A.8 Programa: Tesis05.m	158
A.9 Código para procesos diversos con imágenes	159

APENDICE B

Implementación del Sistema en VC++	163
B.1 Ejecución de la aplicación	163
B.2 Ejecución del evento: Webcam – SelectSource ... dentro de aplicación	165
B.3 Ejecución del evento: Webcam – Acquire ... dentro de aplicación	166
B.4 Ejecución del evento: Transfer ... dentro de interfase de adquisición	168
B.5 Código de procesos realizados por la aplicación	171
B.6 Funciones de la CLASE CDIB	180
B.7 Funciones de la CLASE CTwain	181

APENDICE C

Especificaciones del TWAIN	185
C.1 Introducción	185
C.2 Elementos de Twain	185
C.3 Beneficios de usar Twain	186
C.4 Características técnicas	187
C.5 Interfase del usuario de Twain	188
C.6 El protocolo basado en estados	190
C.7 Cambios necesarios para una sesión con TWAIN	192
C.8 Código de prueba de Twain	192
C.8.1 Niveles para trabajar con Ctwain	195
C.8.2 Modos de transferencia con Ctwain	196

APENDICE D

Cámaras	198
D.1 Modelos y parámetros de cámaras digitales	198

Bibliografía	210
Sitios Web	211

LISTADO DE FIGURAS

CAPITULO 2

Figura 2.1: Esquema general del CIM - URP.	5
Figura 2.2: Interconexión en red de la EVA.	6
Figura 2.3: Estación de Visión Artificial (EVA).	7
Figura 2.4: Módulos que intervienen en el proceso de control de calidad.	9
Figura 2.5: Imágenes de la evaluación de componentes en el circuito impreso.	10
Figura 2.6: Señales de alimentación para prueba del circuito.	10
Figura 2.7: Estructura mecánica de la estación.	11
Figura 2.8: Interfase de usuario y tarjeta de procesamiento.	12
Figura 2.9: Sistema de control de tráfico.	13
Figura 2.10: Diferentes equipos de control de productos llenos.	15
Figura 2.11: Capacidades adicionales.	16
Figura 2.12: Sistema inspector.	16
Figura 2.13: Sistema inspector.	16
Figura 2.14: Sistema clasificador.	17
Figura 2.15. Interfase de software.	17

CAPITULO 3

Figura 3.1: Etapas de un sistema de Visión artificial.	25
Figura 3.2: Representación visual de un objeto, como función bidimensional.	29
Figura 3.3: Imagen con su correspondiente histograma.	30
Figura 3.4: Ejemplo de histograma.	31
Figura 3.5: Histogramas correspondientes a cuatro tipos básicos de imágenes.	31
Figura 3.6: Distancia entre píxeles.	32
Figura 3.7: Transformaciones geométricas elementales en 2D.	32
Figura 3.8: Aplicando escaldado por 0.5 a una imagen en x,y.	34
Figura 3.9: Referencia de coordenadas para la aplicación de rotación.	34
Figura 3.10: Rotación de la imagen en el plano ZY.	35
Figura 3.11: Descomposición de la luz y espectro electromagnético.	36
Figura 3.12: Colores visibles por el ojo humano.	37
Figura 3.13: Combinación de colores de 3 fuentes.	38
Figura 3.14: Captura, digitalización y almacenamiento de imagen.	38
Figura 3.15: Dos imágenes de escena bajo diferentes condiciones de iluminación	39
Figura 3.16: Técnicas de iluminación. (a) Diferentes tipos de iluminación.	40
Figura 3.17: Técnicas de iluminación.	41
Figura 3.18: Ambito puntual	46
Figura 3.19: Ambito local	47
Figura 3.20: Ambito global	47
Figura 3.21: Proceso de inversión de escala de grises.	48
Figura 3.22: Resultado de la inversión de color.	48
Figura 3.23: Efecto de desplazamiento con valores -30 y 30.	49
Figura 3.24: Distribución e histograma de una imagen.	49
Figura 3.25: Ecuando el histograma.	50
Figura 3.26: Imagen de la figura 3.24 ecualizada.	50
Figura 3.27: Imagen 512x512 y 256 niveles de gris y su ecualización.	51

Figura 3.28: Transformación lineal de escala de grises.	51
Figura 3.29: Compresión de rango no lineal.	52
Figura 3.30: Imagen de 256 niveles de gris y sus correspondientes 8 planos de bits.	54
Figura 3.31: Mascara de convolución.	54
Figura 3.32: Mascara de filtro realzante.	56
Figura 3.33: Mascara de filtro High-Boost.	56
Figura 3.34: Mascaras de filtro mediana.	57
Figura 3.35: Otras mascararas de filtro mediana.	57
Figura 3.36: Aplicando operadores lógicos a una imagen.	58

CAPITULO 4

Figura 3.1: Etapas de un sistema de Visión artificial.	25
Figura 4.1: Diagrama de flujo del sistema de control de calidad en simulación.	60
Figura 4.2: Imágenes obtenidas por diferentes medios.	61
Figura 4.3: Representación matricial de imagen a color.	62
Figura 4.4: Representación matricial de imagen monocromática.	62
Figura 4.5: Redefiniendo la matriz de análisis.	63
Figura 4.6: Referencia de coordenadas (sub índices) en la matriz.	64
Figura 4.7: Obtención de la matriz de contornos.	65
Figura 4.8: Obtención de perímetro y orificio central.	66
Figura 4.9: Imágenes obtenidas en el proceso parcial 01.	71
Figura 4.10: Señalización del agujero central del engranaje.	73
Figura 4.11: Imágenes obtenidas en el proceso parcial 02.	74
Figura 4.12: Imagen utilizada para ver efecto de umbralización.	75
Figura 4.13: Imágenes obtenidas en los procesos de umbralización.	77
Figura 4.14: Imagen utilizada para proceso de filtrado.	78
Figura 4.15: Imágenes obtenidas al aplicar diferentes tipos de filtros.	80
Figura 4.16: Imágenes obtenidas al modificar la intensidad.	82

CAPITULO 5

Figura 5.1: Diagrama de flujo del sistema de control de calidad en simulación.	83
Figura 5.2: Diagrama de bloques del sistema.	84
Figura 5.3: Diagrama de bloques del sistema, mostrando la cámara de visión.	84
Figura 5.4: Circuito de interfase entre la cámara de adquisición y la computadora.	86
Figura 5.5: Imágenes obtenidas bajo diferentes condiciones.	87
Figura 5.6: fotografías de sistema de control de calidad y sus componentes.	89
Figura 5.7: Dimensiones del engranaje patrón.	90
Figura 5.8: Coordenadas referenciales relativas de puntos singulares.	91
Figura 5.9: Cambio de referencia de coordenadas para el sistema.	93
Figura 5.10: Barrido circular para determinar el número de dientes del engranaje.	93
Figura 5.11: Estructura de menú de la aplicación.	95
Figura 5.12: Estructura de control de interfase de adquisición de imagen.	96
Figura 5.13: Representación de la imagen para el programa.	96
Figura 5.14: Diagrama de flujo del sistema.	97
Figura 5.15: Diagrama de flujo al ejecutar el sistema.	98
Figura 5.16: Diagrama de flujo al ejecutar WebCam - SelectSource...	98
Figura 5.17: Diagrama de flujo al ejecutar WebCam - Acquire...	99
Figura 5.18: Ejecutando Transfer ... , en interfase de adquisición de imágenes.	100
Figura 5.19: Modulo que se ejecutan al realizar el proceso de control de calidad.	102
Figura 5.20: Distribución de imágenes en la ventana principal del sistema.	103
Figura 5.21: Referencia de coordenadas para el repintado.	104

Figura 5.22: Gráficas mostradas por el sistema.	104
Figura 5.23: Determinando parámetros de áreas.	105
Figura 5.24: Imagen capturada y su histograma respectivo.	106
Figura 5.25: Resultados gráficos de aplicar el algoritmo del color dominante.	107
Figura 5.26: Resultados gráficos de aplicar el algoritmo del segundo color.	108
Figura 5.27: Resultados gráficos de aplicar el algoritmo del promedio global.	109
Figura 5.28: Resultados gráficos de aplicar el algoritmo del promedio local.	110
Figura 5.29: Histograma de los diferentes engranajes	111
Figura 5.30: Resultados gráficos al cambiar el fondo de la imagen.	111
Figura 5.31: Interfase de adquisición de imágenes.	112
Figura 5.32: Algunos colores y sus respectivos negativos.	112

CAPITULO 5

Figura 6.1: Presentación del sistema para diferentes configuraciones.	113
Figura 6.2: Tres tipos de cámaras con sus respectivas interfaces de configuración.	115
Figura 6.3: Diferentes combinaciones de color de engranaje y color de fondo.	117
Figura 6.4: Efecto de la configuración en la imagen.	118
Figura 6.5: Configuración interfase para 2 casos particulares.	118
Figura 6.6: Efecto de valor de umbral en la binarización de la imagen.	120
Figura 6.7: Efecto del movimiento en la captura de la imagen.	121
Figura 6.8: Efecto de iluminación.	122
Figura 6.9: Efecto de la ubicación del engranaje dentro de la cámara de visión.	123
Figura 6.10: Efecto de la ubicación del engranaje dentro de la cámara de visión.	123
Figura 6.11: Parámetros de área del engranaje patrón.	125
Figura 6.12: Resultados finales del sistema.	129

CAPITULO 1

Introducción

Un problema presente en diversos campos de la actividad humana y en la industria es la **evaluación** de imágenes, mediante la identificación de ciertas características particulares, las **cuales** pueden servir para tomar ciertas decisiones.

Algunas tareas que pueden realizarse mediante procesamiento digital de imágenes son:

- Reconocimiento de formas y patrones
- Detección de fallas
- Control de calidad
- Sistemas de seguridad, etc.

Si bien es cierto, existe una diversidad de sistemas comerciales que aplican las **técnicas** del procesamiento digital de imágenes en el mundo y en menor grado en nuestro medio, estos se caracterizan por ser de aplicación específica, además de tener costos elevados.

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

Así mismo existe la necesidad de ampliar en nuestro medio las bases de la **investigación** sobre el procesamiento digital de imágenes y sus aplicaciones para que más personas puedan incursionar en esta área que cada día se hace más relevante.

Este trabajo pretende contribuir en la investigación y desarrollo del procesamiento digital de imágenes y aplicaciones; dando énfasis a la parte algorítmica para la solución de programas específicos.

En este trabajo se desarrolla e implementa una estación automática de control de **calidad** mediante visión artificial, para ello se fotografian engranajes y se clasifican de acuerdo a diferentes características, las cuales se obtienen a partir de imágenes captadas

por **una** cámara de video digital (WebCam), y luego son procesadas en tiempo real aplicando las técnicas de procesamiento digital de imágenes.

1.1 Formulación del problema

En estos tiempos de alta competencia comercial, se ha hecho imprescindible la producción de calidad a menor costo. Dada las carencias de métodos productivos y maquinaria utilizada, el concepto "cero defectos" en la industria es difícil de conseguir, para esto se incorporan sistemas que controlen la calidad del producto en diferentes fases del proceso de fabricación.

Una alternativa para solucionar dicho problema es controlar el producto al final del proceso, y con esto se evita incorporar valor añadido en un producto que finalmente pueda ser rechazado.

Es así como la visión artificial y el monitoreo remoto se han introducido con fuerza en los diferentes campos de la actividad humana y productiva, mediante sistemas de reconocimiento y control de calidad en líneas de proceso que permitan el análisis del total de la producción, mediante métodos no destructivos y sin contacto.

En esta tesis mediante las técnicas del procesamiento digital de imágenes, se desarrollan algoritmos que resuelvan el problema de reconocimiento de características en un lote de engranajes, identificando parámetros representativos, tales como: áreas, firmas, número de dientes, perímetros, etc.; respecto a un engranaje patrón.

Además, se realiza la implementación del sistema de reconocimiento y control de calidad en plataforma de programación visual, utilizando como herramienta el Visual C⁺⁺.

En el desarrollo de la tesis, primero se verificaron los algoritmos con figuras creadas por software y finalmente con fotografías de los respectivos elementos de interés, ambas usando las imágenes en forma "sistemática" en una etapa inicial y luego en forma "aleatoria".

Por lo tanto, podemos concluir que se justifica el uso de la inspección visual artificial en la industria moderna. Finalmente las empresas deberán adaptarse a esta tecnología, realizando control tanto en la salida de sus productos como en la entrada de materias primas y durante el proceso de transformación.

1.2 Perspectiva histórica

El concepto de visión artificial aparece desde el momento en que se logra capturar imágenes de la vida real mediante fotografía. Pero donde empieza a desarrollarse con mucha fuerza es en la década de los 70 a partir de la aparición de los microprocesadores, y a medida que estos se hacen más potentes y se fortalece la inteligencia artificial, el procesamiento de imágenes evoluciona en forma exponencial.

Existen alrededor del mundo, muchos sistemas que vienen utilizando técnicas de procesamiento digital de imágenes en áreas tales como: la industria, medicina, campo militar, agricultura, seguridad, etc. Dichos sistemas se caracterizan por su gran calidad pero también por sus precios elevados. Esta tecnología está entrando muy fuerte en nuestro medio, y es así como se puede ver aplicaciones como, control de tráfico (semáforos, control de velocidad, etc.), sistemas de seguridad (reconocimiento de personas, huella digital, etc.) y sistemas industriales (industria cervecera, farmacéutica, etc.). Esto ha traído también como consecuencia un creciente interés por el tema dentro de la comunidad universitaria.

La perspectiva a futuro es muy prometedora pues existen sistemas en desarrollo que inclusive posibilitan la visión de personas ciegas.

1.3 Organización de la tesis

La presente tesis está organizada del siguiente modo:

En el capítulo 2 describen específicamente algunas aplicaciones de la visión artificial dentro de la industria y actividad humana. Se hace referencia a una estación de visión artificial dentro de un CIM (Computer Integrated Manufacturing), un sistema de inspección industrial, control y evaluación de parámetros de tráfico vehicular, aplicaciones en la industria del embotellado y embalaje, y otras posibilidades de aplicación en nuestro medio.

En el capítulo 3, se desarrollan los diferentes conceptos que se utilizan en el Procesamiento Digital de Imágenes (PDI). Primero se hace una introducción a la Visión por Computadora (VC) ó Visión Artificial (VA), luego se ven los fundamentos, adquisición y almacenamiento de imágenes, además; diferentes técnicas de acondicionamiento de la imagen antes del procesado final.

En el capítulo 4 se desarrollan e implementan en MatLab los diferentes algoritmos para el proceso de control de calidad mediante el procesamiento digital de imágenes.

En el capítulo 5 se muestra la implementación de los algoritmos del sistema de control de calidad y su desarrollo en plataforma de programación Visual C⁺⁺. Se ve desde la descripción del sistema hasta el desarrollo del mismo.

En el capítulo 6 se realiza una evaluación del sistema de control de calidad. Allí se describen los diferentes factores y parámetros que intervienen en el proceso de control de calidad, y además; se determina la confiabilidad de la respuesta del sistema cuando esta se ejecuta. Además, en este capítulo se muestran las observaciones, conclusiones y recomendaciones obtenidas mediante el desarrollo del presente trabajo.

El anexo A contiene los programas en MatLab.

El anexo B contiene los programas en VC⁺.

El anexo C contiene una descripción genérica sobre la clase CTwain.

El anexo D contiene información adicional sobre cámaras fotográficas.

CAPITULO 2

Visión Artificial. Aplicaciones

En nuestros días, el campo de aplicación del tratamiento digital de imágenes ha crecido considerablemente, tanto así que cada vez más actividades se inclinan hacia esta nueva tecnología. Para ilustrar tal afirmación, a continuación hacemos referencia a casos específicos:

2.1 Estación de visión artificial del CIM (Computer Integrated Manufacturing) de la URP (Universidad Ricardo Palma) [15]

Dentro de la estructura del CIM, la cual está conformada por diferentes estaciones de trabajo, una de ellas es la estación de visión artificial (EVA), la cual es la encargada del control de calidad dentro del proceso de manufactura.

En la figura 2.1 se muestra el esquema del CIM – URP.

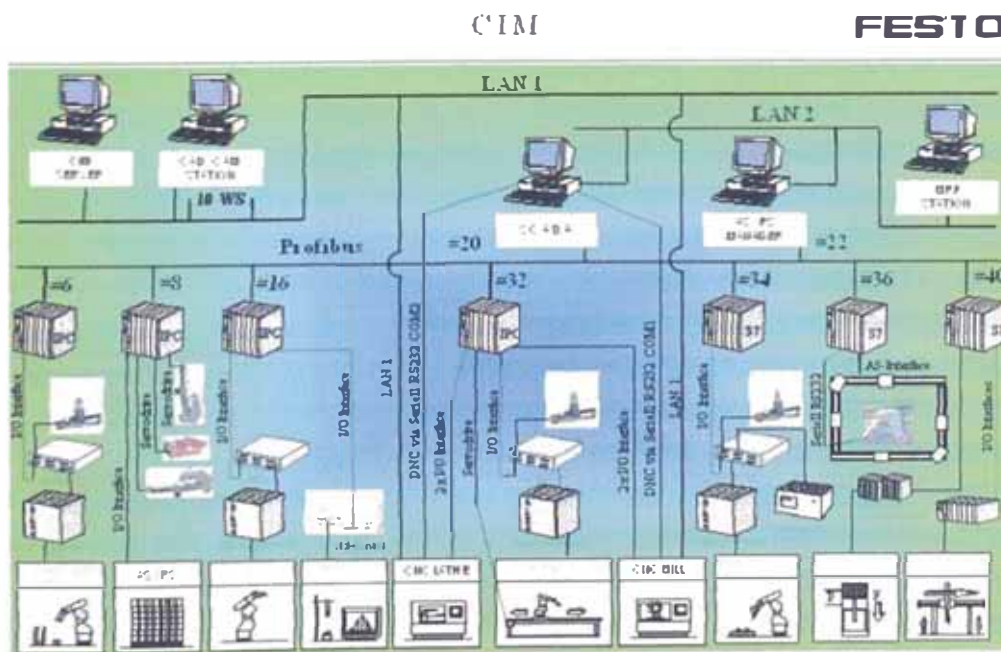


Figura 2.1: Esquema general del CIM - URP.

A su vez como parte integrante del CIM, la EVA está interconectada en red con los otros elementos del sistema, tal como se muestra en la figura 2.2.

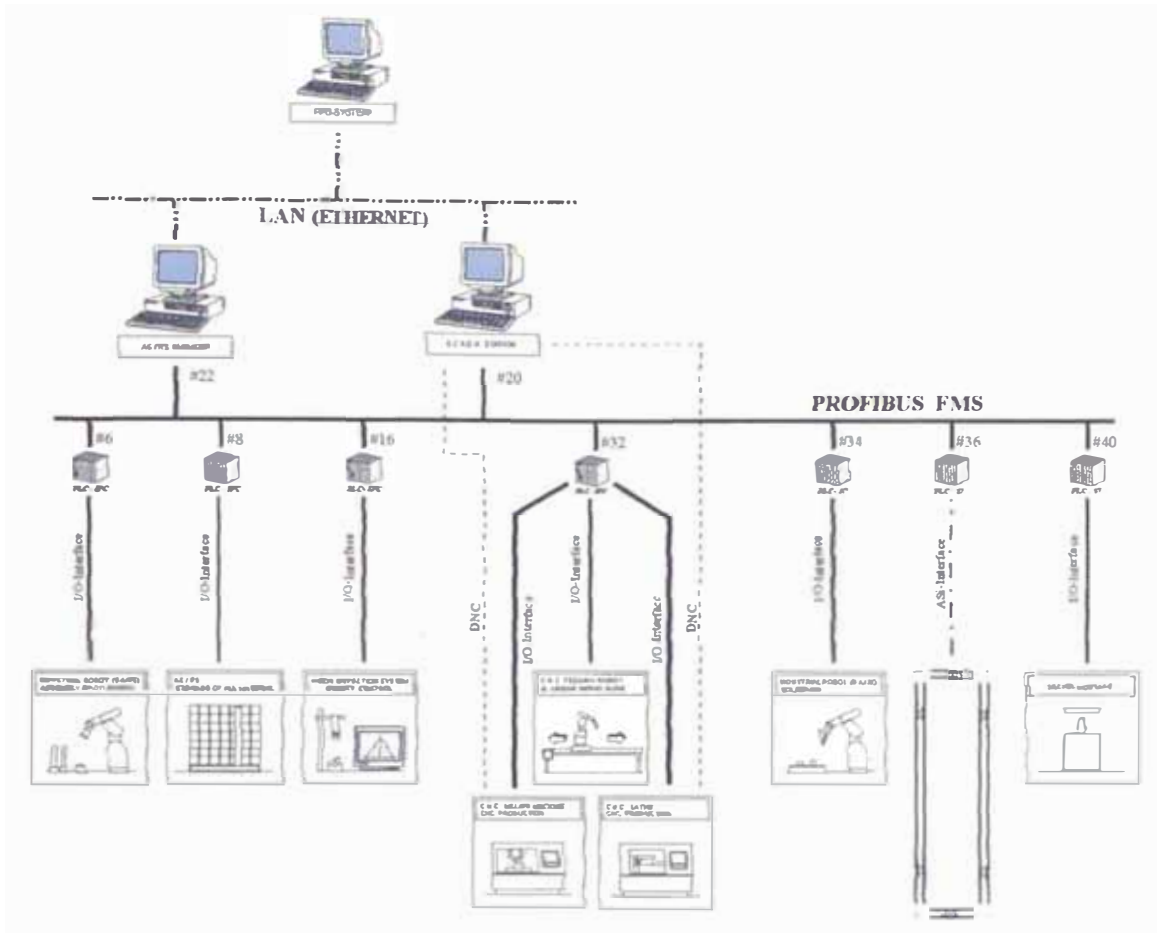


Figura 2.2: Interconexión en red de la EVA.

Entrando al nivel de estaciones de trabajo, nos encontramos con la estación de visión artificial, la cual aparece en la figura 2.3.

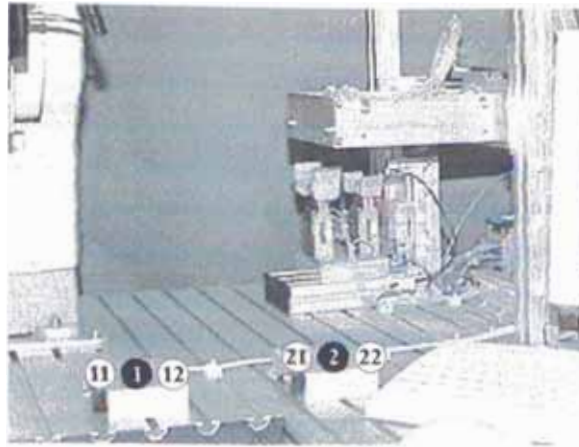
VISION INSPECTION SYSTEM



(a) Símbolo de la EVA



(b) Fotografía



(c) Detalle del soporte de la bandeja.

Figura 2.3: Estación de Visión Artificial (EVA).

2.1.1 Información general

La EVA evalúa, la presencia de componentes en un circuito electrónico impreso y su respectivo funcionamiento, el cual es producido dentro del CIM. Está conformada por 3 partes:

1. Dispositivo de adquisición de imágenes y el entorno de medición.
2. Robot, que maniobra la pieza de trabajo
3. Sistema de control de visión con la computadora respectiva. Procesa y transfiere los datos obtenidos

Las tareas específicas de la EVA son:

- **Evaluar si el circuito impreso tiene todos sus componentes instalados.**
- **Verificar el funcionamiento del circuito, supervisando la secuencia esperada.**

El sistema evalúa el nivel de gris en áreas específicas de la imagen, donde se espera encontrar un determinado elemento.

Dentro del proceso de evaluación del circuito impreso, realiza las siguientes verificaciones:

- Verifica la existencia de los C.I.
- Verifica la existencia de las resistencias
- Verifica la existencia del switch y el clip de baterías.
- Verifica el funcionamiento del circuito.

La EVA puede trabajar individualmente o en conjunto con el robot para sistemas de control de calidad optimo, además; constituye todo un sistema de ingeniería, pues involucra:

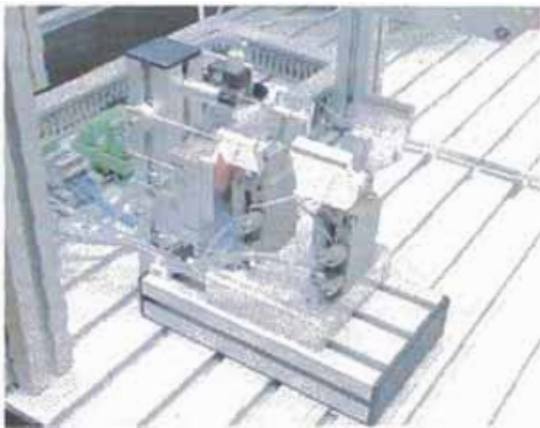
- **Parte mecánica**, es el soporte de ensamblaje de la estación dentro del CIM.

- **Parte neumática**, se utiliza para los diferentes actuadores de la estación.
- **Parte eléctrica**, para suministro de energía a la estación.
- **Parte informática**, soporte de software del sistema.
- **Parte de redes**, para la interconexión de la estación con la red.

En el proceso de control de calidad intervienen directamente:

1. **Dispositivo de medición**
2. **Fuente de luz**
3. **Bandeja de recepción**
4. **Sistema Cámara**
5. **Robot RV – E2**

Dichos módulos se muestran en la figura 2.4.



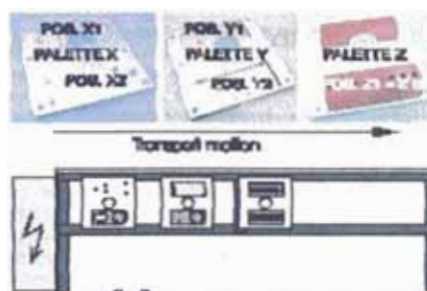
1. Dispositivo de medición Aquí la tarjeta de circuito impreso es fijada y acoplado mecánicamente para la medición.



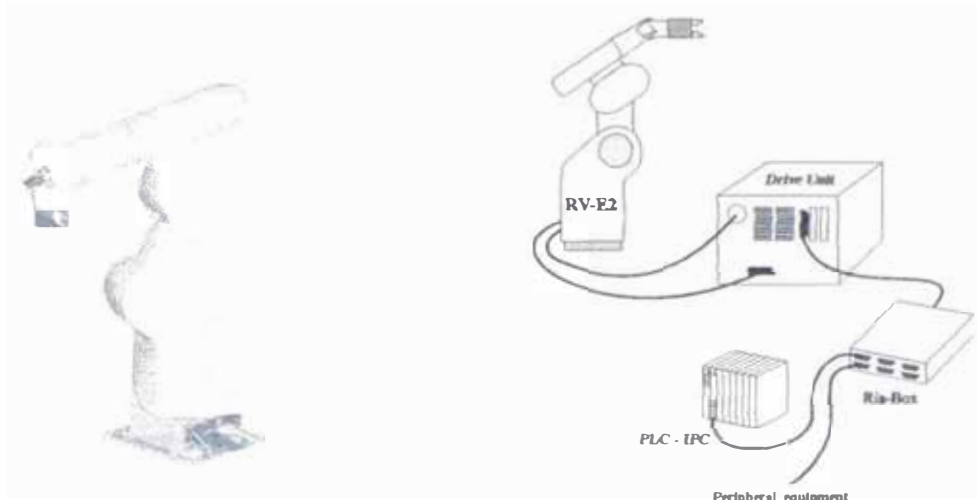
2. Fuente de Luz. La fuente de luz es responsable de suministrar la cantidad de luz requerida para el proceso. Varía según los LED's estén en ON ó OFF.



3. Bandeja de recepción Acomoda el objeto de análisis en una posición específica para ser procesada por la estación.



4. Sistema Cámara Adquiere la imagen.



5. Robot RV – E2. El robot es el responsable de la manipulación de la paleta. Elementos del sistema del Robot RV – E2.

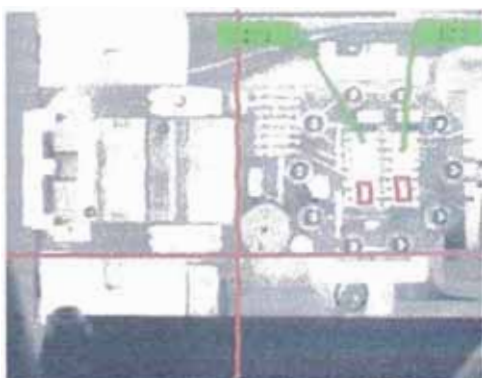
Figura 2.4: Módulos que intervienen en el proceso de control de calidad.

2.1.2 Descripción del sistema

El sistema para evaluar la presencia de los diferentes elementos del circuito impreso, establece una referencia de coordenadas absoluta y dentro de ella hace una evaluación en zonas específicas.

El sistema de coordenadas permanece fijo en todos los casos de medición. además existen niveles de iluminación definidos para cada medición, junto con sus umbrales respectivos dentro del proceso.

En la figura 2.5 se muestran los 4 tipos de evaluación que realiza el proceso. En todas las imágenes se señala la referencia de coordenadas fijas con las líneas perpendiculares rojas.



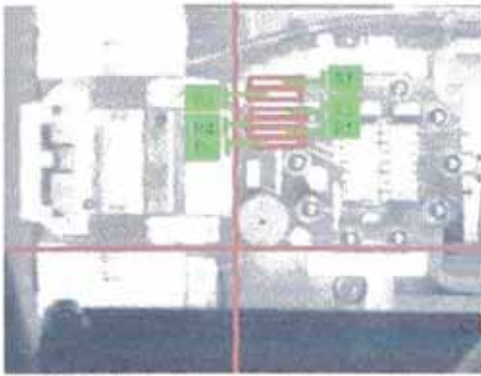
medición de orden 1:

- Dispositivos a medir : IC1 y IC2
- Fuente de imagen : canal azul (canal de video C3)
- Fuente de luz : fuente de luz 1 ON
- Tipo de medida : valor promedio de gris

El recuadro rojo () es el área de medida.

Mensajes de error:

- IC1 perdido
- IC2 perdido
- IC1 y IC2 perdidos



medición de orden 2:

- Dispositivos a medir : Resistencias de R1 a R6
- Fuente de imagen : canal azul (canal de video C3)
- Fuente de luz : fuente de luz 1 ON
- Tipo de medida : valor promedio de gris

El recuadro rojo () es el área de medida.

Mensajes de error:

- Parte 1 *perdido*, hasta
- Parte 6 *perdido*



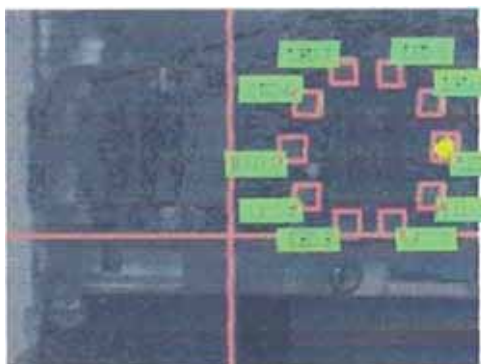
medición de orden 3:

- Dispositivos a medir : Sw1, sw2 y clip de batería
- Fuente de imagen : canal azul (canal de video C3)
- Fuente de luz : fuente de luz 1 ON
- Tipo de medida : valor promedio de gris

El recuadro rojo () es el área de medida.

Mensajes de error:

- Switch 1 *perdido*
- Switch 2 *perdido*
- Clip de batería *perdido*



medición de orden 4:

- Dispositivos a medir : LED's
- Fuente de imagen : canal rojo (canal video C1)
- Fuente de luz : fuente de luz 1 OFF
- Tipo de medida : valor promedio de gris

El recuadro rojo () es el área de medida.

Mensajes de error:

- Falla LED 1 hasta
- Falla LED 10

Figura 2.5: Imágenes de la evaluación de componentes en el circuito impreso.

2.1.3 Control del circuito impreso

La estación se encarga de alimentar la tarjeta de circuito impreso para el control de calidad del mismo. La estación genera las señales eléctricas respectivas para la tarjeta, tal como se muestra en la figura 2.6.

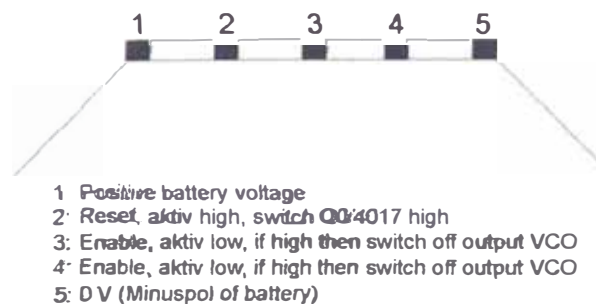


Figura 2.6: Señales de alimentación para prueba del circuito.

Se puede observar en la figura 2.5 que para la prueba de funcionamiento del circuito se apaga la fuente de luz, pues el circuito encenderá los LED's respectivos para luego evaluar dicha información.

En el diseño del sistema también es importante una estructura mecánica sólida que le permite trabajar en condiciones estables en lo que respecta a movimientos o vibraciones. En la figura 2.7 se muestra dicha estructura.



Figura 2.7: Estructura mecánica de la estación.

2.2 MAPS INFORMATICA & CONTROL [13]

Es una empresa española dedicada al desarrollo e implementación de sistemas de visión artificial. Han desarrollado muchas aplicaciones que vienen funcionando en diferentes partes del mundo, a modo de ilustración aquí mencionaremos 2 de ellas

2.2.1 Sistema de inspección industrial VISIOMAPS

Las principales características del sistema son las siguientes:

- **Facilidad** de programación y de utilización por el usuario
- **Flexibilidad** de adaptación a diferentes entornos
- **Alta** velocidad de operación gracias al preprocesador especializado en el **tratamiento** de imágenes. El sistema se basa en una computadora tipo PC.

Forma de operación

El sistema está orientado a la realización de 2 trabajos de inspección:

- **Inspección cuantitativa**, realización de medidas (áreas, longitudes, etc.), contajes y ángulos de orientación.
- **Inspección cualitativa**, verificación de la correcta realización de tareas como el ensamblado, el embotellado, el etiquetado, etc.

Características técnicas del sistema

El sistema está conformado por los siguientes elementos:

- Cámara de TV
- Placa preprocesadora de imagen
- Computadora

Gracias a la tarjeta de preprocesado de vídeo, puede operar con imágenes a una elevada velocidad, tal y como exigen la gran mayoría de aplicaciones industriales. La interconexión con el entorno se realiza a través de una interfase con la computadora.

El sistema tiene las siguientes características:

- Cuatro entradas de vídeo multiplexadas
- Detección y sincronización automática con la pieza o bien sincronización externa
- Resolución de 336 x 286 píxeles
- Tratamiento de imágenes de grises (256 niveles) binarios o contornos
- Posibilidad de discriminación de colores
- Tiempo de procesado, dependiendo del número de parámetros y ventanas a considerar (tiempos típicos entre 80ms y 200ms)
- Posibilidad de definir ventanas de coordenadas fijas sobre la imagen, o bien ventanas que se posicionan y siguen automáticamente el objeto (tracking)
- Posibilidad de simulación del test de desarrollo para validar estrategias y obtener los tiempos de procesado de cada test

En la figura 2.8 se muestra la interfase del usuario y la tarjeta de procesamiento adicional.



Figura 2.8: Interfase de usuario y tarjeta de procesamiento.

Principales Clientes

- Santiberi (Industria Alimentaria)
- Universidad Politécnica de Cataluña
- Vitri (Metalúrgica, geometría circular), etc.

2.2.2 Control y evaluación de parámetros de tráfico

Este sistema realiza las siguientes tareas:

- Medida de velocidades, media instantánea
- Obtención de parámetros para cada carril
- Cálculo de flujo de vehículos
- Detección de incidencias para señalización
 - Atascos
 - Accidentes
 - Contra dirección

La interfase del usuario con el sistema se muestra en la figura 2.9



Figura 2.9: Sistema de control de tráfico.

La empresa brinda servicios entre otras, a las siguientes empresas:

- EUROPISTAS
- AUTOPISTA VASCO ARAGONESA
- MANTENCION TECNICA PANAMERICANA (Chile), etc.

2.3 HEUFT SYSTEMTECHNIK [14]

Esta empresa alemana utiliza en sus sistemas la tecnología de la visión artificial, y su línea de desarrollo está orientado a hacia la industria del envasado y del embalaje. Entre los productos que producen y los procesos que desarrollan, se tiene:

- Inspección de envases vacíos
- Medida de nivel de llenado
- Inspección de productos acabados
- Control de cajas llenas
- Sistemas de información sobre la calidad

A continuación hacemos una referencia a algunos sistemas desarrollados por ellos y que trabajan utilizando la tecnología de la visión artificial.

2.3.1 Sistemas de control de productos llenos

En la figura 2.10 se muestra la estructura física de algunos sistemas desarrollados por HUEFT, para la industria. Dichos sistemas se utilizan en la industria para control de productos llenos. Dependiendo del tipo de producto la adquisición de la imagen en algunos casos será por medio de cámara CCD ó en otros casos por cámaras de rayos X.

A continuación se describen algunas características de estos sistemas de control, los cuales utilizan técnicas de tratamiento digital de imágenes en visión por computadora.

- Reconocimiento de envases sub-llenados y sobrellenados
- Reconocimiento de tapas
- Reconocimiento de etiquetas
- Control por microprocesadores
- **Rendimiento de hasta 72000 envases/hora**
- **Almacenamiento de parámetros para cambio fácil de producto**
- Detección de objetos extraños aun en productos opacos
- Toma de imagen por medio de cámara CCD de alta resolución, o
- Generador de rayos X compacto
- Detección de objetos extraños aun en envases de fondo convexo
- Visualización grafica de todos los valores estadísticos recogidos
- Técnica de medida con rayos X para detectar diferencias de nivel de llenado.
- Reconocimiento de tapas faltantes.

- **Velocidad de transporte de las cajas o cartones de hasta 1 m/s**
- **Verificación de la integridad de las botellas en la caja o el cartón.**

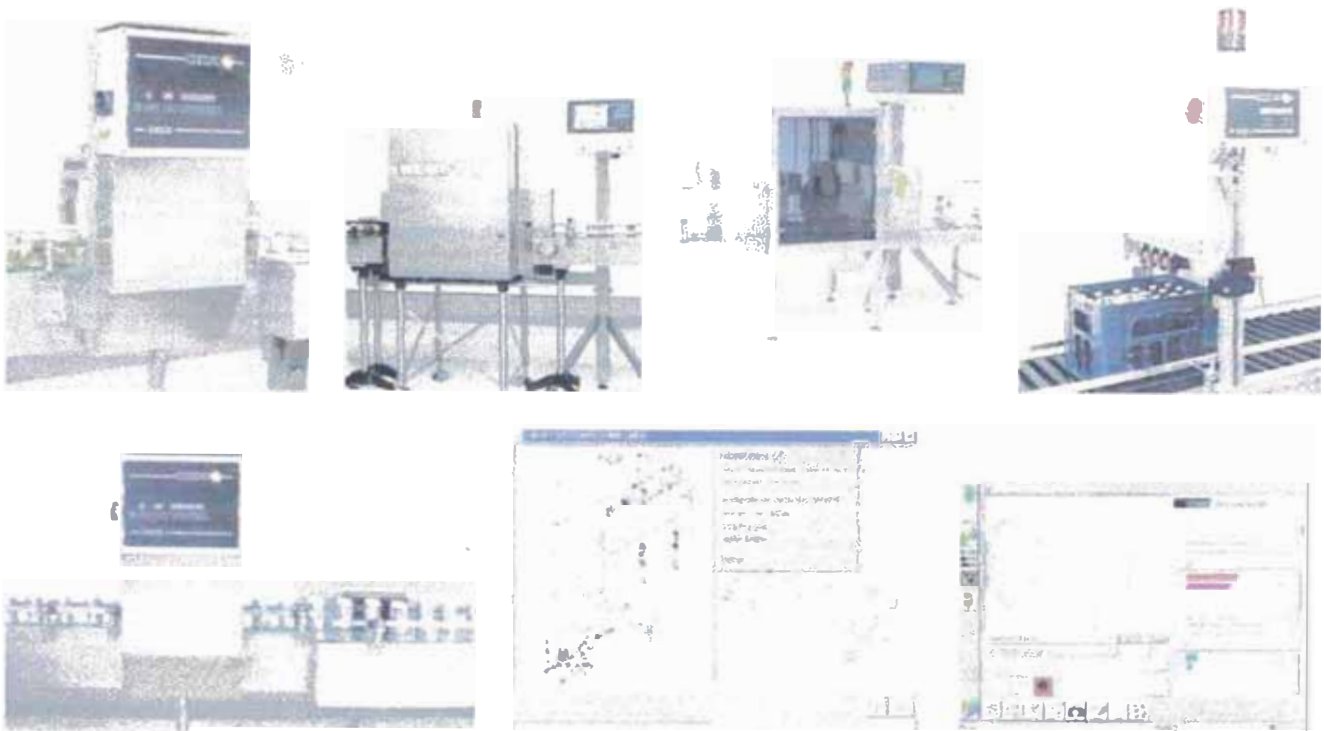


Figura 2.10: Diferentes equipos de control de productos llenos.

Entre los usuarios de esta tecnología se encuentran:

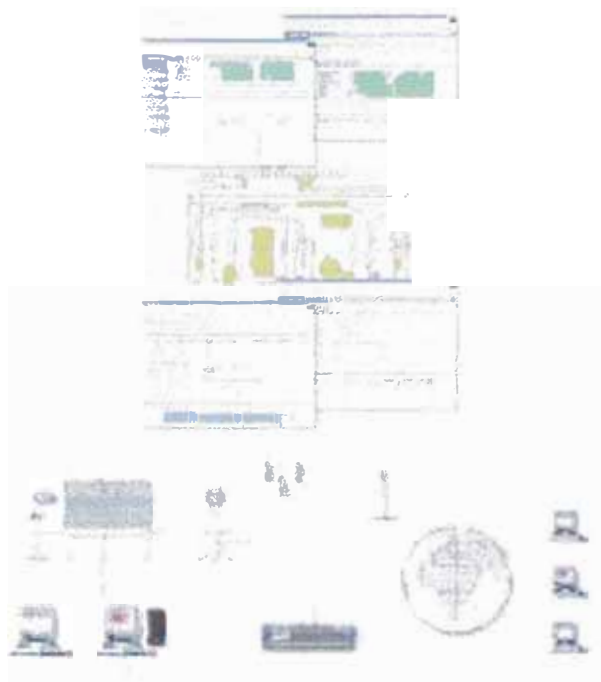
- **Industria de refrescos y cervezas**
- **Industria de víveres**
- **Industrias lácteas**
- **Industria farmacéutica, etc.**

Adicionalmente estos sistemas permiten:

- **Interfase de red integrado (Industrial Ethernet, TCP/IP)**
- **Interfase de base de datos SQL/DDE para almacenamiento y documentación**
- **una inspección final exhaustiva del producto (Final View, ver figura 2.11.a).**
- **Capacidad de analizar el proceso de la línea. Con esta herramienta se puede observar en tiempo real y economizar costos con la consecuente optimización de la línea (análisis On-Line, ver figura 2.11.b).**



(a) Final View



(b) Análisis On-Line

Figura 2.11: Capacidades adicionales.

2.3.2 Inspector de cajas con botellas vacías

Características:

- Inspección y clasificación de cajas para envases
- Reconocimiento de anillos de vidrio (solapa de transporte)
- Procesamiento de imágenes de vidrio en tiempo real con cámaras CCD
- Reconocimiento de objetos
- Cámara de color de envases
- Medida por rayos X
- Reconocimiento de logotipos



Figura 2.12: Sistema inspector.

2.3.3 Inspector de botellas vacías

Características:

- Inspección de fondo para reconocer objetos extraños y contaminaciones
- Inspección del área de boca
- Inspección para reconocer roscas dañadas
- Programa de botellas de prueba
- Rendimiento de hasta 72 000 envases/hora
- Sistemas procesadores de imágenes de alta resolución



Figura 2.13: Sistema inspector.



Figura 2.14: Sistema clasificador.

2.3.4 Clasificación de envases

- Reconocimiento de tapas
- Reconocimiento de etiquetas por iluminación estroboscópica para máxima nitidez.
- Reconocimiento en tiempo real de hasta 72 000 envases/hora (20 envase/seg)
- Reconocimiento de factores perturbadores como pajas, etiquetas despegadas, etc.
- Posibilidad de auto aprender más de 500 tipos de envase



Figura 2.15: Interfase de software.

2.4 Otras aplicaciones

Además de los sistemas mencionados, existen muchas otras posibles aplicaciones que pueden ser manejadas por visión artificial. A continuación hacemos una referencia muy breve a 3 casos puntuales, uno como presencia de control por visión artificial y otros dos como necesidad del uso de esta técnica.

2.4.1 La Compañía Coca Cola – Perú (*)

Cuenta con un sistema de visión artificial llamada Omnivisión dentro de su proceso de producción. Verifica la limpieza de las botellas. El proceso se realiza en tiempo real y además trabaja junto con un sistema automático de separación de botellas.

2.4.2 Universidad Nacional de ingeniería (*)

La facultad de ingeniería mecánica brinda el servicio de análisis de la calidad elementos mecánicos. El procedimiento es tomar fotografías del material, hacer una inspección visual (realizado por personas) y emitir el reporte respectivo.

Como es evidente la forma del proceso podría inducir a errores, y además; se podría automatizar utilizando visión artificial.

2.4.3 Sistema de seguridad en transporte terrestre (*)

En los puestos de control de la policía del Ecuador, dentro del sistema de transporte terrestre interprovincial, como medida de seguridad filman a los pasajeros y posteriormente

realizan el proceso de inspección visual (realizado por personas) para detectar a personas que tengan problemas con la justicia.

Este proceso podría ser perfectamente reemplazado por un sistema de visión artificial, con las ventajas evidentes que eso traería.

(*) Nota.- La verificación se realizó mediante visita a los lugares mencionados.

CAPITULO 3

Procesamiento Digital de Imágenes

3.1 Introducción al procesamiento digital de imágenes

La Visión por Computadora o Visión Artificial puede definirse como el conjunto de procesos de obtención, caracterización e interpretación de la información contenida en imágenes tomadas del mundo real, utilizando para el proceso una computadora.

Antes de identificar una fotografía, un Sistema de Visión por Computadora debe realizar:

- Captación o adquisición de la imagen
- Eliminación de la data que no contiene información
- Separación de objetos
- Extracción de características de cada uno de los objetos
- Reconocimiento y localización de cada uno de los objetos
- Interpretar la escena

3.1.1 Un modelo de imagen [01]

El término imagen se refiere a una función bidimensional de la luz y la intensidad, a la que indicamos por $f(x, y)$, donde el valor o amplitud de f en las coordenadas espaciales (x, y) da la intensidad (iluminación) de la imagen en este punto. Puesto que la luz es una forma de energía, $f(x, y)$ debe de ser estrictamente mayor que cero y finita, es decir,

$$0 < f(x, y) < \infty \quad (3.1)$$

Las imágenes que se perciben en las actividades visuales cotidianas provienen normalmente de la luz reflejada por los objetos. La naturaleza básica de $f(x, y)$ puede estar caracterizada por dos componentes:

- Cantidad de luz incidente procedente de la fuente sobre la escena contemplada, y
- Cantidad de luz reflejada por los objetos de escena.

De **forma** apropiada, reciben el nombre de componentes de iluminación y reflectancia y se **indican** por $i(x,y)$ y $r(x,y)$ respectivamente. Las funciones $i(x,y)$ y $r(x,y)$ se combinan como **producto**, para dar $f(x,y)$:

$$f(x,y) = i(x,y) r(x,y) \quad (3.2)$$

donde

$$0 < i(x,y) < \infty \quad (3.3)$$

y

$$0 < r(x,y) < 1 \quad (3.4)$$

La **ecuación** (3.4) indica que la reflectancia está acotada entre **0 (absorción total)** y **1 (reflexión total)**. La naturaleza de $i(x,y)$ está **determinada** por la fuente de luz y $r(x,y)$ está **determinada** por las características de los objetos de la escena.

Los **valores** dados en las ecuaciones (3.3) y (3.4) son límites **teóricos**.

Los **siguientes** datos promedio ilustran algunos rasgos típicos de $i(x,y)$.

En un día claro, el sol puede producir mas de 9000 cd (candelas)/pie², donde (1 pie ~ 28 cms.) de iluminación sobre la superficie de la tierra. Este valor decrece a menos de 1000 cd /pie² en el caso de un día nublado. En una noche clara, la luna llena produce una **iluminación** de alrededor de 0,01 cd/pie² el nivel típico de iluminación de una oficina **comercial** es de unas 100 cd/pie². De forma **similar**, valores típicos de $r(x,y)$ son: 0,01 para terciopelo negro, 0,65 para el acero inoxidable, 0,80 para un muro con pintura **completamente** blanca, 0,90 para una lamina metálica plateada y 0,93 para la nieve.

Es **una** forma común, a la intensidad de una imagen monocromática f en las **coordenadas** (x,y) denominarla nivel de gris (I) de la imagen en este punto. De las **ecuaciones** (3.2) a la (3.4) resulta evidente que I está en el rango.

$$L_{\min} \leq I \leq L_{\max} \quad (3.5)$$

En **teoría** el único requisito sobre L_{\min} es que sea positiva y sobre L_{\max} que sea finita. En la **práctica** $L_{\min} = i_{\min} r_{\min}$ y $L_{\max} = i_{\max} r_{\max}$. Empleando como guía los valores **precedentes** de iluminación y reflectancia, se pueden esperar unos valores de $L_{\min} \approx 0,005$ y $L_{\max} \approx 100$ para aplicaciones de procesamiento de imágenes de interior.

Al intervalo $[L_{\min}, L_{\max}]$ se le denomina la escala de grises. Una **práctica habitual** **consiste** desplazar numéricamente este intervalo hasta el intervalo $[0, L]$, donde en la escala, $I=0$ se considera como negro y $I=L$ se considera como blanco. Todos los valores **intermedios** son tonos de gris que varían de **forma** continua entre negro y blanco.

3.1.2 Muestreo y cuantificación [01]

Para ser utilizable para el procesamiento por medio de computadora una función de imagen $f(x,y)$ debe ser digitalizada tanto espacialmente como en su amplitud. La digitalización de las coordenadas espaciales (x,y) se denomina muestreo de imagen y la digitalización de la amplitud se conoce bajo el nombre de cuantificación de nivel de gris

Se supone una imagen continua $f(x,y)$ se describe de forma aproximada por una serie de muestras igualmente espaciadas organizadas en forma de una matriz $N \times M$ como se indica en la ecuación (3.6) donde cada elemento de la matriz es una cantidad discreta.

$$f(x,y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & \dots & f(1,M-1) \\ \vdots & \vdots & & & \\ \vdots & \vdots & & & \\ f(N-1,0) & f(N-1,1) & \dots & \dots & f(N-1,M-1) \end{bmatrix} \quad (3.6)$$

El término de la derecha de la ecuación (3.6) representada lo que comúnmente se denomina una imagen digital. A cada elemento de la matriz se le puede denominar elemento de la imagen, pixel ó pel.

En el caso de imágenes a color el formato que más se utiliza en los sistemas de procesamiento por computadora, es el RGB; donde la imagen está representada por 3 matrices bidimensionales. Cada una maneja el componente R (Red, Rojo), G (Green, Verde) y B(Blue, Azul) respectivamente, como se indica en las ecuaciones (3.7), (3.8) y (3.9).

$$f_R(x,y,1) \approx \begin{bmatrix} f(0,0,1) & f(0,1,1) & \dots & \dots & f(0,M-1,1) \\ f(1,0,1) & f(1,1,1) & \dots & \dots & f(1,M-1,1) \\ \vdots & \vdots & & & \\ \vdots & \vdots & & & \\ f(N-1,0,1) & f(N-1,1,1) & \dots & \dots & f(N-1,M-1,1) \end{bmatrix} \quad (3.7)$$

$$f_G(x,y,2) \approx \begin{bmatrix} f(0,0,2) & f(0,1,2) & \dots & \dots & f(0,M-1,2) \\ f(1,0,2) & f(1,1,2) & \dots & \dots & f(1,M-1,2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f(N-1,0,2) & f(N-1,1,2) & \dots & \dots & f(N-1,M-1,2) \end{bmatrix} \quad (3.8)$$

$$f_B(x,y,3) \approx \begin{bmatrix} f(0,0,3) & f(0,1,3) & \dots & \dots & f(0,M-1,3) \\ f(1,0,3) & f(1,1,3) & \dots & \dots & f(1,M-1,3) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f(N-1,0,3) & f(N-1,1,3) & \dots & \dots & f(N-1,M-1,3) \end{bmatrix} \quad (3.9)$$

A veces puede resultar útil expresar el muestreo y la cuantificación en términos matemáticos más formales. Si Z y R representan a los conjuntos de números enteros y reales, respectivamente, el proceso de muestreo puede entenderse como una partición en una cuadrícula de plano xy , siendo las coordenadas del centro de cada elemento de la cuadrícula un par de elementos del producto cartesiano $Z \times Z$ (también indicado por Z^2), que es el conjunto de todos los pares ordenados de el elemento (a,b) , siendo a y b números enteros pertenecientes a Z . Por tanto $f(x,y)$ representa una imagen digital si (x,y) son enteros de $Z \times Z$ y f es una función que asigna un nivel de gris (es decir un número real del conjunto de los números reales R) a cada par de coordenadas (x,y) distinto. Esta asignación funcional es el proceso de cuantificación descrito. Si los niveles de gris son números enteros (como es el caso normal) entonces Z reemplaza a R y una imagen digital se convierte en una función bidimensional (2-D) cuyas coordenadas y valores de amplitud son números enteros.

Este proceso de digitalización necesita decisiones sobre los valores para N,M y el número de niveles de gris permitido para cada píxel. En el procesamiento digital de imágenes es una práctica común que estas cantidades sean potencias enteras de 2 (ecuación (3.10)).

$$N = 2^n, \quad M = 2^k \quad (3.10)$$

y

$$G = 2^m \quad (3.11)$$

Donde G indica el número de niveles de gris. Empleando las ecuaciones (3.10) y (3.11) obtenemos el número, b, de bits necesarios para armar una imagen digitalizada

$$b = N \times M \times m \quad (3.12)$$

Si $M = N$,

$$b = N^2 m \quad (3.13)$$

Por ejemplo una imagen de 128 x 128 puntos con 64 niveles de gris necesita 98304 bits de memoria. La tabla 1 resume los valores de b, de la ecuación (3.13), para algunos valores típicos de los parámetros N y m. La tabla 2 da el número correspondiente de bytes de 8 bits, y la tabla 3 utilizando múltiplos de bytes.

Tabla 3.1: Numero de bits de almacenamiento para varios valores de N y m ($b=N^2m$).

N : m	1	2	3	4	5	6	7	8
32	1024	2048	3072	4096	5120	6144	7168	8192
64	4096	8192	12288	16384	20480	24576	28672	32768
128	16384	32768	49152	65536	81920	98304	114688	131072
256	65536	131072	196608	262144	327680	393216	458752	524288
512	262144	524288	786432	1048576	1310720	1572864	1835008	2097152
1024	1049576	2097152	3145728	4194304	5242880	6291456	7340032	8388608

Tabla 3.2: Numero de bytes de almacenamiento para varios valores de N y m ($B=N^2m / 8$)

N : m	1	2	3	4	5	6	7	8
32	128	256	512	512	1024	1024	1024	1024
64	512	1024	2048	2048	4096	4096	4096	4096
128	2048	4096	8192	8192	16384	16384	16384	16384
256	8192	16384	32768	32768	65536	65536	65536	65536
512	32768	65536	131072	131072	262144	262144	262144	262144
1024	131072	262144	393216	524288	655360	655360	917504	1048576

Tabla 3.3: Almacenamiento usando múltiplos de byte para varios valores de N y m

N : m	1	2	3	4	5	6	7	8
32	128 B	256 B	512 B	512 B	1 KB	1 KB	1 KB	1 KB
64	512 B	1 KB	2 KB	2 KB	4 KB	4 KB	4 KB	4 KB
128	2 KB	4 KB	8 KB	8 KB	16 KB	16 KB	16 KB	16 KB
256	8 KB	16 KB	32 KB	32 KB	64 KB	64 KB	64 KB	64 KB
512	32 KB	64 KB	128 KB	128 KB	256 KB	256 KB	256 KB	256 KB
1024	128 KB	256 KB	384 KB	512 KB	640 KB	640 KB	896 KB	1 MB

Puesto que la ecuación (3.6) es una aproximación a una imagen continua.

La resolución (el grado de detalle discernible) de una imagen depende estrechamente del proceso de muestreo y el número de niveles de gris usados. Cuanto más se incrementan más se aproxima la imagen digitalizada a la imagen original. Sin embargo, la ecuación (3.12) indica de que el almacenamiento y, consecuentemente, las necesidades de procesamiento crecen rápidamente en función de N, M y m

Es difícil definir que es una “buena imagen”, puesto que la calidad de la misma no es

solo ~~may~~ subjetiva si también depende de las necesidades de una aplicación determinada. La ~~imagen~~ digital se degrada conforme disminuye la resolución espacial y la cuantificación de los niveles.

3.2 Etapas fundamentales de la visión por computador [01] [02]

Un proceso completo de Visión Artificial da lugar a las siguientes seis etapas:

- a. **Captación:** A partir de una escena 3D se obtiene una representación digitalizada en 2D. Consiste en el muestreo, discretización y almacenamiento de la imagen digital.
- b. **Preprocesado:** La imagen captada en la etapa anterior se mejora resaltando aquellas características de interés y eliminando o reduciendo en lo posible las indeseadas. Se intenta preparar la imagen para las etapas posteriores.
- c. **Segmentación:** Una vez disponible una imagen mejorada en la etapa anterior, mediante la etapa de segmentación se separan los distintos objetos presentes en ella. Tiene por objeto determinar en la imagen regiones o zonas de la misma cuyos píxeles compartan algún tipo de característica. Estas regiones, previsiblemente, van a corresponder a objetos de interés en la escena.
- d. **Descripción y Representación:** Para cada uno de los objetos separados en la etapa anterior, se extraen determinadas características denominadas **descriptores** que serán representativas de ellos. Mediante alguna formulación matemática estos **descriptores** nos servirán para representar cada uno de los objetos.
- e. **Reconocimiento:** Cada uno de los objetos caracterizados en la etapa anterior es localizado e identificado, de acuerdo con una base de conocimiento previo.
- f. **Interpretación:** Una vez reconocidos todos y cada uno de los objetos, se asocia un significado a la escena. **Esta etapa está estrechamente ligada a la Inteligencia Artificial.**

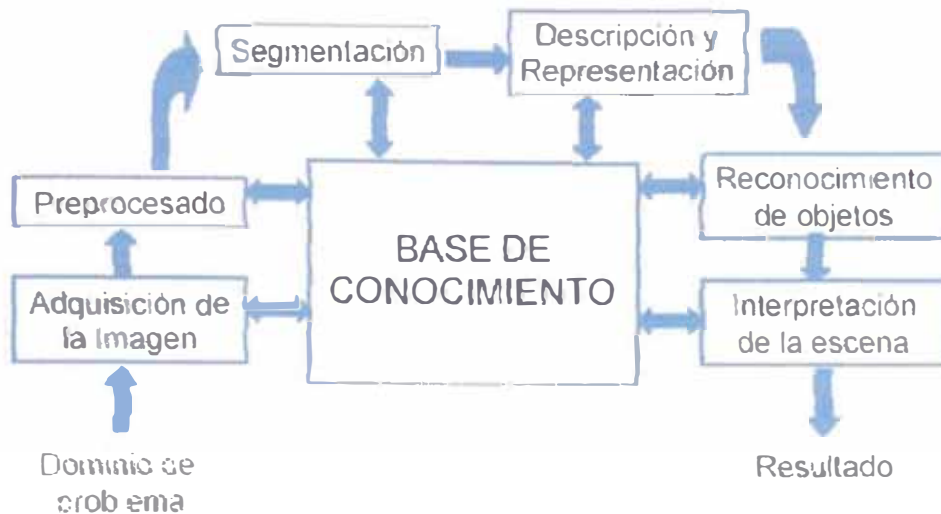


Figura 3.1: Etapas de un sistema de Visión artificial.

Las seis etapas anteriores se pueden agrupar en tres niveles de procesamiento:

- a. **nivel bajo:** Abarca desde la Captación de la imagen hasta el Preprocesado. En estas etapas se busca entregar una imagen mejorada a los niveles superiores. Por lo general, reduce la información captada, mejorando la útil y reduciendo la inútil. Por ello, este nivel debe conocer el tipo de procesamiento que harán los superiores. A este nivel, existe un **alto grado de paralelismo** y muchos de los algoritmos pueden realizarse mediante DSP especializados. Son técnicas muy sencillas, por lo que este nivel está muy desarrollado.
- b. **nivel medio:** Abarca las etapas de Segmentación y Descripción. Este nivel debe entregar al superior características concretas de todos y cada uno de los objetos existentes en la imagen, y de ser posible, su descripción de acuerdo con la base de conocimiento previa. El **grado de paralelismo no es muy alto** y son importantes las **contribuciones** de Bases de Datos e Inteligencia Artificial. La dificultad a este nivel se podría considerar como media.
- c. **Nivel alto:** Abarca las etapas de Reconocimiento e Interpretación. En este nivel **existe** una gran dependencia de técnicas modernas de programación y de **Inteligencia Artificial**. La computación es muy irregular y el **grado de paralelismo es indeterminado**. Hoy día sólo existen técnicas de este nivel en entornos reducidos y conocidos, por lo que las expectativas de desarrollo futuro son grandes.

Por otro lado, algunas operaciones fundamentales que se engloban dentro del Procesamiento Digital de Imágenes son:

- a. **Filtrado:** Mediante el filtrado de una imagen se busca una reducción o eliminación de la información indeseable o ruido que tiene asociada.
- b. **Restauración:** Consiste en perfeccionar una imagen degradada por las condiciones de obtención, ruido o errores de transmisión.
- c. **Realce:** Consiste en aumentar el contraste de los niveles de gris de una imagen a un nivel perceptible por el ojo humano; es decir, mejorar la visibilidad o resaltar características que serán útiles en etapas posteriores.
- d. **Detección y extracción de contornos:** Consiste en localizar líneas en el global de una imagen y resaltarlas. Una vez localizadas las líneas, mediante una extracción y seguimiento de contornos, se representa y extrae información de los contornos de la imagen.
- e. **Codificación:** Consiste en compactar los datos de la imagen sin reducir o reduciendo de forma controlada la información que tiene asociada. Se busca entonces, reducir su espacio de almacenamiento sin reducir su información.
- f. **Extracción de características:** Mediante la extracción de características, se genera un vector descriptor del objeto, que será utilizado por algoritmos de identificación y reconocimiento. A veces, pueden ser características geométricas tales como; áreas, perímetro, ejes, etc.. Características de contorno tales como, longitud de lados, vértices, curvaturas, etc.
- g. **Reconocimiento de objetos:** Se comparan las características anteriores con patrones almacenados en una base de datos.

En todos los casos, las técnicas de Procesamiento Digital de Imágenes se enfrentan a tres grandes problemas:

- a. **Espacio de almacenamiento:** Una imagen digital ocupa un gran espacio de almacenamiento. En la tabla 3.4 se muestran algunas cifras.

Tabla 3.4: Capacidad de almacenamiento de imágenes

Imagen	Tamaño	Capacidad	Total
Estándar gris	512x512 con 256 niveles	$512 \times 512 = 262144$ bytes	256 KB
Estándar color	512x512 con 256 niveles/plano	$512 \times 512 \times 3 = 786432$ bytes	768 KB
Satélite color	4000x4000x32 bits/pixel	$4000 \times 4000 \times 4 \approx 64000000$ bytes	62.5 MB

Lo anterior se intenta reducir de dos formas:

- Utilizando métodos compresores.
- Eliminando aquella información no útil al proceso que se va a realizar

b. Capacidad de proceso: Como consecuencia del alto número de datos a procesar, es necesaria una elevada capacidad de cálculo. Además, un proceso global lleva asociadas varias etapas. Veamos 2 ejemplos para una imagen de 512x512 con 256 niveles:

Ejemplo 1: Un algoritmo de extracción de bordes sobre una imagen monocromo estándar, necesita manipular 262.144 bytes, con unas 3000 instrucciones máquina / píxel ($\approx 5T$), lo que significa que se realizan:

Lectura de 262144 bytes

Escritura de 262144 bytes

Ejecución de $262144 \times 3000 = 786432000 \approx 768$ millones de instrucciones

Ejemplo 2: El mismo caso anterior en tiempo real (aprox. 30 imágenes/seg.):

Lectura de 262144×30 bytes/seg. = 7864320 bytes/seg.

Escritura de 262144×30 bytes/seg. = 7864320 bytes/seg

Ejecución de $7864320 \times 3000 \approx 2.3 \times 10^9$ instrucciones/seg. ≈ 23.6 Gips ≈ 23600 Mips

El problema anterior se intenta reducir:

- Eliminando aquella información no útil al proceso que se va a realizar.
- Utilizando arquitecturas paralelas
- Diseñando algoritmos eficientes

c. Diversidad de áreas implicadas en un proceso global: Un proceso global tiene relación con física (captación), lenguajes (algoritmos), hardware, Inteligencia Artificial, Bases de Datos, etc.

3.2.1 Aplicaciones [02]

Entre los campos en los que hoy día es fundamental el empleo de técnicas de procesamiento digital de imágenes se puede mencionar:

Medicina:

Pruebas de Laboratorio: recuento de glóbulos, detectar células anormales, etc.

Diagnóstico por Computado: resonancia magnética, tomografía, etc.

Industria:

Control de Calidad: inspección de láminas, nivel en llenado de botellas, etc.

Clasificación Automática de Piezas

Montaje de Circuitos Integrados en Placas de Circuito Impreso

Medición de Objetos: medición del espesor granular en lingotes de hierro, etc.

Guiado de Robots, etc.

Militar:

Vigilancia por Satélite: detección de movimiento de tropas o misiles, etc.

Visión Nocturna

Armas Inteligentes: guiado de misiles por visión, misiles aire-aire, etc.

Tele medición:

Cartografía, interpretación de fotografías aéreas, meteorología, etc.

Otros:

Microscopía electrónica: biología y medicina

Desactivación de explosivos

Sistemas de Seguridad detección e identificación de intrusos

Agricultura: control de cultivos, control de plagas, etc.

Arqueología: datación de restos, etc.

Astronomía: recuento de estrellas, galaxias, etc.

3.3 Fundamentos de la visión por computador

3.3.1 Conceptos generales [01] [02]

- a. **Imagen:** Es la representación visual de un objeto o escena. Una imagen monocroma viene representada por una función bidimensional de intensidad de luz $f(x,y)$, donde x e y representan las coordenadas espaciales, y el valor de f en un punto cualquiera (x,y) es proporcional al brillo ó nivel de gris de la imagen en ese punto. Normalmente se utiliza el convenio de asignar valores proporcionalmente más altos a las áreas más brillantes.

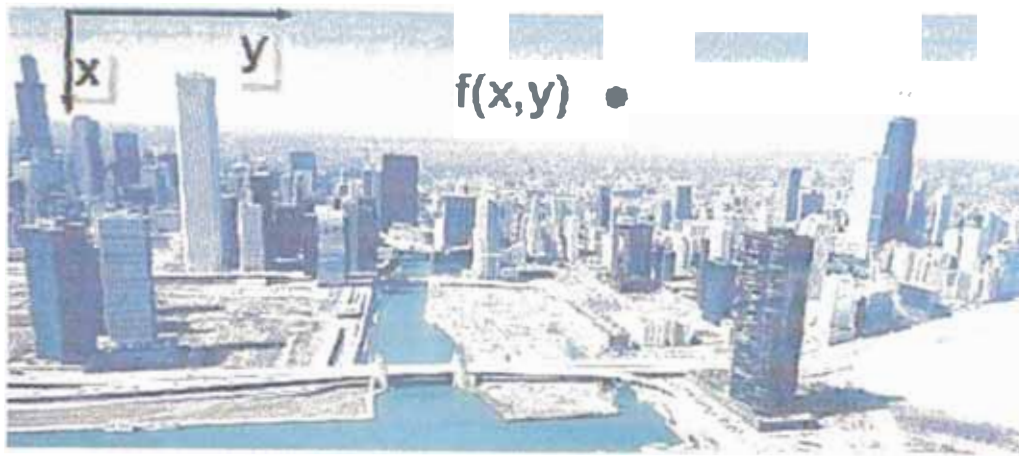


Figura 3.2: Representación visual de un objeto, como función bidimensional.

Una **imagen digital** es una imagen $f(x,y)$ que se ha discretizado tanto en las coordenadas espaciales como en el brillo. Puede considerarse como una matriz 2D de números enteros sin signo cuyos índices de fila y columna identifican un punto de la imagen y el valor del correspondiente elemento de la matriz puede indicar:

- Un nivel de luminosidad.
- Grado de absorción de un cuerpo (rayos-x)
- Sección de cruce en un destino (radar)
- Temperatura de una región, etc., etc.

Los tipos más comunes de imágenes son:

Binaria, sus valores pueden ser 0 (negro) o 1 (blanco)

Monocroma ó niveles de gris, normalmente sus valores oscilan entre 0 (negro) y 255 (blanco).

Color, normalmente utilizan 3 valores para calcular el color de un píxel, según las componentes RGB (red, green y blue).

b. Píxel: Es el elemento mínimo en una imagen. Cada píxel tiene una **resolución espacial**, que es el área que representa dicho píxel. Dicha resolución espacial es del orden de nanómetros en imágenes microscópicas y de decenas o cientos de metros en imágenes de satélite.

Longitud horizontal = nº píxeles horizontal x resolución horizontal

Longitud vertical = nº píxeles vertical x resolución vertical

Algunas resoluciones horizontales y verticales típicas son:

- 256 x 256
- 512 x 512
- 1024 x 1024

Además, cada píxel tiene una **profundidad de píxel**, que es el número de bits utilizados para representar el nivel máximo de un píxel. Las profundidades típicas son:

- **Imagen binaria.** 1 bit (2 niveles)
- **Imagen Niveles gris:** 8 bits (256 niveles de grises)
- **Imagen color baja densidad:** 8 bits x 3 planos (256 colores)

De acuerdo a lo anterior, las características de digitalización de una imagen vienen dadas por la expresión $M \times N \times L$, donde $M \times N$ indica el tamaño de la imagen, n° de filas y n° de columnas y L hace referencia al n° de bits utilizados para representar el nivel de un píxel.

c. **Histograma:** Es una medida de la distribución de los valores de los píxeles en una imagen. El histograma de una imagen es una representación gráfica de la frecuencia con que los niveles de intensidad aparecen en dicha imagen. El eje de abscisas indica los distintos niveles discretos de intensidad y el de ordenadas el número de píxeles que poseen el nivel de intensidad. El histograma se construye simplemente recorriendo toda la imagen y sumando el número de píxeles que poseen cada nivel de gris, ver figura 3.4.

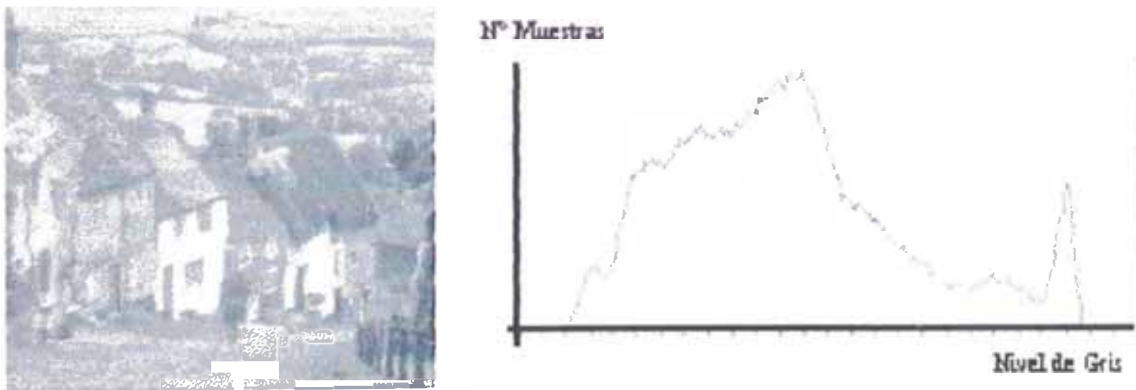


Figura 3.3: Imagen con su correspondiente histograma.

En la figura 3.4 se puede apreciar gráficamente la obtención del histograma, a partir de los niveles de gris del mismo.

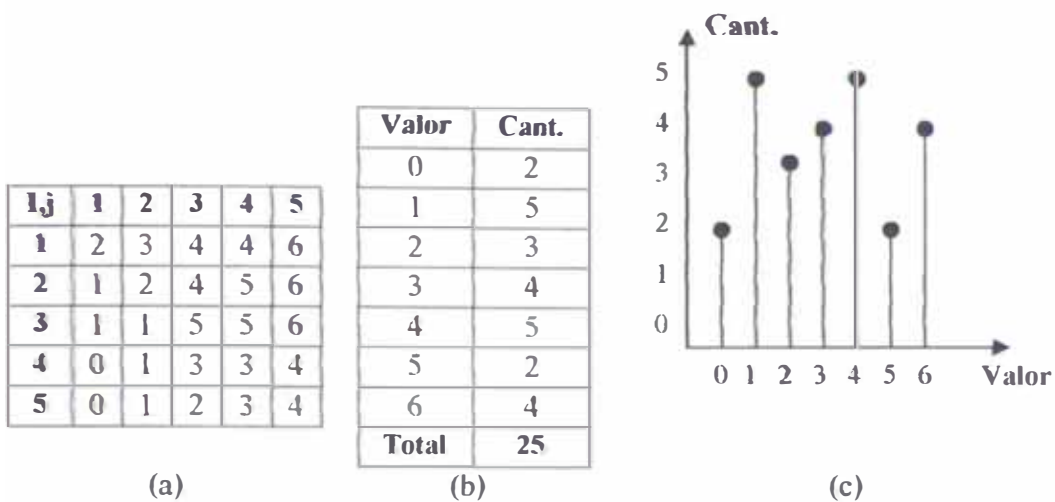


Figura 3.4: Ejemplo de histograma. (a) Imagen. (b) Distribución de niveles. (c) Histograma.

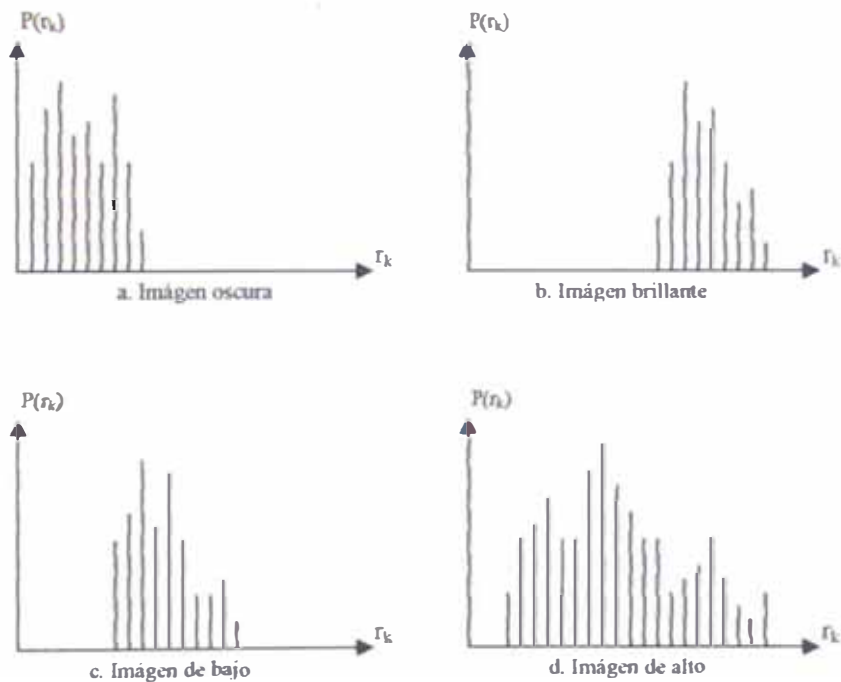


Figura 3.5: Histogramas correspondientes a cuatro tipos básicos de imágenes.

h. Medidas de distancia: En procesamiento digital de imágenes es usual utilizar distintos tipos de distancias. Dados tres píxeles $p(x,y)$, $q(s,t)$, $z(u,v)$, D es una función distancia si cumple que:

- $D(p,q) > 0$; $D(p,q) = 0$ si y sólo si $p = q$
- $D(p,q) = D(q,p)$
- $D(p,z) < D(p,q) + D(q,z)$

Las principales funciones de distancia son:

- **Distancia euclídea:**

$$D_e(p,q)=[(x-s)^2+(y-t)^2]^{1/2} \quad (3.14)$$

- **Distancia de bloques:** La distancia de bloques entre dos píxeles p y q es igual a la longitud del 4-camino más corto entre esos dos puntos. Es 1 para los 4-vecinos.

$$D_B(p,q)=|s-x|+|y-t| \quad (3.15)$$

- **Distancia de ajedrez:** La distancia de ajedrez entre dos píxeles p y q es igual a la longitud de 8-camino más corto entre esos dos puntos.

$$D_A(p,q)=\max(|x-s|, |y-t|) \quad (3.16)$$

		2		
	2	1	2	
2	1	0	1	2
	2	1	2	
		2		

(a) Distancia bloques

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

(b) Distancia ajedrez

Figura 3.6: Distancia entre píxeles.

3.3.2 Geometría de la formación de imágenes [01] [02]

En este apartado se muestran algunas transformaciones importantes usadas en el procesamiento digital de imágenes TDI. Para ello, consideramos un sistema de coordenadas cartesiano en 3D en el que cada punto se referencia por sus coordenadas (x,y,z). Cuando se trate de imágenes bidimensionales (2D) se mantendrá la norma de indicar como (x,y) las coordenadas del punto.

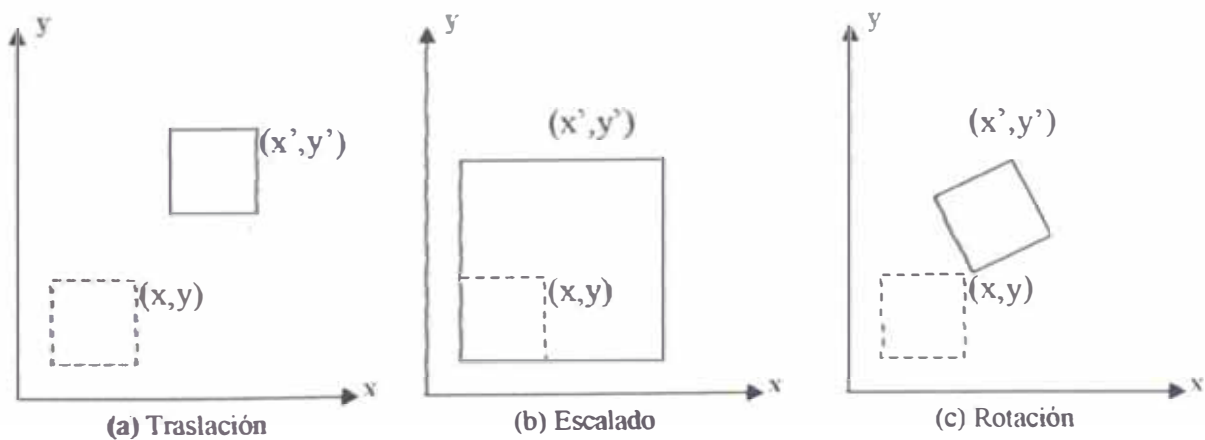


Figura 3.7: Transformaciones geométricas elementales en 2D.

Las principales transformaciones son:

a. **Traslación** (Figura 3.7.a): Implica el desplazamiento del objeto en el espacio tridimensional. Esta transformación traslada un punto (x,y,z) a una nueva posición desplazada de la original (x_0,y_0,z_0) . La traslación puede realizarse de la forma:

$$x^* = x + x_0 \quad (3.17)$$

$$y^* = y + y_0 \quad (3.18)$$

$$z^* = z + z_0 \quad (3.19)$$

Expresado en forma matricial queda como:

$$\begin{bmatrix} x^* \\ y^* \\ z^* \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & y_0 \\ 0 & 0 & 1 & z_0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3.20)$$

Para uniformizar, se suelen usar matrices cuadradas, quedando la traslación como:

$$\begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & y_0 \\ 0 & 0 & 1 & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3.21)$$

b. **Escalado** (Figura 3.7.b): Produce un cambio de escala, haciendo que éste resulte más grande o más pequeño; pero también puede usarse para alterar las proporciones del mismo. Para el cambio de escala se emplean los factores S_x , S_y y S_z :

$$x^* = x * S_x \quad (3.22)$$

$$y^* = y * S_y \quad (3.23)$$

$$z^* = z * S_z \quad (3.24)$$

Expresado en forma matricial queda como:

$$\begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3.25)$$

En la figura 3.8 se observa el efecto del escalado, aplicado a una imagen.

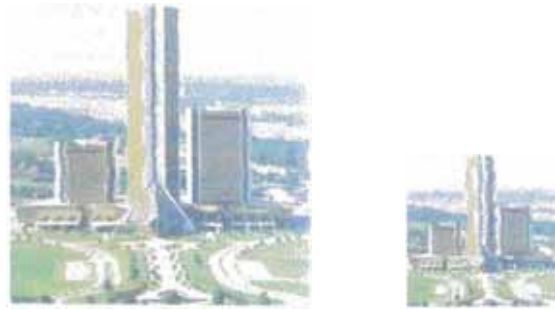


Imagen original

Escalado. $S_x=S_y=0.5$

Figura 3.8: Aplicando escaldado por 0.5 a una imagen en x,y.

c. **Rotación** (Figura 3.7.c): Se realiza girando el objeto alrededor de un eje, modificando así su orientación en el espacio (ver figura 3.9).

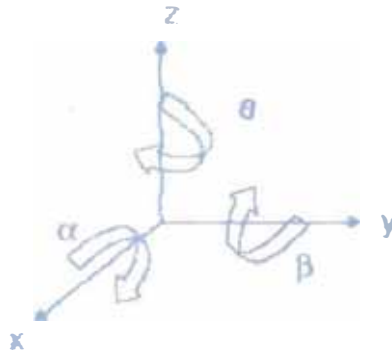


Figura 3.9: Referencia de coordenadas para la aplicación de rotación.

La rotación de un punto, un ángulo θ alrededor del eje de coordenadas Z se consigue con la transformación:

$$R_{\theta} = \begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \text{sen}(\theta) & 0 & 0 \\ -\text{sen}(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3.26)$$

El ángulo de rotación θ se mide en el sentido de las agujas del reloj. Esta transformación afecta únicamente a los valores de las coordenadas x e y.

La rotación de un punto, un ángulo α sobre el eje x se consigue con la transformación:

$$R_{\alpha} = \begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & \text{sen}(\alpha) & 0 \\ 0 & -\text{sen}(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3.27)$$

La **rotación** de un punto, un ángulo β sobre el eje y se consigue con la transformación:

$$R_\alpha = \begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\beta) & 0 & -\text{sen}(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ \text{sen}(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3.28)$$

En la figura 3.10, se observa el efecto de la rotación de una imagen original. El ángulo de **medida** como se puede observar es en el sentido antihorario.

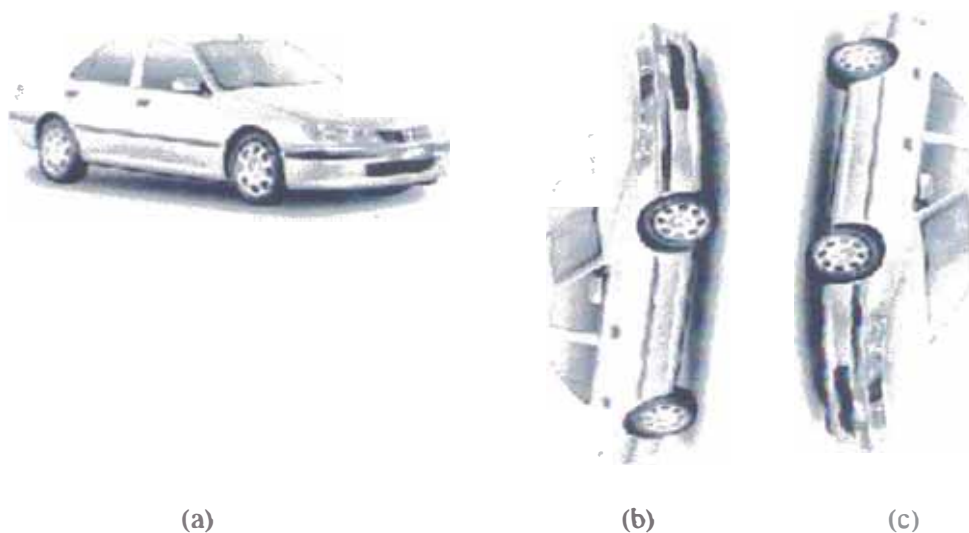


Figura 3.10: Rotación de la imagen en el plano ZY.
 (a) **Imagen** original. (b) Rotación de 270°. (c) Rotación de 90°.

Sobre un mismo punto podemos encadenar una tras otra distintas transformaciones. En esta **serie** de transformaciones, la resultante de la transformación precedente es **multiplicada** por la **matriz** correspondiente a la transformación siguiente.

Esta matriz producto representa la composición de todas esas transformaciones. Es **muy importante** apreciar que el **orden** en que se aplican las transformaciones afecta al **resultado final**.

Al encadenar algunas transformaciones, algunas no son conmutativas. Por ejemplo, es posible encadenar **[traslación ► escalado ► rotación]**. Pero no múltiples rotaciones.

3.3.3 El color [01][02][03][04][05]

Como se ha mencionado anteriormente, el color de un objeto no depende únicamente del **propio** objeto, sino también de las fuentes de luz que lo iluminan y del resto de objetos que **lo rodean**. Pero también depende de la percepción visual humana. Las imágenes que

percibimos están compuestas de ondas electromagnéticas que percibimos como colores. A diferentes longitudes de onda, diferentes sensaciones de color. Una imagen visible por un humano, se obtiene capturando la radiación proveniente de un objeto dentro del espectro visible. Es posible capturar radiación fuera del espectro visible, con lo que obtendríamos información no visible por un observador que, debidamente tratada, proporcionaría información extra (ver figura 3.11).

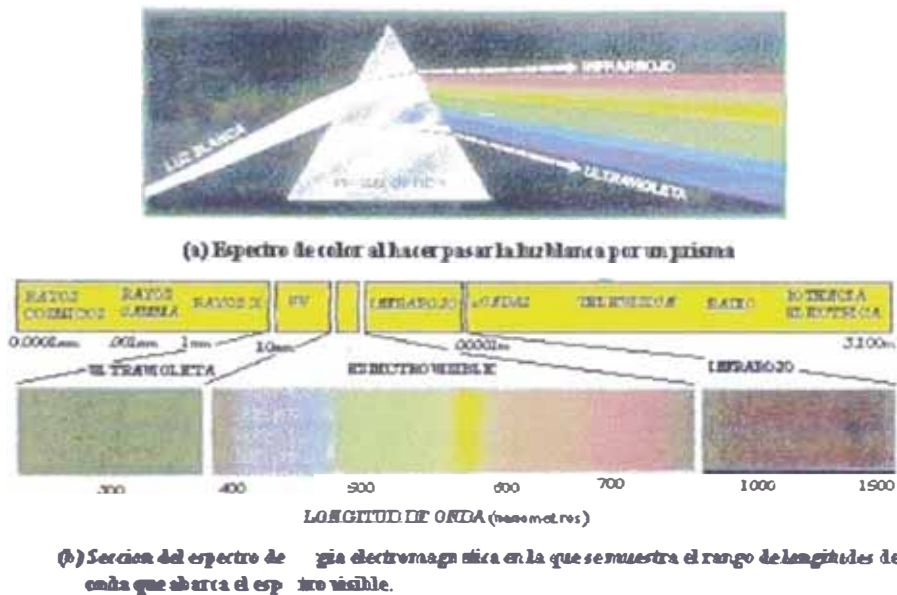


Figura 3.11: Descomposición de la luz y espectro electromagnético.

Un **espacio o modelo de color** es un sistema de coordenadas diseñado para permitir medir colores cuantitativamente. Los espacios de color pueden ser:

- **Triestímulo:** Se basan en la existencia de tres colores básicos ó primarios y el resto de colores se representa mediante una combinación lineal de ellos. Los espacios *triestímulo* son muy adecuados para la adquisición de imágenes y visualización, puesto que se corresponde con el sistema visual humano.
- **Cromático:** Se basan en la utilización de atributos cromáticos para representar los colores.

Un **color** no se corresponde a una única longitud de onda. En general un color suele caracterizarse por 3 valores:

- **Brillo (Brightness, Lighthness):** Es la cantidad de luz que percibimos. Está relacionado con la noción cromática de intensidad. En un objeto puesto al sol o a la sombra, sólo cambia el brillo.
- **Tono o Matiz (Hue):** Es un atributo que permite decir que un objeto es azul,

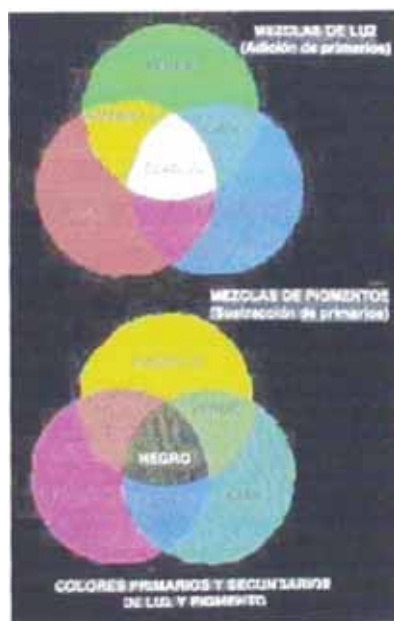
amarillo, etc. El tono está asociado a la longitud de onda dominante del espectro de radiación del cuerpo observado.

- **Saturación:** Informa de la pureza del color, es decir, nos da una medida del grado en que un color puro está diluido en luz blanca. Si mezclamos pintura roja con gris, obtendremos pintura roja con distinta saturación.

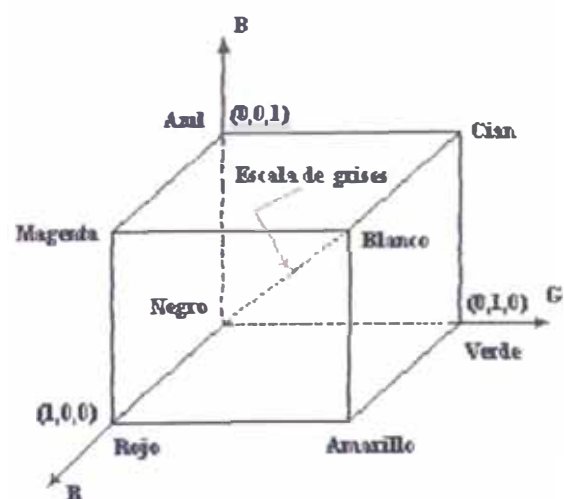
■ Modelo RGB

Es el usado por los monitores CRT y la mayoría de las videocámaras. Este modelo de color está basado en la teoría de la existencia de 3 colores primarios: rojo (R), verde (G) y azul (B). Cualquier otro color puede ser obtenido sumando adecuadamente estos tres colores primarios. Es posible representar los colores en el espacio RGB mediante un cubo RGB, donde cada una de las aristas corresponde a un color primario.

En la figura 3.12.a, se muestra dicha representación. El espacio de colores RGB se especifica mediante un sistema 3D de coordenadas cartesianas. El subconjunto de este espacio que representa los colores visibles para el ojo humano se corresponde con un cubo unidad. Uno de los vértices es el origen de coordenadas y representa el color negro. El color opuesto es el color blanco cuyas coordenadas son (1, 1, 1). Los colores primarios rojo, verde y azul también se corresponden con vértices del cubo.



(a)



(b)

Figura 3.12: Colores visibles por el ojo humano.
(a) Combinación de colores primarios. (b) Espacio de colores RGB.

Sea el caso de tres fuentes luminosas que emiten luz con las siguientes características:

Fuente 1: Color (1.0, 0.2, 0.1)

Fuente 2: Color (0.1, 1.0, 0.1)

Fuente 3: Color (0.1, 0.1, 1.0)

Los colores obtenidos en cada una de las regiones son, ver figura 3.13:

A: Rojo más oscuro (1.0, 0.2, 0.1)

B: Violeta claro (0.1, 0.1, 1.0)

C: Verde más oscuro (1.0, 1.0, 0.2)

D: Blanco (1.0, 1.0, 1.0)

E: Verde azulado (0.2, 1.0, 1.0)

F: Blanco (1.0, 1.0, 1.0)

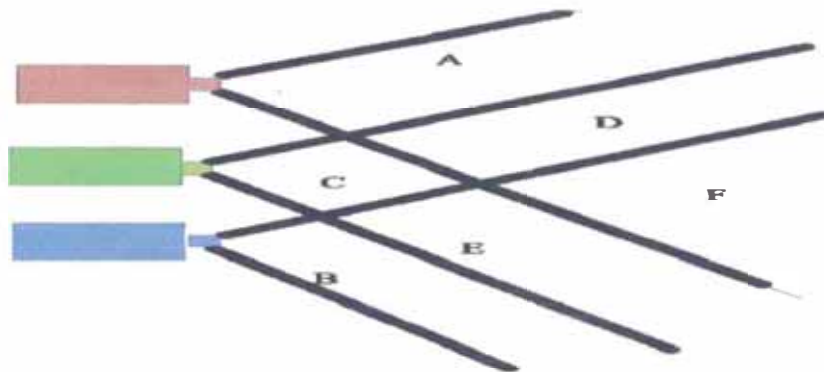


Figura 3.13: Combinación de colores de 3 fuentes.

El problema del espacio de color RGB es que en los 3 valores del color va mezclada tanto la información del color (tono y saturación) como de la intensidad.

3.4 Adquisición y almacenamiento de imágenes

3.4.1 Introducción [02] [03]

Una imagen se produce cuando diversas formas de radiación, las cuales han sido afectadas por objetos físicos, inciden y son registradas por un sensor.

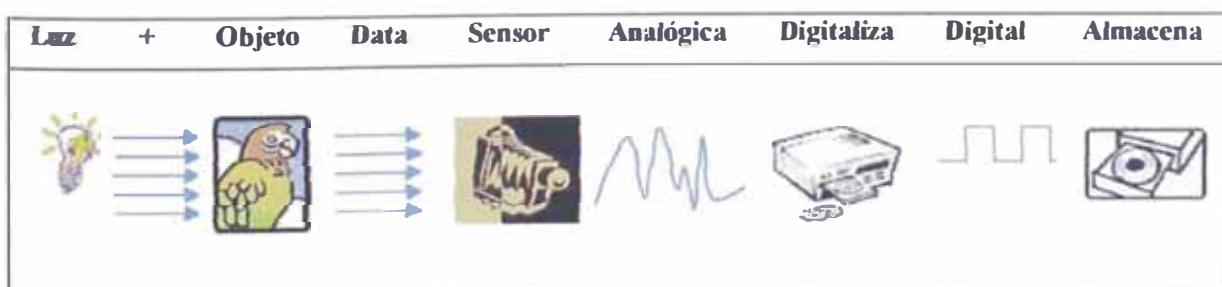


Figura 3.14: Captura, digitalización y almacenamiento de imagen.

El **sensor** convierte la información visual en señales eléctricas analógicas que son **muestreadas** espacialmente y cuantificadas en **amplitud** dando lugar a una señal digital.

Una vez adquirida la imagen y convertida a formato digital, será importante **almacenarla** en un formato que permita reducir el espacio de almacenamiento.

3.4.2 Técnicas de iluminación [02]

La **iluminación** es uno de los factores más importantes en TDI. **Normalmente va a determinar si un algoritmo va a funcionar o no. Suele afectar a la complejidad de los algoritmos de visión.**

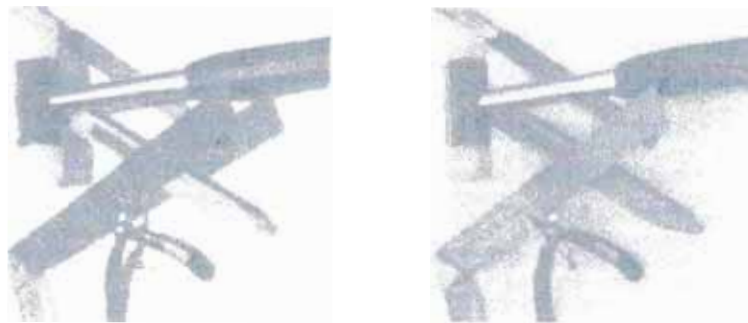


Figura 3.15. Dos imágenes de una escena bajo diferentes condiciones de iluminación

Las **técnicas más utilizadas para la iluminación de una escena** son:

- **Iluminación Difusa:** Trata de que al objeto le llegue la misma cantidad de luz en todos sus puntos. Se suele utilizar en aplicaciones donde se quiere analizar las características de la superficie del objeto (textura, color, etc.). Muy útil cuando el objeto tiene superficies suaves y regulares.
- **Iluminación a Contraluz o Retroiluminación:** Se emplea cuando la información a **extraer** reside en la silueta (contorno) del objeto, y no en la superficie del mismo. Por **ejemplo**, para medir superficies.
- **Iluminación Estructural:** Consiste en colocar un patrón delante de la fuente luminosa de forma que las características del patrón queden en el objeto, de forma que se pueda extraer información tridimensional del mismo.
- **Iluminación Direccional:** Consiste en iluminar el objeto de forma rasante. Es útil para inspeccionar la superficie de los objetos en lo referente a desperfectos o irregularidades que pudieran presentar (las imperfecciones se agrandan)

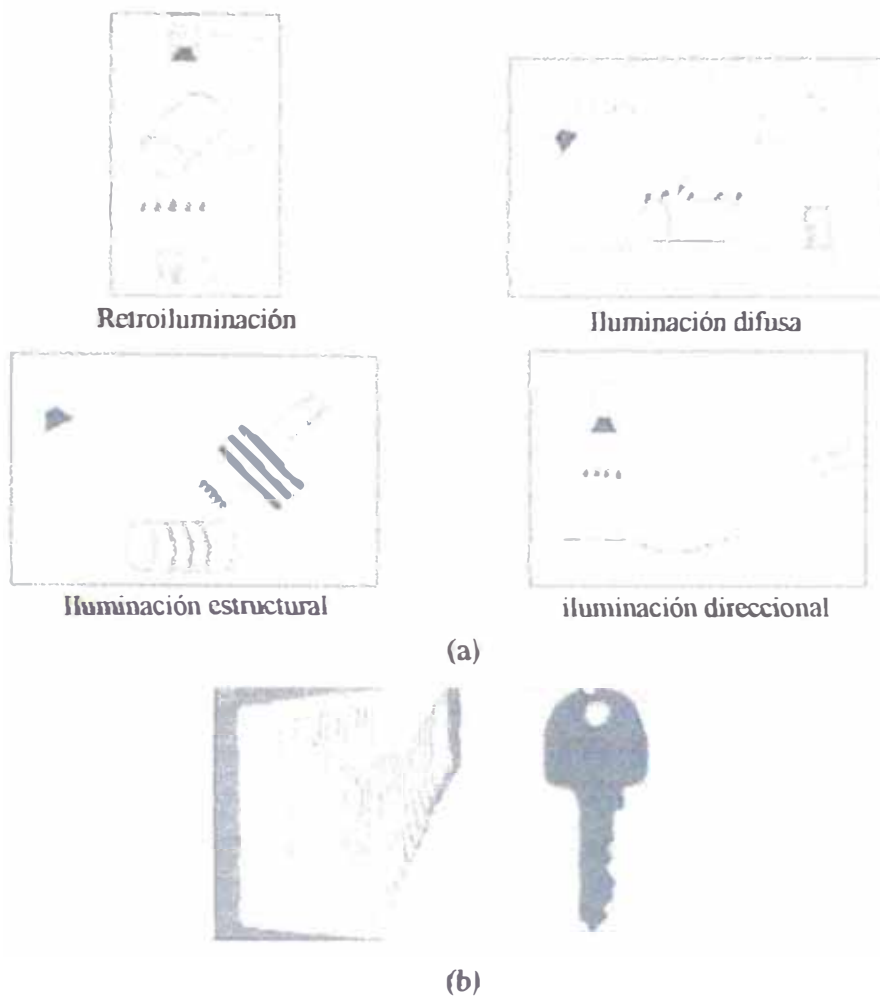


Figura 3.16: Técnicas de iluminación. (a) Diferentes tipos de iluminación. (b) Imagen proyectada con un patrón de luz consistente en franjas. Imagen de llave obtenida mediante retroiluminación.

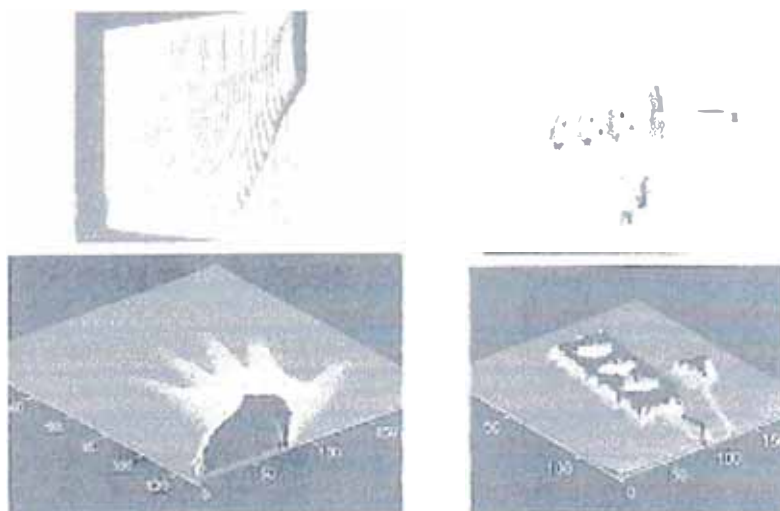


Figura 3.17: Técnicas de iluminación. Mapas de elevación de dos imágenes conseguidos utilizando iluminación estructural.

3.4.3 Ejemplos de Formatos de Imágenes

Formato BMP [04] [05] [06]

Los archivos de mapas de bits (bitmaps) de Windows están almacenados en formato **DIB (Device Independent Bitmap)** que permite a Windows visualizar el bitmap en cualquier tipo de dispositivo de visualización (monitor, impresora, etc.). El término **Device Independent** significa que el bitmap especifica el color del píxel de forma independiente del método utilizado por el visualizador para representar el color. La extensión por defecto del nombre de un archivo DIB es **.BMP**.

Un archivo BMP consta de 4 partes: una estructura **BITMAPFILEHEADER** seguida de otra estructura **BITMAPINFOHEADER**, de la tabla de colores (o paleta) y de un array de bits que forman la imagen. El archivo tiene la siguiente forma:

BITMAPFILEHEADER	BmFH;
BITMAPINFOHEADER	BmIH;
RGBQUAD	Colores[];
BYTE	BitmapBits[];

La estructura **BITMAPFILEHEADER** contiene información acerca del tipo y tamaño del archivo que contiene el mapa de bits, así como de la colocación del mismo dentro del fichero (todo lo que se escribe después de // es solo comentario, no tiene significado para el compilador):

```
typedef struct tagBITMAPFILEHEADER
{
    WORD        bfType; // Indica el tipo de fichero. Debe ser "BM" (ó 0x4d42)
    DWORD       bfSize; // Indica el tamaño en bytes del archivo.
    WORD        bfReserved1; // Reservado para uso futuro. Debe valer 0.
    WORD        bfReserved2; // Reservado para uso futuro. Debe valer 0.
    DWORD       bfOffBits; // Indica el desplazamiento desde el comienzo del
} BITMAPFILEHEADER; // archivo hasta donde empiezan los bits del bitmap.
```

La estructura **BITMAPINFOHEADER** contiene información sobre las dimensiones y el formato de color:

```
typedef struct tagBITMAPINFOHEADER
{
    DWORD       biSize; // Tamaño en bytes requerido para estructura
                // BITMAPINFOHEADER.
    LONG        biWidth; // Anchura de mapa de bits en píxeles (nº de columnas)
    LONG        biHeight; // Altura del mapa de bits en píxeles (número de filas)
    WORD        biPlanes; // Número de planos de la imagen. Siempre es 1
    WORD        biBitCount // Nº de bits por píxel de la imagen(1,4,8,16,24 ó 32)
    DWORD       biCompression; // Tipo de compresión de imagen: BI_RGB (no
```

```

// hay compresión), BI_RLE4 y BI_RLE8
DWORD    biSizeImage; // Tamaño en bytes del mapa de bits.
LONG     biXPelsPerMeter; // Resolución horizontal el píxeles por metro.
LONG     biYPelsPerMeter; // Resolución vertical en píxeles por metro.
DWORD    biClrUsed; // Número de colores de la paleta de colores que son usados.
//Si vale cero indica que utiliza todos los
// colores de la paleta (2^biBitCount)
DWORD    biClrImportant; //Número colores considerados importantes

} BITMAPINFOHEADER; // para ver el bitmap. Suele ser biClrUsed.

```

El tercer componente es la tabla de colores. Un array de estructuras RGBQUAD define la tabla de colores. Cada estructura define un valor de color RGB. El tamaño del array viene definido por el número de colores del bitmap. Así su tamaño puede ser calculado como 2 elevado al valor indicado en BITMAPINFOHEADER.biBitCount. La estructura RGBQUAD define un color en su intensidad de rojo, verde y azul:

```

typedef struct tagRGBQUAD
{
    BYTE    rgbBlue; // Indica la intensidad de azul en el color.
    BYTE    rgbGreen; // Indica la intensidad de verde en el color.
    BYTE    rgbRed; // Indica la intensidad de rojo en el color.
    BYTE    rgbReserved; // Reservado. Debe ser cero.
} RGBQUAD;

```

La estructura BITMAPINFO combina la estructura BITMAPINFOHEADER y la tabla de colores para proporcionar una definición completa de las dimensiones y de los colores del DIB.

```

typedef struct tagBITMAPINFO
{
    BITMAPINFOHEADER    bmiHeader;
    RGBQUAD              bmiColors[1]; // Matriz de tipo RGBQUAD
} BITMAPINFO;

```

Sólo las imágenes BMP con 2, 16 ó 256 niveles de intensidad utilizan paleta. En imágenes que dedican 16, 24 ó 32 bits/píxel, el valor del píxel indica directamente el color de ese punto.

En cuanto al mapa de bits, los píxeles son almacenados por filas, de izquierda a derecha dentro de cada fila. Las filas son almacenadas de abajo a arriba; el origen del bitmap se encuentra en la esquina inferior izquierda.

La alineación para cada fila es DWORD (llenar cada fila hasta completar grupos de 4 bytes).

A continuación un modelo de un bitmap de 16 colores (4 bits por píxel):

BitmapFileHeader

bfType	19778
bfSize	3118
bfReserved1	0
bfReserved2	0
bfOffsetBits	118

BitmapInfoHeader

biSize	40
biWidth	80
biHeight	75
biPlanes	1
biBitCount	4
biCompression	0
biSizeImage	3000
biXPelsPerMeter	0
biYPelsPerMeter	0
biColorsUsed	16
biColorsImportant	16

Tabla de Colores (Paleta)

	Azul	Verde	Rojo	No Usado
[00000000]	84	252	84	0
[00000001]	252	252	84	0
[00000002]	84	84	252	0
[00000003]	252	84	252	0
[00000004]	84	252	252	0
[00000005]	252	252	252	0
[00000006]	0	0	0	0
[00000007]	168	0	0	0
[00000008]	0	168	0	0
[00000009]	168	168	0	0
[0000000A]	0	0	168	0
[0000000B]	168	0	168	0
[0000000C]	0	168	168	0
[0000000D]	168	168	168	0
[0000000E]	84	84	84	0
[0000000F]	252	84	84	0

- **Técnicas en el dominio de frecuencias:** transforman la imagen a una representación frecuencial compuesta por sinusoides complejas. Están basadas fundamentalmente en la aplicación de la Transformada de Fourier.

■ Dominio del espacio

Tienen la forma general:

$$g(x,y) = T[f(x,y)] \quad (3.29)$$

donde:

- $f(x,y)$ \implies Imagen original
- T \implies Transformación (Operador)
- $g(x,y)$ \implies Imagen resultante.

El operador T puede operar en un ámbito **puntual**, **local** o **global**.

- **Ámbito Puntual:** cada píxel de la imagen resultado se obtiene como combinación de un único píxel de la imagen original.

Por ejemplo, aplicando ruido a la imagen $g(x,y) = a f(x,y)$, ver figura 3.18.

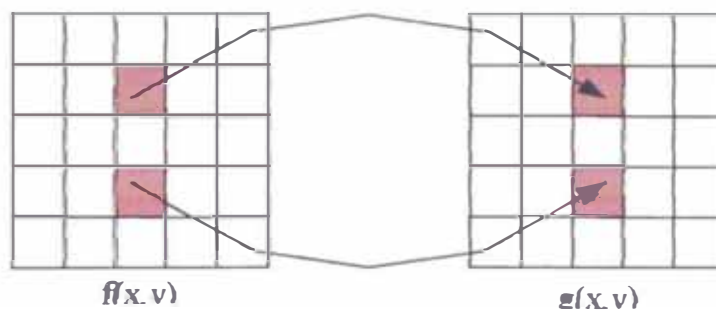


Figura 3.18: Ambito puntual

Una **manipulación puntual** sobre una imagen $N \times N$ necesita $N \times N$ lecturas y $N \times N$ escrituras, pudiéndose aplicar el resultado directamente sobre la imagen original, es decir, no se necesita de una **imagen auxiliar**.

- **Ámbito Local:** para cada píxel destino se leen varios píxeles origen. Se define una **ventana** que comprende varios píxeles de la imagen origen y el resultado de operar éstos es un único píxel en la imagen resultante. A esta operación se le denomina **convolución o filtrado de imagen**.

La **convolución** es una operación que **modifica** el nivel de gris de los píxeles de la **imagen** teniendo en cuenta los píxeles de su entorno de vecindad. Es una operación fundamental como herramienta para el **preprocesado de imágenes** y en **detección de bordes**.

El **producto** de convolución entre dos señales bidimensionales discretas f y h se define como:

$$g(x,y) = f(x,y) * h(m,n) = \sum_m \sum_n f(x-m,y-n)h(m,n) \quad (3.30)$$

donde las sumatorias se extienden sobre el área de solapamiento de las dos señales. En el caso de imágenes digitales, f es la imagen a procesar mientras que h es la denominada **máscara o filtro de convolución**. El tamaño de la máscara puede ser muy variado, desde las más comunes de 3×3 hasta máscaras enormes.

Si el tamaño de la máscara es V , realizar una manipulación local sobre la imagen supondrá $N \times N \times V$ lecturas y $N \times N$ escrituras, **no pudiéndose aplicar directamente sobre la imagen original, pues se sobrescribiría información necesaria.**

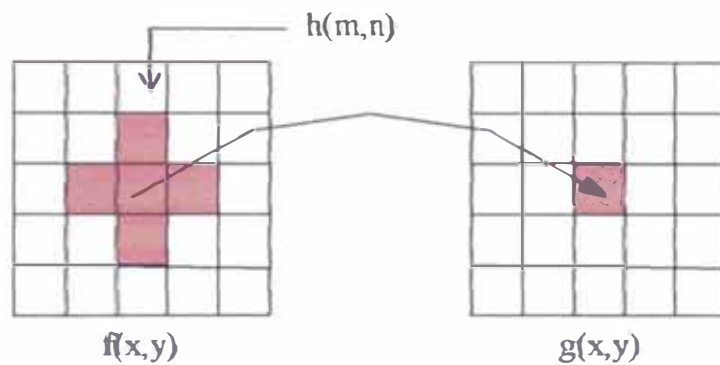


Figura 3.19: Ambito local

- **Ámbito Global:** cada píxel de la imagen destino es resultado de operar con todos los de la imagen original.

En este caso son necesarias $N \times N \times N \times N$ lecturas y $N \times N$ escrituras para manipular una imagen con una operación global. Se necesita una capacidad de proceso muy grande.

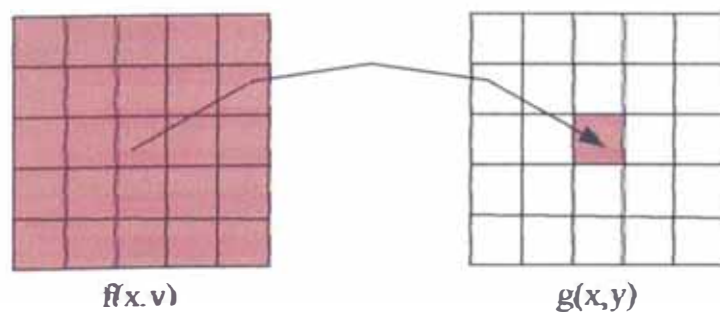


Figura 3.20: Ambito global

3.5.3 Técnicas de realce [01] [02]

El **realce local** ó **realce global** según la técnica en cuestión se aplica sólo a una zona de la imagen o a toda la imagen completa. La mayor parte de las técnicas de realce son técnicas puntuales.

■ **negativo**

El **negativo** de una imagen digital se obtiene aplicando la función:

$$g(x,y) = L - 1 - f(x,y) \quad (3.31)$$

donde L es el nº de niveles de gris de la imagen, ver figura 3.21.

Se trata de un procedimiento de inversión de la escala de grises, muy utilizado en imágenes médicas.

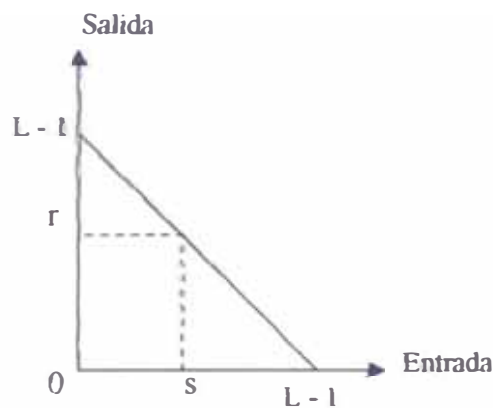


Figura 3.21: Proceso de inversión de escala de grises.

La figura 3.21 ilustra el empleo de esta sencilla transformación (r y s indican los niveles de gris de entrada y de salida respectivamente)

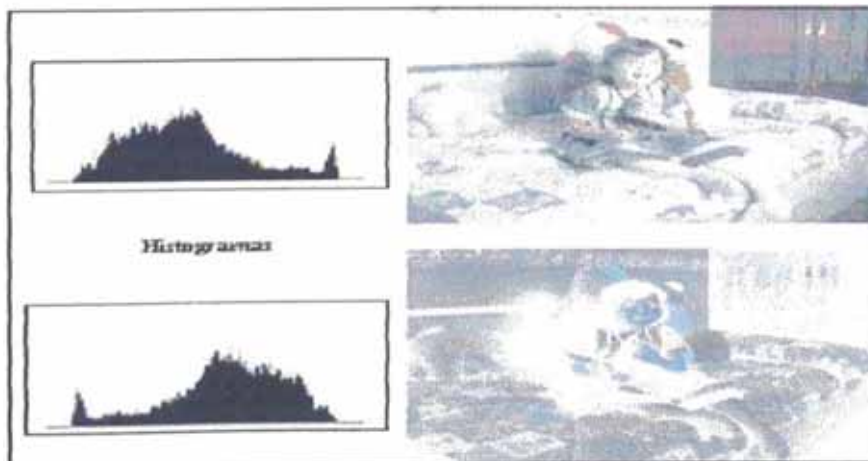


Figura 3.22: Resultado de la inversión de color.

■ Desplazamiento del Histograma [05]

Se usa para aclarar u oscurecer una imagen pero manteniendo la relación entre los valores de los niveles de gris. Esta operación se efectúa sumando o restando un valor fijo a todos los valores de la imagen de la forma:

$$g(x,y) = f(x,y) + \text{valor con } \text{valor} \in Z \quad (3.32)$$

Cuando los valores de $g(x,y)$ sobrepasen el máximo y el mínimo se redondea al máximo y al mínimo posibles permitidos. Si **valor** es positivo se incrementa el brillo de la imagen, mientras un valor negativo la oscurece.



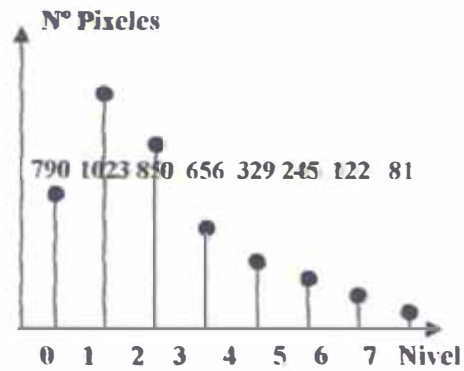
Figura 3.23: Efecto de desplazamiento con valores -30 y 30.

■ Ecuación del Histograma [01] [02]

El objetivo que se busca con la ecualización (igualación) es conseguir un histograma lo más plano posible, es decir, con el máximo contraste posible.

El método de ecualización consiste en calcular la frecuencia relativa de aparición de cada nivel de intensidad en la imagen, a partir de ella se estima la probabilidad de la intensidad de un determinado píxel de la imagen. A continuación, se tiene una imagen 64 x 64 píxeles con 8 niveles de gris/píxel, cuyo histograma aparece en la figura 3.24.

Nivel(k)	N°
0	790
1	1023
2	850
3	656
4	329
5	245
6	122
7	81



Datos de imagen 64x64

Histograma

Figura 3.24: Distribución e histograma de una imagen.

Nivel(k)	N° pixels(n_k)	Prob. $P_k = n_k/n$	P.Acum. S_k	$S_k * (L-1)$	Redondeo
0	790	0.19	0.19	1.33	1
1	1023	0.25	0.44	3.08	3
2	850	0.21	0.65	4.55	5
3	656	0.16	0.81	5.67	6
4	329	0.08	0.89	6.23	6
5	245	0.06	0.95	6.65	7
6	122	0.03	0.98	6.86	7
7	81	0.02	1	7	7
	$n=64 \times 64 = 4096$	Suma=1			

Figura 3.25: Ecuando el histograma.

siendo L el nº de niveles de intensidad distintos. Otro tipo de redondeo hubiera dado unos valores ligeramente diferentes. La función de transformación que se aplica pues a los píxeles y el histograma resultante se muestra en la figura 3.26.

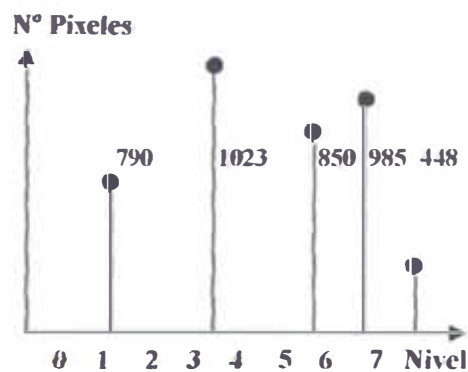
E	S	N	P
0	1	0	0
1	3	1	790
2	5	2	0
3	6	3	1023
4	6	4	0
5	7	5	850
6	7	6	$656+329=985$
7	7	7	$245+122+81=448$

E= Entrada

S= Salida

N = Nivel(k)

P= Pixeles(n_k)



Histograma

Figura 3.26: Imagen de la figura 3.24 ecualizada.

La transformación aplicada por este algoritmo simplemente reasigna niveles de grises a los píxeles, lo que significa que las frecuencias (barras) del histograma se desplazan en el eje de abscisas, ó se acumulan, pero nunca se fragmentan, lo que hace que el histograma no ~~aparezca~~ exactamente uniforme (plano).

La igualación del histograma se adapta perfectamente al realce automático, ya que la función de transformación viene determinada exclusivamente por el histograma de la imagen de entrada. A diferencia de otras técnicas de aumento del contraste, ésta resulta completamente automática.

Otra de las características de esta técnica consiste en el aprovechamiento automático de toda la escala de grises (siempre alcanza el blanco) por lo que es especialmente útil en imágenes con detalles ocultos en regiones de pocas tonalidades. Para imágenes de buena calidad, la ecualización a menudo provoca su degradación, no siendo aconsejable por tanto su utilización.

Además, es una técnica de una sola pasada; es decir, que aunque apliquemos el algoritmo más de una vez no obtendremos un resultado distinto.

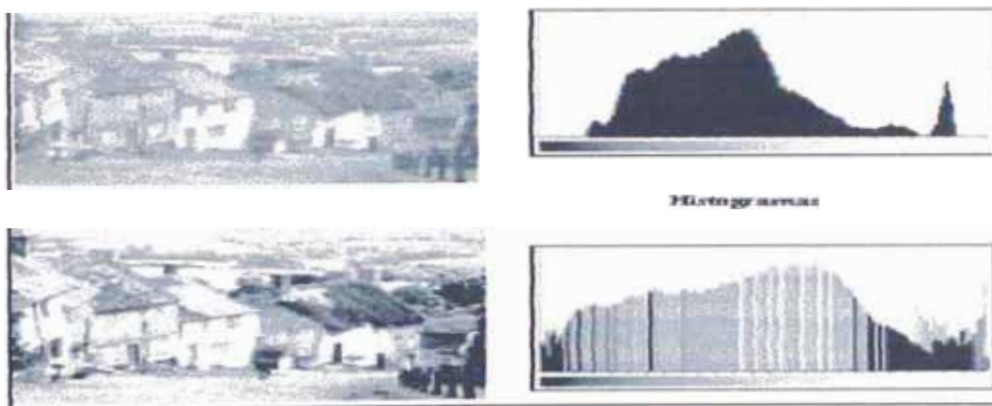


Figura 3.27: Imagen 512x512 y 256 niveles de gris y su correspondiente ecualización.

■ **Transformación Lineal de la Escala de Grises [02]**

Se trata de una técnica de aumento de contraste en donde para cada punto de la imagen de entrada se aplica una transformación lineal de la escala de grises obteniéndose el punto correspondiente en la imagen de salida:

$$g(x,y) = a f(x,y) + b \quad (3.33)$$

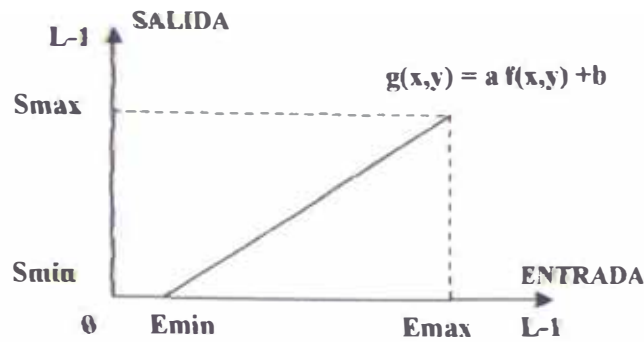


Figura 3.28: Transformación lineal de escala de grises.

A cada punto del intervalo (E_{min} , E_{max}) le corresponde uno del intervalo (S_{min} , S_{max}) en la forma:

$$g(x,y) = \frac{S_{max} - S_{min}}{E_{max} - E_{min}} (f(x,y) - E_{min}) + S_{min} \quad (3.34)$$

En general, si $a > 1$ entonces expande el rango, pero si $a < 1$ entonces comprime el rango.

Si deseamos transformar $[40,150] \implies [0,255]$,
 $a = (255 - 0) / (150 - 40) = 2,318$
 $g(x,y) = 2,318 * (f(x,y) - 40) + 0$,

De donde, para:

$f(x,y) = 40 \implies g(x,y) = 0$
 $f(x,y) = 88 \implies g(x,y) = 111$
 $f(x,y) = 123 \implies g(x,y) = 192$
 $f(x,y) = 150 \implies g(x,y) = 255$

■ Compresión de Rango Dinámico [01]

A veces el rango dinámico de la imagen excede la capacidad del dispositivo de presentación. Además de utilizar una transformación lineal con $a < 1$; una manera efectiva de comprimir el rango dinámico de los valores de cada píxel consiste en realizar transformación de intensidad para pasar del rango $[0, R]$ al $[0, L - 1]$:

$$g(x,y) = C \log_{10}(1 + |f(x,y)|) \quad (3.35)$$

con $C = (L - 1) / \max(\log_{10}(1 + |f(x,y)|))$

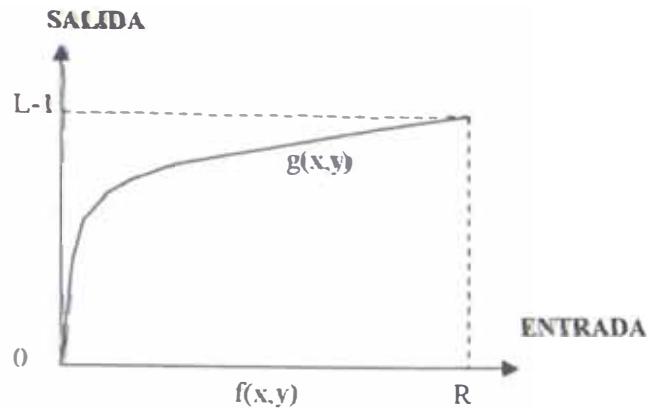


Figura 3.29: Compresión de rango no lineal.

La figura 3.29, muestra la función de transformación logarítmica.

Por ejemplo, si:

$$[0, 2.5 * 10^6] \implies [0, 255] , \quad (L=256)$$

$$C=255/\log_{10}(1+2.5 * 10^6)=39,84375 \implies g(x,y) = 39,84375 * \log_{10}(1-f(x,y))$$

■ Modificación del Contraste mediante funciones No Lineales [06]

Una serie de transformaciones muy usuales se llevan a cabo mediante las siguientes funciones no lineales:

- **Función Cuadrada:**

$$g(x,y) = \frac{f(x,y)^2}{L-1} \quad (3.36)$$

con $L = n^\circ$ de niveles de gris de la imagen

- **Función Cúbica:**

$$g(x,y) = \frac{f(x,y)^3}{L-1^2} \quad (3.37)$$

Con estas funciones las imágenes resultantes son más oscuras que la original. Las funciones no son lineales. Además, la información contenida en la imagen resultante es menor que en la original.

Si consideramos una imagen con 256 niveles de gris y aplicamos la función cuadrada al nivel de gris original 128 obtenemos 64 en la imagen nueva. Por ello, los píxeles oscuros de la imagen original son el 25% en la imagen nueva. Por ello la imagen resultante es más

oscura ya que ahora los píxeles claros se reparten el 75% de los niveles de gris y no el 50%. El contraste por tanto, entre los píxeles claros de la imagen mejora.

■ Fraccionamiento de los Planos de Bits [01]

En lugar de destacar ciertos rangos de intensidades, a veces puede desearse destacar la contribución que realizan a la imagen determinados bits específicos.

Así por ejemplo, una imagen con 8 bits/píxel estará compuesta por 8 planos de un bit que van del plano 0 para el bit menos significativo al plano 7 para el bit más significativo. En términos de bytes, el plano 0 contiene todos los bits de orden más inferior de los bytes que forman los píxeles de la imagen, mientras que el plano 7 contiene todos los bits de orden más superior. Se debe mencionar también que, los planos de mayor orden contienen los datos más significativos visualmente, mientras que; los de orden menor contribuyen a los detalles más finos de la imagen.

También se puede observar que el plano 7 se corresponde exactamente con una imagen binarizada cuyo umbral se ha tomado en el nivel de gris 128.

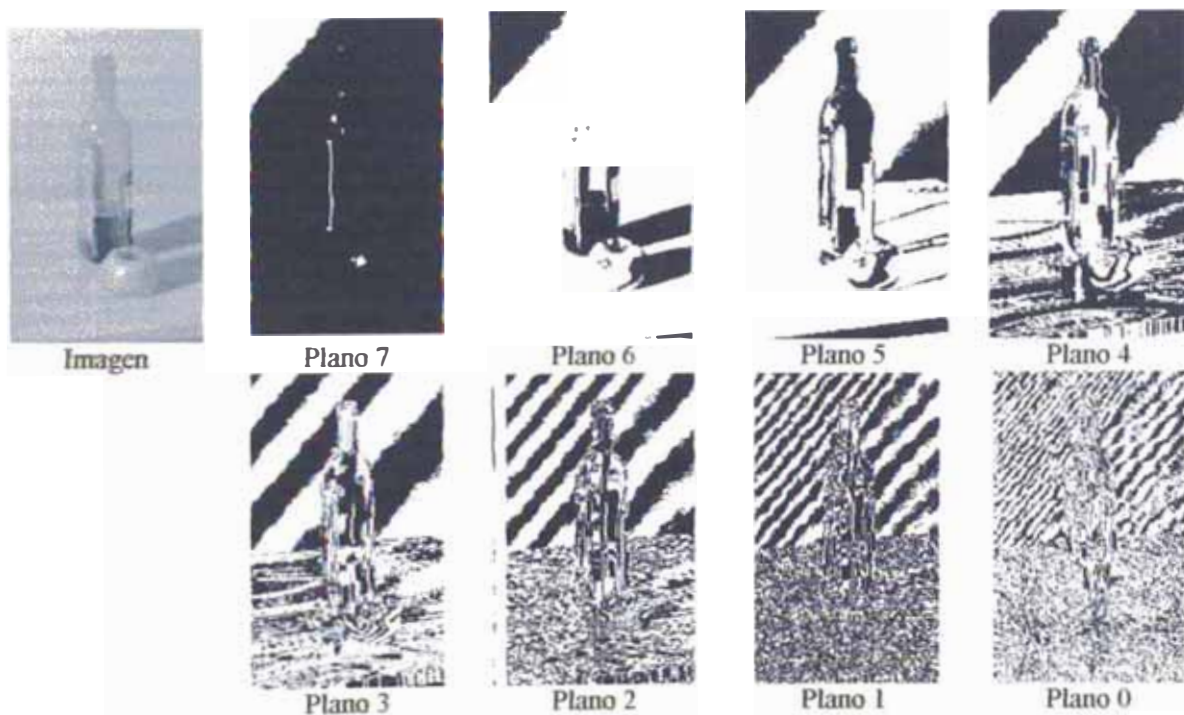


Figura 3.30: Imagen de 256 niveles de gris y sus correspondientes 8 planos de bits.

■ Realce local. Filtros realzantes [01] [02]

Las técnicas de procesamiento del histograma vistas, son técnicas globales. El objetivo principal del realce es el de destacar los detalles finos de una imagen o intensificar detalles

que han sido difuminados, bien por error o bien por efecto natural del método de adquisición de la imagen.

Aunque en el campo del procesamiento digital de imágenes es habitual trabajar en frecuencias, en aplicaciones de visión por computador estos procesos de realce suelen hacerse directamente en el dominio espacial, ya que el **costo computacional de la utilización de transformadas (usualmente la FFT) puede resultar excesivo.**

En la práctica, la forma más común de implementar un método de realce espacial (y también de suavizado) es a partir de la distribución de intensidades sobre un entorno de vecindad de cada píxel de la imagen. En otras palabras, lo que hace es pasarle a la imagen una **máscara de convolución**, definiendo una ventana cuadrada o rectangular y moviendo su **centro píxel a píxel** por la imagen, operando solamente en dicho entorno de acuerdo a atributos de las intensidades de sus píxeles.

Así, para una máscara 3 x 3 como la siguiente:

w ₁	w ₂	w ₃
w ₄	w ₅	w ₆
w ₇	w ₈	w ₉

Figura 3.31: Máscara de convolución.

Este procedimiento implica el cómputo de la **suma de productos** de los coeficientes de la máscara con los niveles de gris contenidos en la región de la imagen abarcados por la máscara. Así, la respuesta de la máscara en cualquier punto de la imagen es:

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9 = \sum_{i=1}^9 w_i z_i \quad (3.38)$$

donde z_i es el nivel de gris del píxel asociado con el coeficiente w_i de la máscara.

- **Media y varianza**

Dos de los atributos más utilizados son la media y la **varianza**. La **media de intensidades** nos da una **medida del brillo medio** mientras que la **varianza** proporciona una **estimación del contraste** de la región considerada de la imagen. Así pues, **cuanto mayor sea la media, mayor será el brillo medio y mientras mayor sea la varianza, mayor será el contraste.**

La varianza, σ^2 se define como:

$$\sigma^2 = \sum_x \frac{(x_i - \mu)^2}{n} \quad (3.39)$$

con x_i = valor del píxel en cuestión; μ = media y n = n° de píxeles.

A continuación se muestra una transformación típica basada en estos conceptos, donde los valores de intensidad de la imagen de entrada $f(i,j)$ son modificados píxel a píxel, obteniéndose una nueva imagen $g(i,j)$ de la siguiente forma:

$$g(i,j) = A(i,j) [f(i,j) - \mu(i,j)] + \mu(i,j) \quad (3.40)$$

con $A(i,j) = K * M / \sigma(i,j)$; $0 < K < 1$

donde:

$\mu(i,j)$ y $\sigma(i,j)$ son la media y la desviación típica (raíz cuadrada de la varianza) calculadas sobre entorno de vecindad centrado en el píxel (i,j)

M es la media global de $f(i,j)$

K es una constante que toma valores entre 0 y 1.

La utilización del término $A(i,j)$ permite amplificar las variaciones locales según una ganancia local. De esta forma, en zonas de bajo contraste $\sigma(i,j)$ será pequeña y, por tanto, la ganancia $A(i,j)$ grande, produciéndose valores elevados de $g(i,j)$.

En la función de transformación anterior se ha sumado la media $\mu(i,j)$ para de esta forma restablecer el brillo medio de intensidad.

Para evitar saturaciones en la intensidad resultante $g(i,j)$, la ganancia A suele limitarse dentro de un intervalo $[A_{min}, A_{max}]$

- **Filtro realzante** (filtro espacial de paso alta que realza las altas frecuencias) debe tener coeficientes positivos cerca de su centro y negativos en la periferia. La implementación más clásica de un filtro de realce 3 x 3 es:

	-1	-1	-1
1/9	-1	8	-1
	-1	-1	-1

Figura 3.32: Mascara de filtro realzante.

Obsérvese que la suma de los coeficientes es 0. Así cuando la máscara está sobre una zona de nivel constante o lentamente variable, la salida proporcionada por la máscara es 0 o un valor muy pequeño.

Además, la aplicación de este filtro reduce la media de niveles de gris de la imagen a 0, reduciendo significativamente el contraste global. La reducción del nivel medio de la imagen a 0 implica que la imagen debe tener valores negativos. Como sólo estamos considerando niveles positivos de gris, los resultados de este filtrado necesariamente

implican alguna forma de desplazamiento o cambio de escala para que al final los niveles de gris queden en el intervalo [0, L-1].

- **Filtrado High-Boost ó de énfasis de las frecuencias altas**

En general obtiene mejores resultados que el filtro anterior. La mascara es:

$$\frac{1}{9} \begin{bmatrix} -1 & -1 & -1 \\ -1 & W & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Figura 3.33: Mascara de filtro High-Boost.

Donde $w = 9A - 1$, con $A \geq 1$. El valor del nº real A determina la naturaleza del filtro

3.5.4 Promediado del Entorno de Vecindad o Filtro de la Media

Dada una imagen $f(x,y)$, el procedimiento consiste en generar una nueva imagen $g(x,y)$ cuya intensidad para cada píxel se obtiene promediando los valores de intensidad de los píxeles de $f(x,y)$ incluidos en un entorno de vecindad predeterminado, es decir, la imagen suavizada se obtiene como:

$$g(x,y) = \frac{1}{P} \sum_{(m,n) \in S} f(m,n) \quad (3.41)$$

donde S es un subconjunto de “p” píxeles situados en el entorno de (x,y), incluyendo (x,y).

Algunos filtros de media son:

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Figura 3.34: Mascaras de filtro mediana.

El problema del suavizado es que al eliminar las altas frecuencias también se eliminan los bordes.

Otras máscaras usadas en suavizado, pero que intentan respetar los bordes, son:



Figura 3.35: Otras máscaras de filtro mediana.

3.5.5 Filtro de la Mediana

Una de las dificultades principales del promediado del entorno es que desdibuja los contornos y otros detalles de la imagen. Una alternativa, especialmente cuando el objetivo es más bien la reducción de ruido que el difuminado, es el uso del filtro de la mediana, que consiste en reemplazar el nivel de gris de cada píxel por la mediana de los niveles de los vecinos.

Esta técnica no se implementa mediante una convolución de la imagen con una determinada máscara (método lineal), es más bien un método no lineal. Es decir:

$$\text{mediana} (f + g) \neq \text{mediana} (f) + \text{mediana} (g)$$

La mediana m de un conjunto de valores es tal que la mitad de los valores del conjunto quedan por debajo de m y la otra mitad por encima. Por ejemplo, en un entorno 3×3 , la mediana es el quinto mayor valor. Así, si tenemos:

{20,1,125,55,22,200,7,19,48} y ordenamos. {1,7,19,20,22,48,55,125,200}; la mediana será 22.

Algunas propiedades del filtro de mediana son:

- Es un filtro paso-baja que intenta eliminar ruido respetando las altas frecuencias, pero que dependerá de la imagen en cuestión el que proteja mejor los contornos o no.
- Es un método muy útil para eliminar píxeles aislados.
- Es un método bastante costoso de implementar, ya que para cada operación de salida conlleva una ordenación de números.

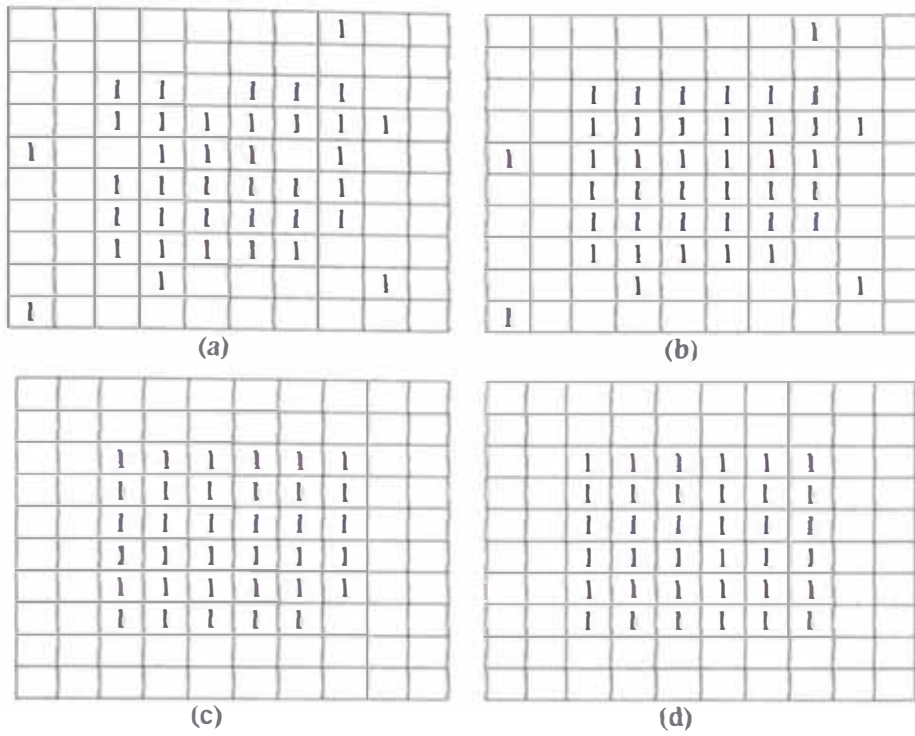


Figura 3.36: Aplicando operadores lógicos a una imagen.

(a) Imagen Original. (b) Resultado de aplicar B1.

(c) Resultado de aplicar B2. (d) Resultado final después de aplicar de B3 a B6.

CAPITULO 4

Algoritmos y Simulación del Sistema

4.1 Introducción

En este capítulo se muestran los algoritmos y programas respectivos en MatLab que permiten realizar diferentes procesos a partir de la imagen de una pieza mecánica. Debido a la lentitud en la ejecución de programas, y a la dependencia del entorno de programación cuando se trabaja en MatLab, se hace imprescindible implementar el sistema en otro lenguaje de programación, en este caso el Visual C⁺⁺.

Antes de realizar la implementación del sistema, el MatLab es utilizado como herramienta para verificar la funcionalidad y aplicabilidad de los diferentes algoritmos; esto es conveniente debido a la complejidad que puedan mostrar los algoritmos y la simplicidad del MatLab.

En los procesos de procesamiento digital de imágenes mostrados en esta etapa de simulación se utilizan imágenes con diferentes grados de calidad (propios del proceso de captura de imagen), para poner a prueba los algoritmos.

En la figura 4.1 se muestra el diagrama de flujo del sistema de control de calidad mediante el MatLab.

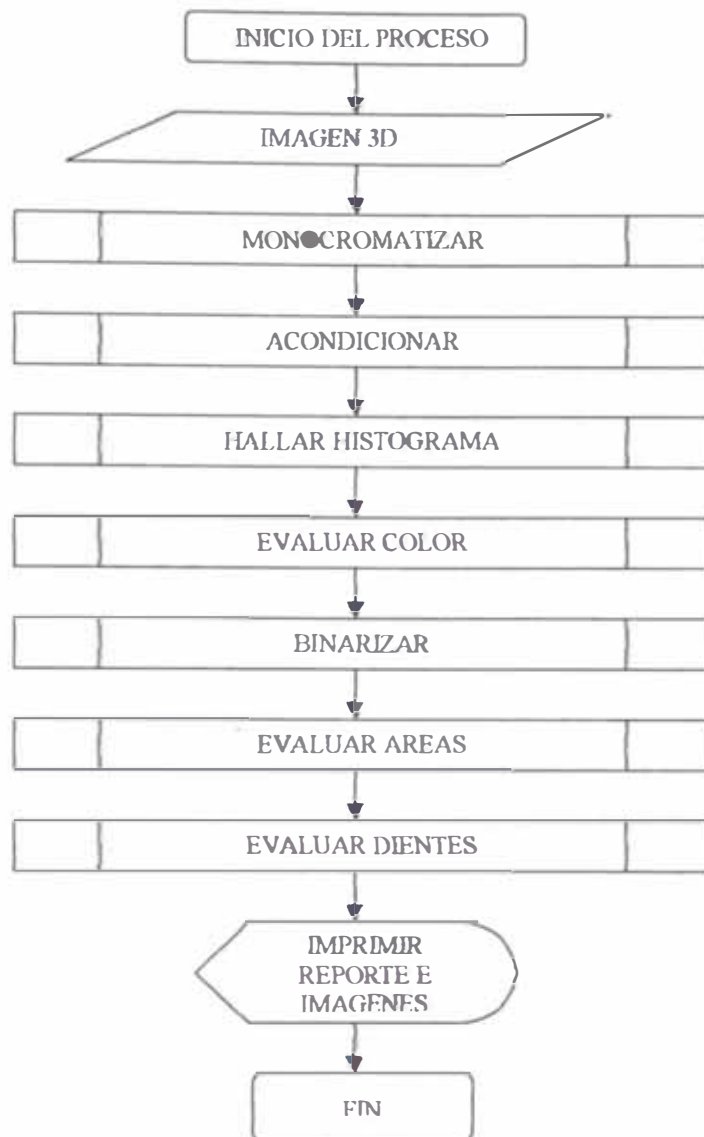


Figura 4.1: Diagrama de flujo del sistema de control de calidad en simulación.

4.2 Proceso parcial 01

4.2.1 Descripción del proceso

En esta etapa se realiza las siguientes tareas:

- **Cargar** la imagen a color en 3D. Luego llevarla a monocromático en 2D.
- **Obtener** el histograma de la imagen en 2D.
- **Binarizar** la imagen monocromática, y determinar el área de la pieza.
- **Calcular** el centro de masas.
- **Obtener** los contornos de la imagen.
- **Hallar** la firma de la pieza.

Este proceso se puede realizar con cualquier imagen digitalizada, independiente de la calidad de la misma. En la figura 4.2 se muestran algunas imágenes obtenidas con

diferentes dispositivos de adquisición, y allí se puede apreciar con claridad como varía la calidad de dichas imágenes de acuerdo a su fuente de adquisición.

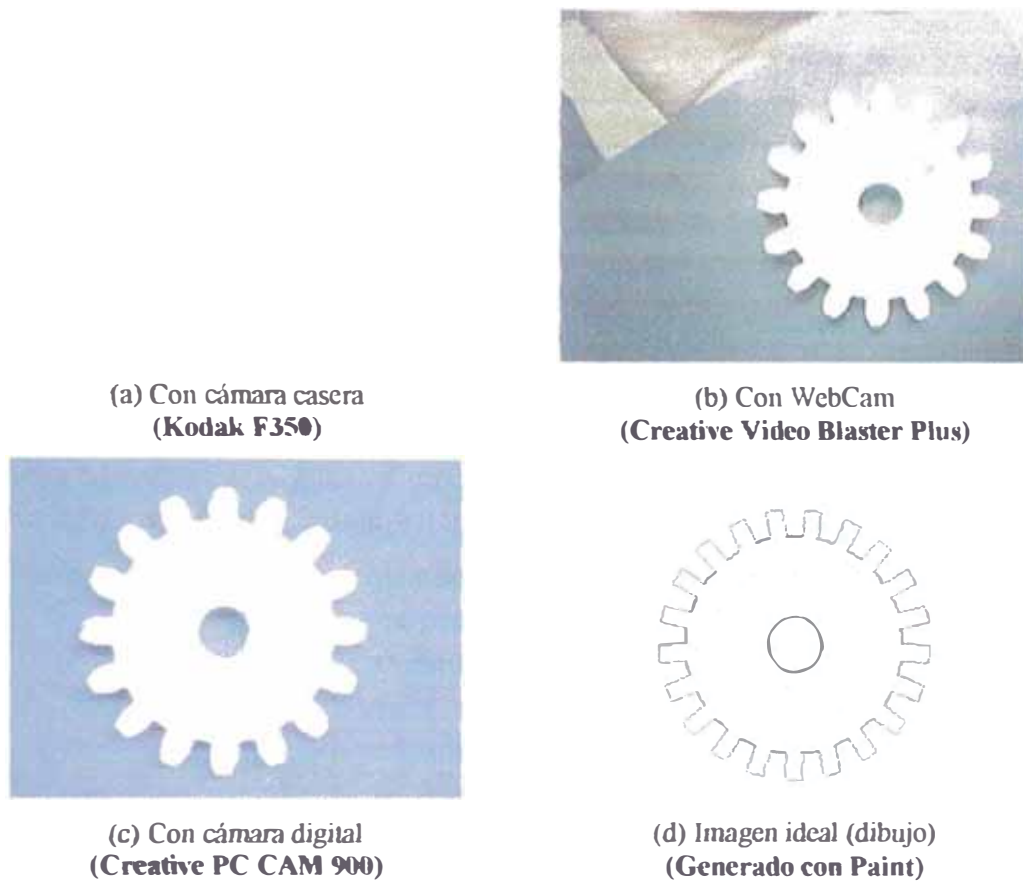


Figura 4.2: Imágenes obtenidas por diferentes medios.

La **calidad** de la imagen obtenida, la **técnica** del procesamiento digital, y además los **algoritmos** determinarán la **eficiencia** del sistema de control de calidad.

4.2.2 Algoritmo

El **algoritmo** describe los pasos a seguir para obtener los resultados exigidos por el proceso. Luego el algoritmo será implementado bajo la plataforma de programación del MatLab. A continuación se muestran los pasos a seguir:

1. Cargar la imagen a color del engranaje recto (EngRecto.bmp). Se obtiene la matriz A en 3D.

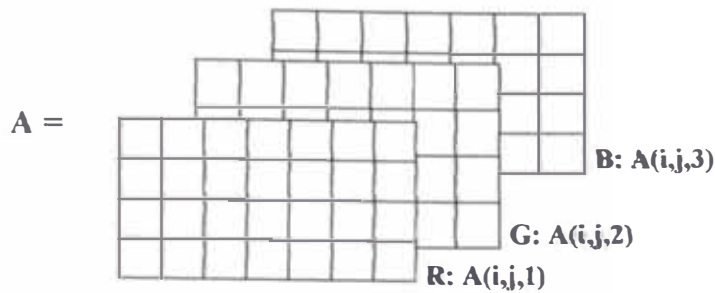


Figura 4.3: Representación matricial de imagen a color.

Donde : R = Componente Red (Rojo) de la imagen.

G = Componente Green (Verde) de la imagen.

B = Componente Blue (Azul) de la imagen.

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

$$\text{Max}(A) = 255$$

$$\text{Min}(A) = 0$$

$$\text{Size}(A) = M \times N \times 3. \text{ Ver figura 4.9.a.}$$

2. Llevar la imagen a niveles de gris, separando la componente Green(G) de la matriz

A.

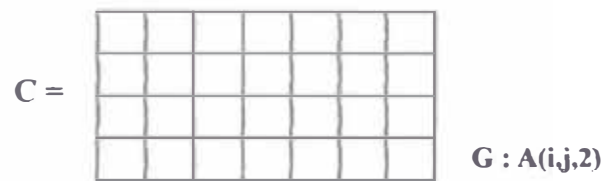


Figura 4.4: Representación matricial de imagen monocromática.

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

$$\text{Max}(C) = 255$$

$$\text{Min}(C) = 0$$

$$\text{Size}(C) = M \times N. \text{ Ver figura 4.9.b.}$$

3. Obtener el histograma de la imagen en niveles de gris.

3.1 $H(k) = 0$; Para $k = [1, 256]$.

Luego, $H(k) = [0 \ 0 \ 0 \ 0 \ \dots \ 0]$;

3.2 $H(C(i,j)+1) = H(C(i,j)+1) + 1$; Para $i=[1,M]$, $j=[1,N]$.

$\text{Size}(H) = 256$. Ver figura 4.9.c.

- 3.3 Calcular la media aritmética (μ):

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (4.1)$$

Donde, X_i es el nivel de gris de cada pixel [C(i,j)]

- 3.4 Calcular la varianza (σ^2):

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2 \quad (4.2)$$

- 3.5 Desviación estándar (σ):

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \quad (4.3)$$

3.6 Mediana (Md):

$$M_d = \begin{cases} x_{(n+1)/2} & ; \text{ si } n \text{ es impar} \\ \frac{1}{2}(x_{n/2} + x_{(n+2)/2}) & ; \text{ si } n \text{ es par} \end{cases} \quad (4.4)$$

3.7 Máximo valor: **max(max(C))**

3.8 Mínimo valor: **min(min(C))**

4. Binarizar la imagen en grises. El valor del umbral (en este caso 50), se obtiene a partir del histograma.

$$CC(i, j) = \begin{cases} 0 & ; C(i, j) \leq 50 \\ 255 & ; C(i, j) > 50 \end{cases} \quad (4.5)$$

Para $i=1$ a M

Para $j=1$ a N

Si $C(i, j) < 220 \implies CC(i, j) = 0$ (Negro)

Si $C(i, j) \geq 220 \implies CC(i, j) = 255$ (Blanco)

Size(CC) = $M \times N$. Ver figura 4.9.d.

5. Redefinir la matriz C, reduciendo el tamaño de la matriz de análisis separando el área de interés de los bordes que no contienen información. Mediante barrido horizontal y barrido vertical desde los extremos hacia el centro.

- 5.1 Determinar los subíndices máximos y mínimos (i_{min} , j_{min} , i_{max} , j_{max}) de la matriz C que contiene la imagen binarizada.

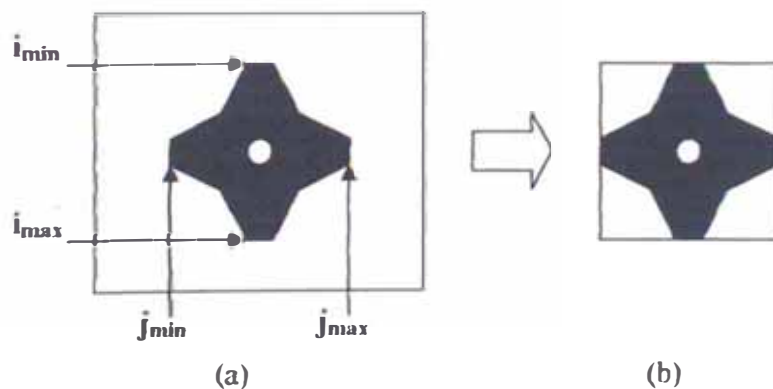


Figura 4.5: Redefiniendo la matriz de análisis. (a) Matriz C inicial. (b) Nueva Matriz C.

5.2 $C(ii, jj) = C(i, j)$;

Para $i = [i_{min}, i_{max}]$, $j = [j_{min}, j_{max}]$,

$ii = [1, (i_{max} - i_{min}) + 1]$, $jj = [1, (j_{max} - j_{min}) + 1]$

Size(C) = P×Q. Ver figura 4.8.e.
 Donde: P<M, Q<N.

6. Determinar el área de la imagen en número de píxeles.

Calculando la constante de proporcionalidad para áreas de la imagen,

$$Kp = \pi \times R^2 \quad (4.6)$$

$$Area = Kp \times \frac{Na}{Np} \quad (4.7)$$

donde: Kp=Constante de proporcionalidad (área del agujero).

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 cm²

6.1 Para i=1 a P

Para j=1 a Q

Si C(i,j) = 0

Ap = Ap + 1

Ap = 0.950 * Ap / Np

6.2 área libre : $Ab = PxQ - Ap$ (4 8)

$$Ab = 0.950 * Ab / Np \quad (4 9)$$

7. Calcular el centro de masas.

$$Xm = \frac{1}{NP} \sum_{j=1}^Q \sum_{i=1}^P i \quad ; \forall C(i, j) = 1 \quad (4.10)$$

$$Ym = \frac{1}{NP} \sum_{i=1}^P \sum_{j=1}^Q j \quad ; \forall C(i, j) = 1 \quad (4.11)$$

donde, (Xm, Ym) es el centro de masa del engranaje, y

NP=0 luego:

$$NP = \sum_{i=1}^P \sum_{j=1}^Q C(i, j) \quad ; \forall C(i, j) = 1 \quad (4.12)$$

Matriz C

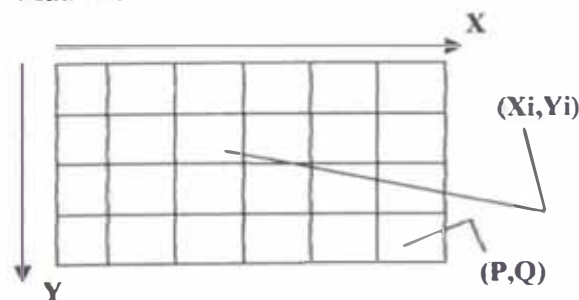


Figura 4.6: Referencia de coordenadas (sub índices) en la matriz.

Centro de masa: (Xm, Ym). Ver figura 4.9.e.

8. Obtener los contornos de la imagen.

8.1 $D(i,j) = 0$; Para $i = [1,P]$, $j = [1,Q]$.

8.2 Haciendo un barrido horizontal (Para $i = [1,P]$), capturar los puntos donde $C(i,j)=0$ y $C(i,j+1)=1$, en la matriz D. Ver figura 4.9.f.

$$D(i,j) = \begin{cases} 1 & ; \text{ para } A(i,j) + A(i+1,j) = 1 \\ 0 & ; \text{ para } A(i,j) + A(i+1,j) \neq 1 \end{cases} \quad (4.13)$$

8.3 Haciendo un barrido vertical (Para $j = [1,Q]$), capturar los puntos donde $C(i,j)=0$ y $C(i+1,j)=1$, en la matriz D. Ver figura 4.9.g.

$$D(i,j) = \begin{cases} 1 & ; \text{ para } A(i,j) + A(i,j+1) = 1 \\ 0 & ; \text{ para } A(i,j) + A(i,j+1) \neq 1 \end{cases} \quad (4.14)$$

Obteniéndose la matriz D:

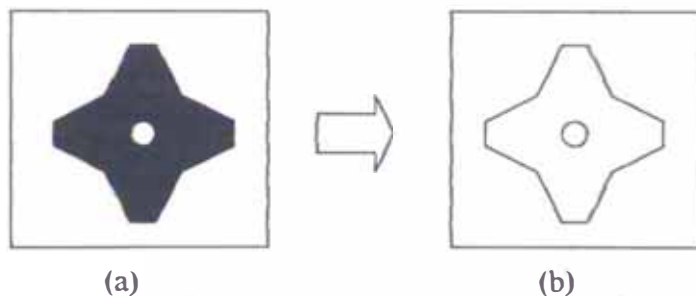


Figura 4.7: Obtención de la matriz de contornos. (a) Matriz C inicial. (b) Matriz D de contornos.

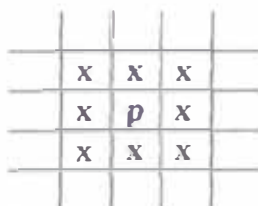
$Size(D) = P \times Q$.

9. A partir de la matriz de contornos(D) se detecta el perímetro:

9.1 Se define la matriz E para almacenar el perímetro:

$E(i,j) = 0$; Para $i = [1,P]$, $j = [1,Q]$.

9.2 Por barrido horizontal se detecta un primer punto de referencia (X_r, Y_r) . Luego se iguala a una coordenada de prueba, $(X_p, Y_p) = (X_r, Y_r)$ y se almacena en la matriz E.



Cada píxel "x" está en la vecindad del píxel "p".

9.3 Se busca un próximo punto en la vecindad del punto de prueba (X_p, Y_p) , igual a (X_{p+1}, Y_{p+1}) . Luego se iguala a la coordenada de prueba, $(X_p, Y_p) = (X_{p+1}, Y_{p+1})$ y se almacena en la matriz E.

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

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

$$\text{Donde: } k = [1, T]$$

$$T = \# \text{ de puntos del perimetro}$$

T se obtiene de la cuenta de los puntos que están en el contorno exterior.

9.4 Se repite el paso (9.3) hasta que:

$$(X_{p+1}, Y_{p+1}) = (X_r, Y_r).$$

9.5 Al ir eliminando el contorno de la matriz D, al final en esta queda almacenado el interior de la pieza.

Obteniéndose las matrices de la figura 4.8.

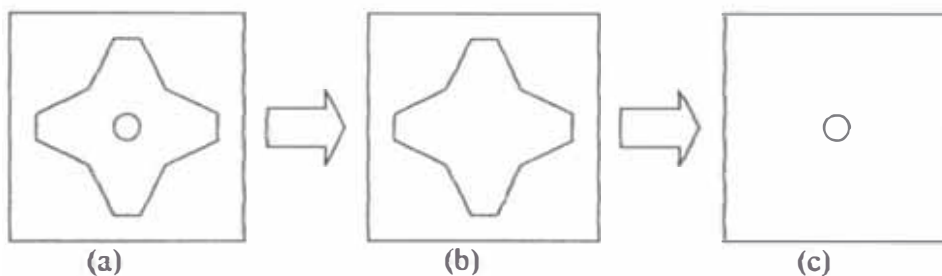


Figura 4.8: Obtención de perimetro y orificio central.
 (a) Matriz D inicial. (b) Matriz E de perimetro. (c) Matriz D de final.

Matriz E. Ver figura 4.9.h.

Matriz D. Ver figura 4.9.i.

10. Hallar la firma de la pieza:

10.1 Se almacena en matriz VE las coordenadas del borde de la pieza (Matriz E).

$$\text{Si } E(i,j)=1 \implies VE = \{VE; E(i,j)\}$$

10.2 Se halla las distancias euclidianas del centro de masa (X_m, Y_m) a los puntos del borde de la pieza (Matriz VE):

$$Vr(i) = \sqrt{(VE(i,1) - X_m)^2 + (VE(i,2) - Y_m)^2} \quad (4.15)$$

Ver figura 4.9.j.

11. Ubicar como referencia la menor distancia entre el borde (perimetro) y el centro de masa.

$$\text{menor} = V_k(1)$$

Para $i=1$ a k
 Si $\text{menor} > V(i)$
 $\text{menor} = V(i)$
 $ii=i$

Donde: k = número de puntos del borde
 ii = sub-índice del menor valor

Ver figura 4.9.k.

12. Mostrar la firma (donde la firma es la distancia del centro de masa a todos los puntos del borde) en forma sombreada para una mejor apreciación. Figura 4.9.l.

13. Mostrar parte de la firma, para poder observar en detalle los dientes del engranaje.

Ver figura 4.9.m y figura 4.9.n.

14. Mejorar la presentación de los dientes del engranaje.

Ver figura 4.9.o.

15. Agregar ruido aleatorio al perímetro del engranaje:

$$vrde(i) = vrd(i) + rand \quad ; \quad \text{tal que } 0 < rand < 1 \quad (4.16)$$

y hallar error del engranaje vrde respecto de vrd.

$$err(i) = vrde(i) - vrd(i) \quad ; \quad \text{para } i = [1, T] \quad (4.17)$$

Ver figura 4.9.p.

16. Emitir reporte final del proceso.

4.2.3 Código de Programa

En el Anexo A se muestra el código fuente que corresponde al algoritmo previamente desarrollado. PROGRAMA33: Engra01.m.

Para la ejecución del programa se utilizó la imagen de la figura 4.1.d.

4.2.4 Ejecución del programa

Luego de la codificación el programa se almacenó con el nombre engra01.m, posteriormente se ejecutó desde el MatLab, allí se visualizan algunos resultados parciales y también algunas figuras que se van generando, todas estas de acuerdo a los requerimientos del programa. Finalmente el programa emite un **reporte final**, la cual contiene información relacionada al proceso y que será utilizada en la siguiente etapa.

» engra01
 CARGANDO IMAGEN EN 3D...

Size(A)=

ans = 207 199 3

EXTRAYENDO IMAGEN EN 2D...

Size(C)=

ans = 207 199

HALLANDO HISTOGRAMA IMAGEN GRIS-2D...

Size(H)=

ans = 1 256

UMBRALIZANDO IMAGEN EN 2D...

REDUCIENDO DATA DE INTERES EN 2D...

xy =

14 198 8 192

Size(C)=

ans = 185 185

CALCULANDO AREA DE LA PIEZA...

área de la pieza=

Ap = 20302

área en blanco =

Ab = 13923

CALCULANDO EL CENTRO DE MASA...

xm =

93

ym =

93

HALLANDO CONTORNOS DE IMAGEN 2D...

Size(D)=

ans = 185 184

HALLANDO PERIMETRO DE IMAGEN 2D...

x1 =

3

y1 =

77

HALLANDO DISTANCIAS DEL C.M. AL BORDE...

Size(VE)=

ans = 1075 2

ans = 183 183

ans = 3 3

DESPLAZANDO FIRMA A PARTIR DE MENOR DISTANCIA...

GENERANDO ERROR ALEATORIO EN EL BORDE...

Presione una tecla para ver reporte final...

REPORTE FINAL	
área de interés	: Matriz C
Contornos	: Matriz PASO
Borde	: Matriz E
Interior	: Matriz I
Puntos del Borde	: Vector VE
Firma	: Vector vr
Firma normalizada	: Vector vrd
Centro de Masa	:
	ym(fil)= 93, xm(col)= 93
Menor distancia CM-Borde	: 71
Mayor distancia CM-Borde	: 91
área de la pieza	: 20302 Puntos
área en blanco	: 13923 Puntos
perimetro	: 1075 Puntos

»

» **vrd % Distancias del Centro de Masa al borde.**

Tabla 4.1 Valores obtenidos en la firma

72	72	72	72	72	72	72	72	74	75	76	77	78	79	80	82	83	84	85
86	87	88	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	88
87	86	85	84	83	82	81	80	78	77	76	75	74	72	72	72	72	72	72
72	72	72	72	72	72	72	72	75	76	77	78	80	81	82	83	85	86	87
88	90	90	90	90	90	90	90	90	90	90	90	90	87	86	85	84	82	81
80	79	77	76	75	74	72	72	72	72	72	72	72	72	72	72	72	74	75
77	78	79	80	81	83	84	85	86	88	88	90	90	90	90	90	90	90	90
90	90	90	88	87	86	85	84	82	81	80	79	78	77	75	74	72	72	
72	72	72	72	72	72	72	72	72	72	72	74	75	76	77	79	79	80	82
83	84	85	86	87	88	90	90	90	90	90	90	90	90	90	90	90	90	90
90	90	87	86	85	84	83	82	81	80	79	78	77	76	75	74	72	72	
72	72	72	72	72	72	72	72	72	72	72	74	75	76	77	78	79	80	81
82	83	84	85	86	87	88	90	90	90	90	90	90	90	90	90	90	90	90
90	90	90	90	88	87	86	85	84	83	82	81	80	79	78	77	76	75	

74	72	72	72	72	72	72	72	72	72	72	72	72	74	75	76	77	78
79	80	81	82	83	84	85	86	87	90	90	90	90	90	90	90	90	90
90	90	90	90	90	88	87	86	85	84	83	82	80	79	79	77	76	75
74	72	72	72	72	72	72	72	72	72	72	72	72	74

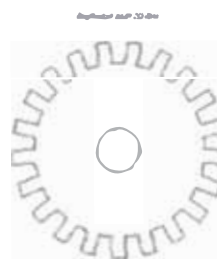
En la tabla 4.1 se muestran las distancias del centro de masa al borde del engranaje, y allí se puede observar que la distancia a la parte interna de los dientes es de 72, mientras que la distancia a la parte externa de los dientes es de 90. los otros valores que aparecen entre 72 y 90, son las distancias del centro de masa a los lados de los dientes

4.2.5 Resultados gráficos

En la figura 4.9 se observan las diferentes figuras generadas durante el proceso, las cuales son mostradas solo con fines ilustrativos; debido a que para el resultado final todas estas figuras son transparentes y tienen forma numerica matricial.



(a) Imagen a procesar.



(b) Imagen capturada.



(c) Histograma de (b).



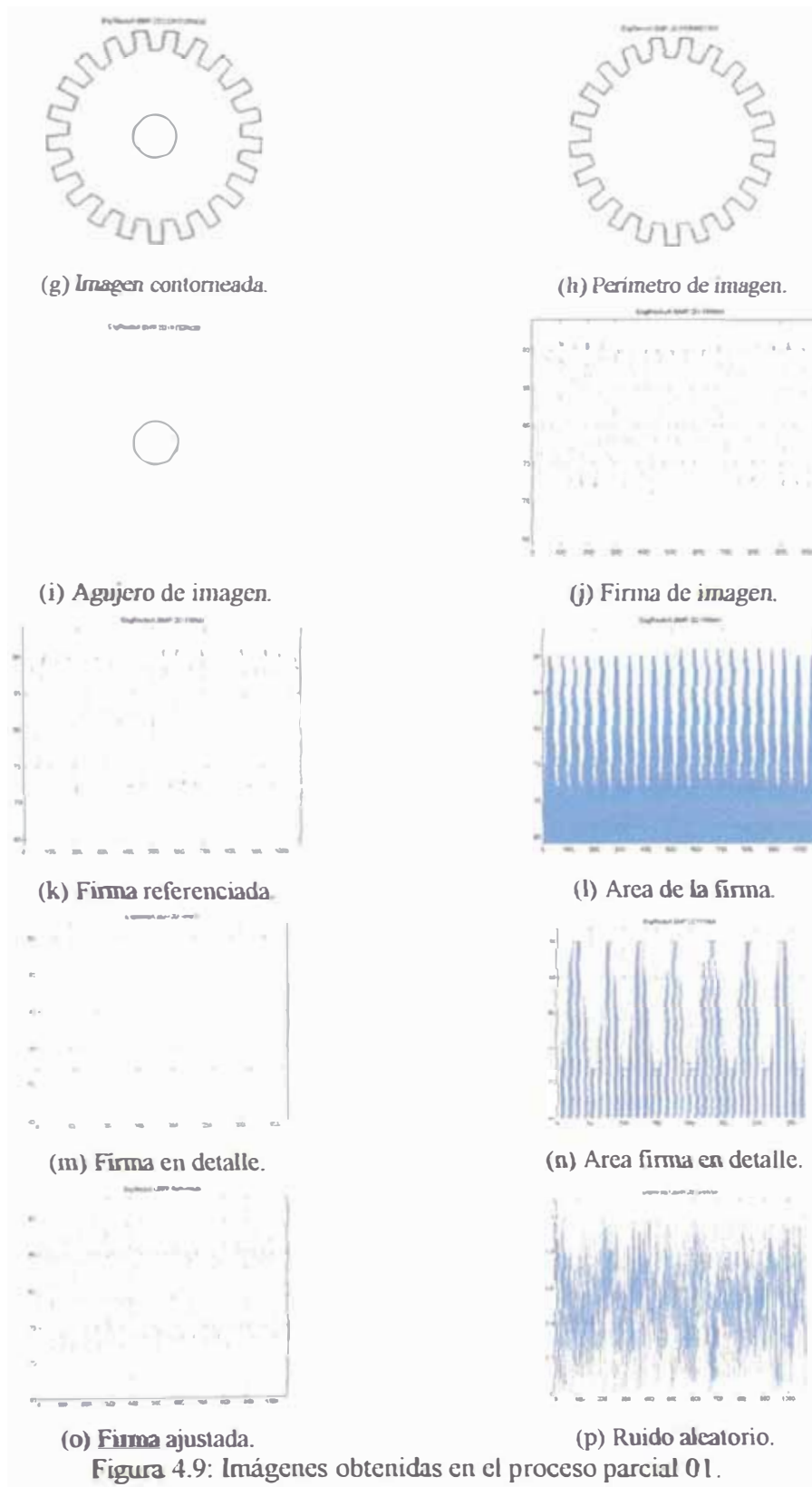
(d) Imagen monocromática.



(e) Imagen binarizada.



(f) Imagen contorneada.



4.3 Proceso parcial 02

4.3.1 Descripción del proceso

En esta etapa, a partir de los resultados del proceso anterior se realizan las tareas:

- Obtener numero de dientes
- Calcular área de la superficie
- Calcular área del agujero
- Calcular área exterior al engranaje

4.3.2 Algoritmo

La numeración del algoritmo aquí se inicia en 17 porque para el proceso global, es la continuación del anterior. Luego, como en la primera parte el algoritmo será implementado bajo la plataforma de programación del MatLab.

17. A partir del vector vrd (es el vector que almacena las distancias del centro de masa al borde), eliminar los lados de los dientes, y almacenar solo los bordes interior y exterior del engranaje.

$$vc(k) = \begin{cases} 72 \Leftrightarrow 71 \leq vrd(i) \leq 73 \\ 90 \Leftrightarrow 89 \leq vrd(i) \leq 91 \end{cases} \quad (4.18)$$

donde : $k = [1, n]$

$n =$ Numero de puntos de bordes de dientes.

Vector de bordes = $vc(i)$

18. Ubicar el inicio del borde interior de un diente y reordenar el vector vc .

$$ii = i + 1 \quad / \quad (vc(i) = 90) \cap (vc(i + 1) = 72) \quad (4.19)$$

Vector de bordes reordenado = $vcc(i)$

Donde: $vcc(1) = vc(ii)$

$vcc(2) = vc(ii + 1)$

$vcc(3) = vc(ii + 2) \dots$

19. Determinar el número de dientes, contando el número de transiciones entre borde interior a borde exterior.

$dientes = 0$

luego :

$$dientes = dientes + 1 \Leftrightarrow (vcc(j) = 72) \cap (vcc(j + 1) = 90) \quad (4.20)$$

dientes = Numero de dientes.

Graficar la imagen en niveles de gris.

Engranaje = C(i,j)
 Size(C) = MxN. Figura 4.11.a.

20. Cambiar interior del engranaje con nivel gris = 150, diferente al de la pieza(0) y al del exterior(255).

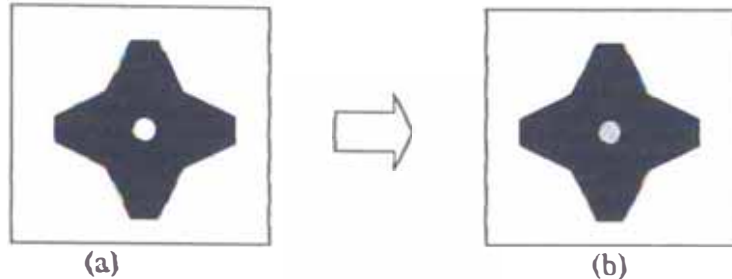


Figura 4.10: Señalización del agujero central del engranaje.
 (a) Matriz C inicial. (b) Matriz C final.

Engranaje = Matriz nueva C(i,j). Figura 4.11.b.

21. Hallar las áreas de la figura, de acuerdo al nivel de gris de las mismas.
 Calculando la constante de proporcionalidad para áreas de la imagen,

$$Kp = \pi \times R^2 \quad (4.21)$$

$$Area = Kp \times \frac{Na}{Np} \quad (4.22)$$

donde: Kp=Constante de proporcionalidad (área del agujero).
 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 cm²
 Area engranaje = Ap, en numero de puntos.
 Area exterior = A1, en numero de puntos.
 Area interior = Aa, en numero de puntos.

22. Emitir reporte final del proceso.

4.3.3 Código de Programa

En el Anexo A se muestra el código fuente que corresponde al algoritmo previamente desarrollado. PROGRAMA35: Engra03.m.

Los datos de entrada para este programa son los resultados del programa anterior.

4.3.4 Ejecución del programa

El programa se almacenó con el nombre engra03.m. Al ejecutar desde el MatLab, se visualizan el **reporte final**. También aparecen las figuras generadas en el proceso.

» engra03

Size(C)=

ans = 185 185

CAMBIANDO DE TONO EL INTERIOR...

REPORTE FINAL

Número de dientes	21
Area de la pieza	20302 Puntos
Area libre	12952 Puntos
Area agujeros	971 Puntos

»

» vcc % Bordes de los dientes del engranaje.

Tabla 4.2 Valores obtenidos en la firma, sin considerar lados de dientes.

72	72	72	72	72	72	72	72	72	72	72	72	72	90	90	90	90	90	90
90	90	90	90	90	90	90	90	72	72	72	72	72	72	72	72	72	72	72
90	90	90	90	90	90	90	90	90	90	90	90	90	72	72	72	72	72	72
72	72	72	72	72	72	72	90	90	90	90	90	90	90	90	90	90	90	90
90	72	72	72	72	72	72	72	72	72	72	72	72	72	72	90	90	90	90
90	90	90	90	90	90	90	90	72	72	72	72	72	72	72	72	72	72	72
72	90	90	90	90	90	90	90	90	90	90	90	90	90	90

En la tabla 4.2 se muestran las distancias del centro de masa al borde del engranaje, pero esta vez se han eliminado las distancias a los lados de los dientes. Con estos datos se puede determinar el número de dientes, el cual será igual al numero de transiciones entre los valores de 72 y 90.

4.3.5 Resultados gráficos

En la figura 4.11 se observan las figuras generadas durante el proceso, tienen forma numérica matricial para el programa.



(a) Imagen binarizada



(b) Imagen señalizada

Figura 4.11. Imágenes obtenidas en el proceso parcial 02.

4.4 Umbralización de la imagen

4.4.1 Proceso

Aquí se muestra el efecto de la variación del valor de umbral en el proceso de binarización. Esta vez se utiliza la fotografía que aparece en la figura 4.12.



Figura 4.12: Imagen utilizada para ver efecto de umbralización.

4.4.2 Algoritmo

A continuación se lista el algoritmo, para verificar el efecto de umbral, esta vez se utiliza una fotografía.

1. Cargar la imagen a color del engranaje recto (Azul.jpg).

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

$$\text{Max}(A) = 255$$

$$\text{Min}(A) = 0$$

$$\text{Size}(A) = M \times N \times 3. \text{ Ver figura 4.13.a.}$$

2. Llevar la imagen a niveles de gris, promediando las componentes RGB de la matriz

A.

$$C(i,j) = [A(i,j,1)+A(i,j,2)+A(i,j,3)]/3$$

$$\text{Max}(C) = 255$$

$$\text{Min}(C) = 0$$

$$\text{Size}(C) = M \times N. \text{ Ver figura 4.13.b.}$$

3. Obtener el histograma de la imagen en niveles de gris.

$$3.1 H(k) = 0 ; \text{ Para } k = [1,256].$$

$$3.2 H(v(i,j)+1) = H(v(i,j)+1) ; \text{ Para } i=[1,M], j=[1,N].$$

$$\text{Size}(H) = 256. \text{ Ver figura 4.13.c.}$$

4. Binarizar la imagen en grises. El valor del umbral(50) se obtiene a partir del histograma.

$$CC(i,j) = \begin{cases} 0 & ; \text{ Otro valor} \\ 255 & ; (C(i,j) \leq 3) \cap (C(i,j) \geq 65) \end{cases} \quad (4.23)$$

Para $i=1$ a M

Para $j=1$ a N

$$\text{Si } C(i,j) \leq 3 \longrightarrow C(i,j) = 255 \text{ (Negro)}$$

$$\text{Si } C(i,j) \geq 65 \implies C(i,j) = 255 \text{ (Negro)}$$

Nota: Los parámetros de binarización(3 y 65) se obtienen del histograma anterior.

Figura 4.14.d.

5. Se obtiene el negativo de la imagen.

Para $i=1$ a M

Para $j=1$ a N

$$NC(i,j) = 255 - C(i,j)$$

Ver figura 4.13.e.

6. Extraer histograma de imagen binarizada.

Ver figura 4.13.f.

7. Umbraliza la imagen para identificar el fondo de la foto.

Para $i=1$ a M

Para $j=1$ a N

$$\text{Si } [C(i,j) \geq 27 \& C(i,j) \leq 45]$$

$$C(i,j) = 150$$

Nota: Los parámetros de binarización (27 y 45) se obtienen del histograma anterior.

Ver figura 4.13.g.

8. Extraer histograma de imagen binarizada.

Ver figura 4.13.h.

4.4.3 Código de Programa

En el Anexo A se muestra el código fuente que corresponde al algoritmo previamente desarrollado. PROGRAMA02: UAzul.m.

4.4.4 Resultados gráficos

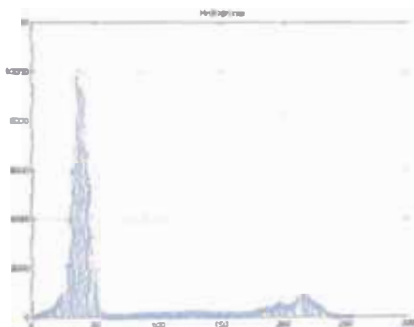
En la figura 4.13 se observan las figuras generadas durante el proceso.



(a) Imagen a color.



(b) Imagen monocromática.



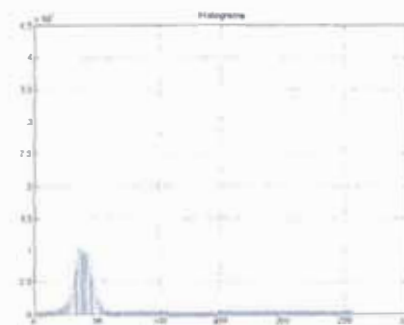
(c) Histograma de (b).



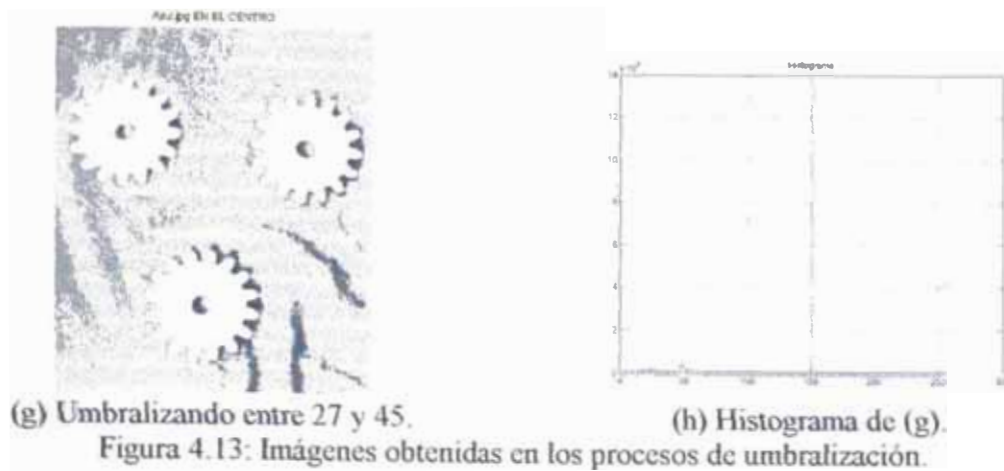
(d) Umbralizando para 50.



(e) Negativo de (d).



(f) Histograma de (e).



4.5 Aplicación de diferentes tipos de filtro

4.5.1 Proceso

Aquí se aplican diferentes tipos de filtros, para ver el efecto sobre una imagen deteriorada. Esta vez se utiliza la fotografía que aparece en la figura 4.14.

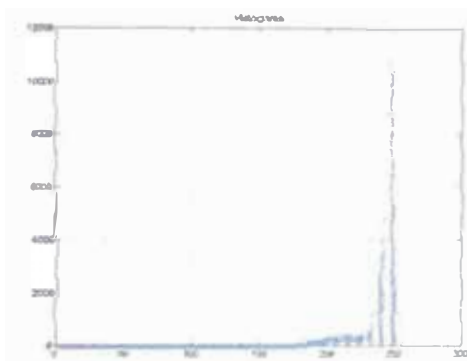
Figura 4.14: Imagen utilizada para proceso de filtrado.

Esta vez no se adjunta el código del proceso, pero esta aparece junto con otros procesos más en el anexo A.

4.5.2 Gráficos obtenidos al ejecutar programa

En la figura 4.15 se observan las figuras obtenidas al ejecutar el proceso.

(a) Engra03.bmp en 3D



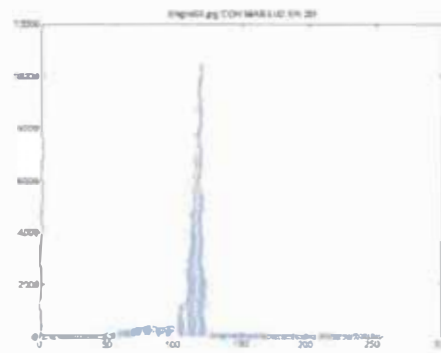
(b) Engra03.bmp en 2D



(c) Histograma de imagen b



(d) Restando intensidad 120 en 3D

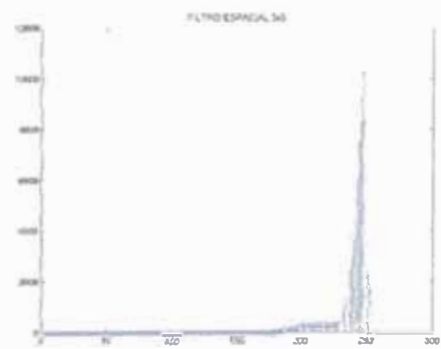


(e) Restando intensidad 130 en 2D

FILTRO ESPACIAL 3x3



(f) Histograma de imagen e

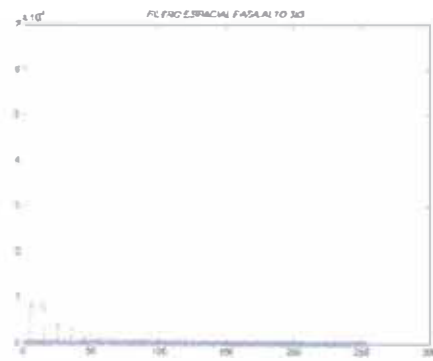


(g) Imagen con filtrado espacial 3x3

(h) Histograma de imagen (g)



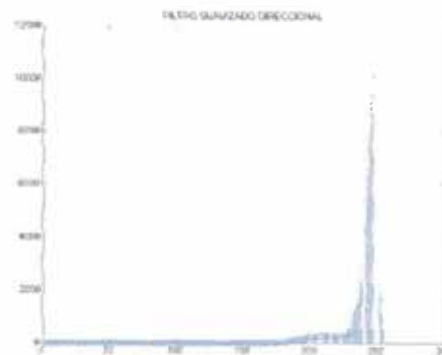
(i) Imagen con filtrado pasa alto



(j) Histograma de imagen (i)



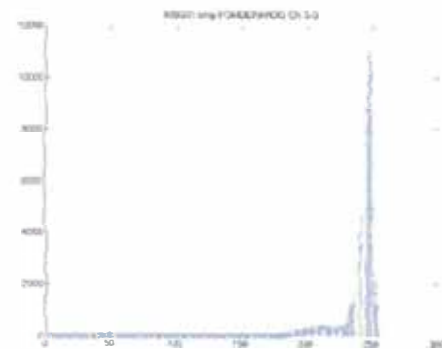
(k) Imagen con filtrado direccional



(l) Histograma de imagen (k)



(m) Imagen con filtrado de mediana



(n) Histograma de imagen (m)

Figura 4.15: Imágenes obtenidas al aplicar diferentes tipos de filtros.

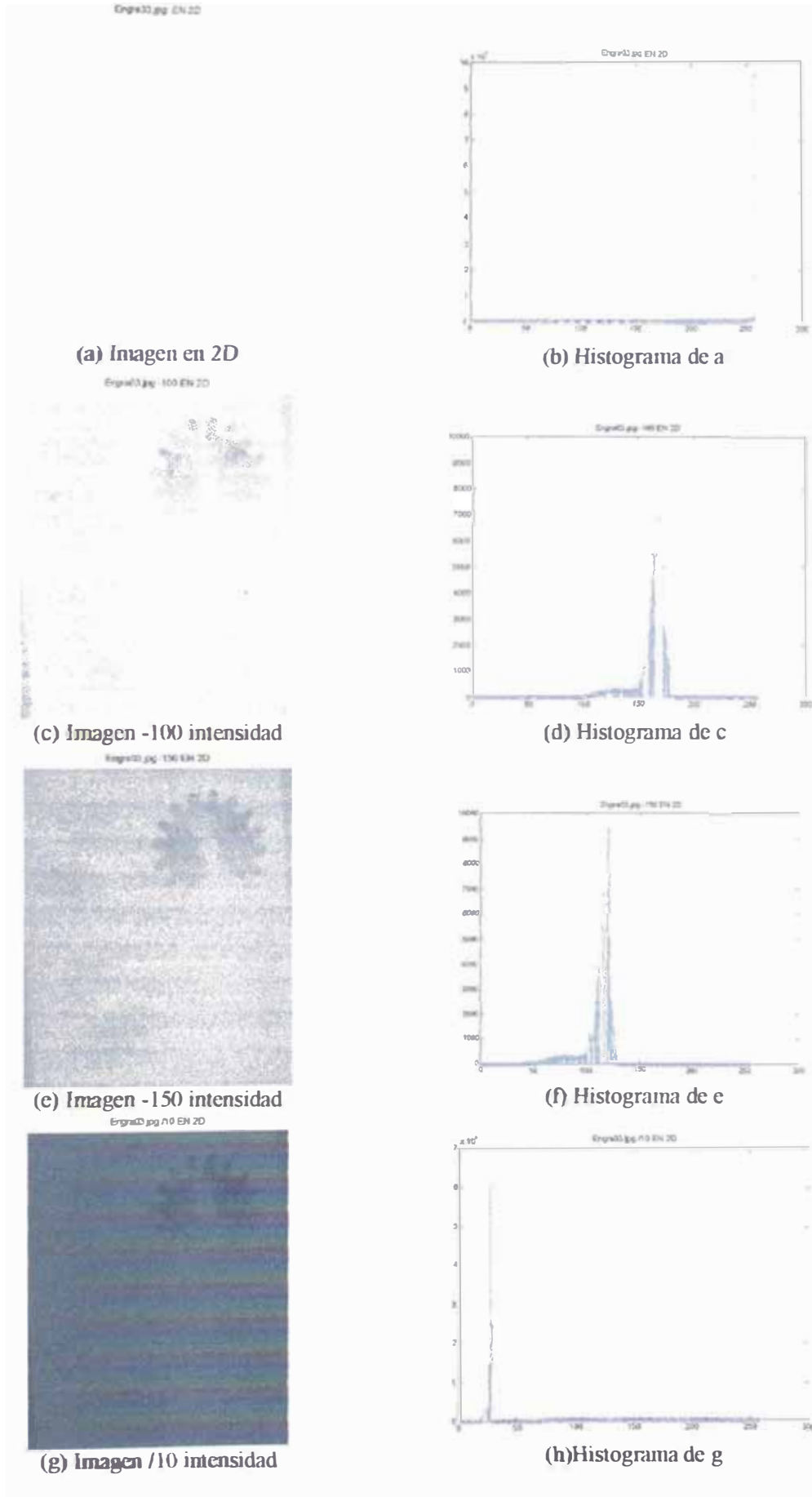
4.6 Manipulación de la intensidad de luz

4.6.1 Proceso

Aquí se manipula la intensidad de luz de la imagen, para ver el efecto sobre una imagen deteriorada. Se utiliza la misma fotografía del proceso anterior (ver figura 4.14).

4.6.2 Resultados gráficos

En la figura 4.16 se observan las figuras obtenidas al ejecutar el proceso.



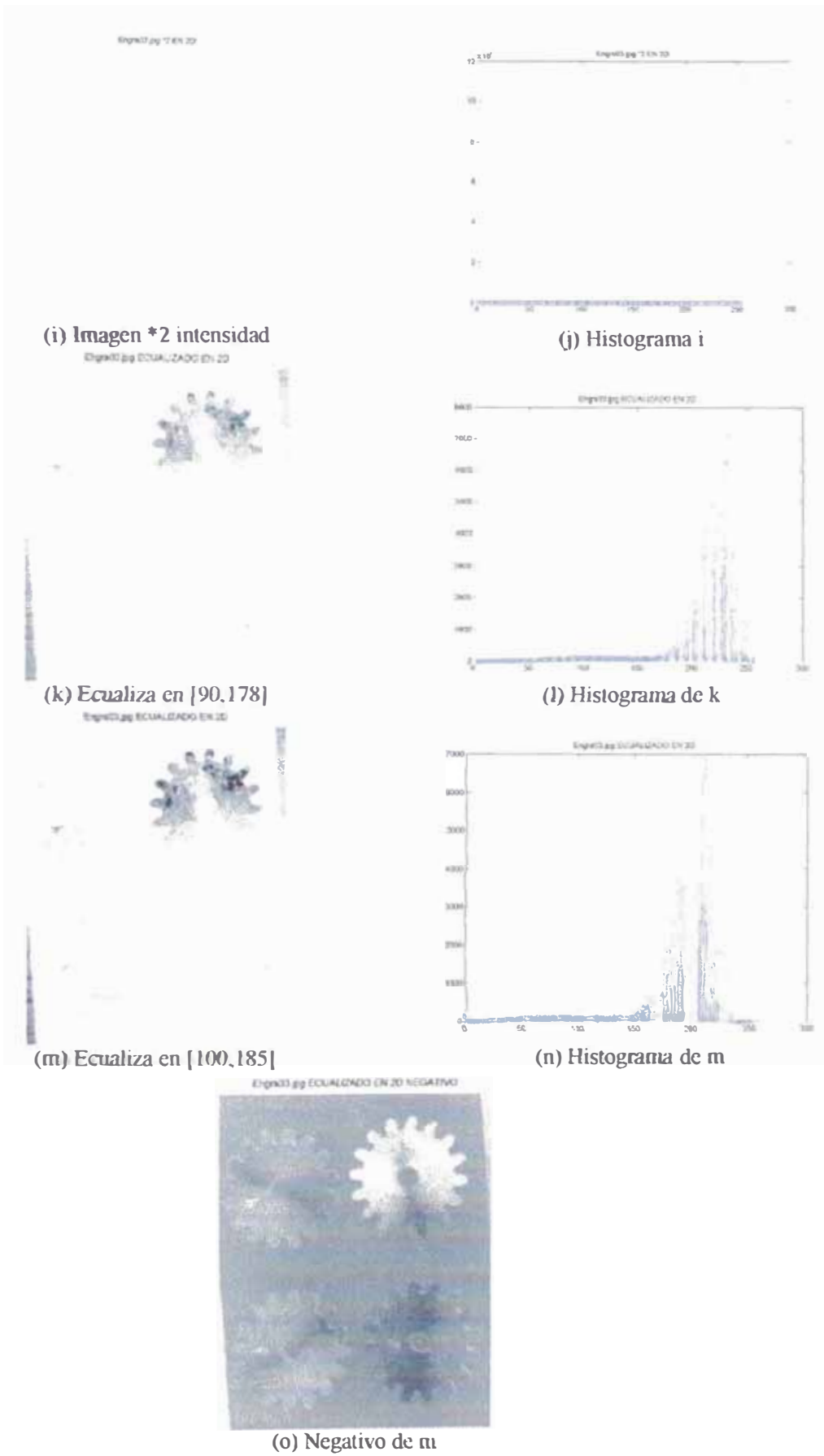


Figura 4.16: Imágenes obtenidas al modificar la intensidad.

CAPITULO 5

Descripción del Sistema Propuesto

En este capítulo se muestra la implementación de los algoritmos del sistema de control de calidad y su desarrollo en plataforma de programación Visual C++.

En la figura 5.1 se muestra el diagrama de flujo del sistema de control de calidad.



Figura 5.1: Diagrama de flujo del sistema de control de calidad en simulación.

El capítulo abarca los siguientes puntos:

- Descripción del sistema.
- Parámetros del engranaje patrón.
- Desarrollo del software.
- Ejecución del sistema.

5.1 Descripción del sistema

En la figura 5.2 se muestra el diagrama de bloques del sistema, allí se puede apreciar los diferentes módulos con los que cuenta el sistema.

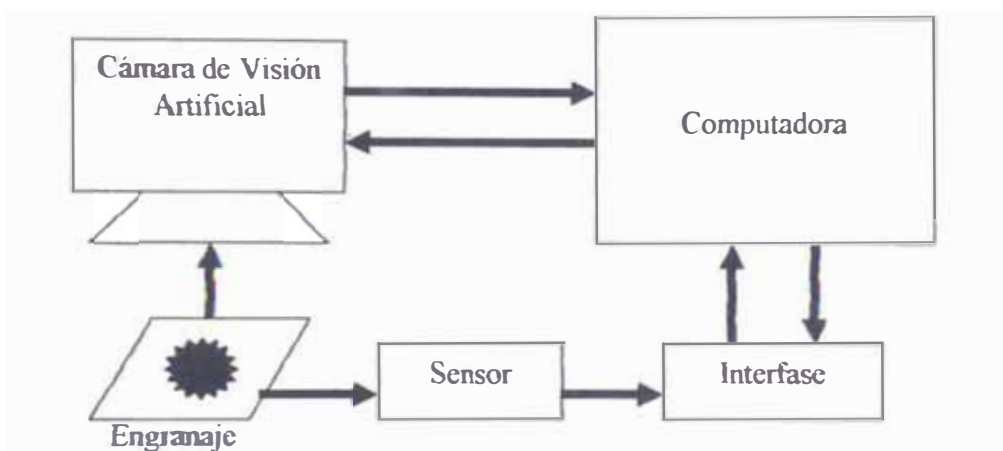


Figura 5.2: Diagrama de bloques del sistema.

A continuación en la figura 5.3 se muestra el diagrama de bloques del sistema, mostrando el detalle de la cámara de visión artificial, donde uno de los componentes fundamentales del mismo es el sistema de iluminación.

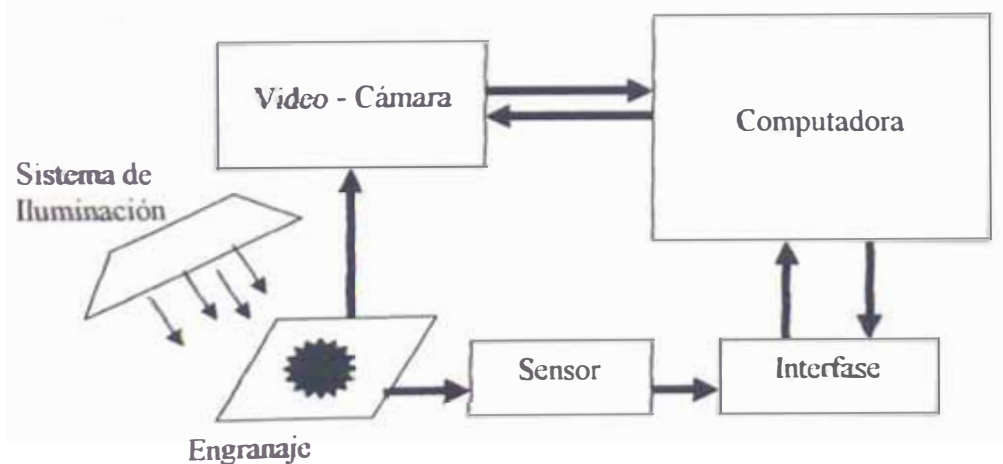


Figura 5.3: Diagrama de bloques del sistema, mostrando la cámara de visión artificial.

Los elementos del sistema que aparecen en el diagrama de la figura 5.3, realizan funciones específicas dentro del sistema

Video-cámara.- Es el sensor principal del sistema. Es el elemento encargado del sensado ó captura de la imagen que posteriormente será procesada. La calidad de la cámara, junto con el programa monitor (proporcionado por el fabricante), junto con el sistema de iluminación determinaran la calidad de la imagen inicial en el proceso, y esto será determinante para la obtención de los resultados finales.

Sistema de iluminación.- Es el encargado de proporcionar la cantidad y calidad de luz adecuada para obtener una imagen de calidad, la cual simplificara o complicara el procesamiento posterior de la imagen.

Computadora (Hardware).- Es el cerebro del procesado, y será el elemento determinante en lo que respecta al tiempo de respuesta del sistema, junto con la calidad de los algoritmos y técnicas de procesamiento utilizadas.

Programas (Software).- Son el alma del sistema. Es la parte lógica, la cual definirá la forma ó estrategia de proceso de la imagen. Junto con el hardware determinaran el tiempo de respuesta del sistema. El software tendrá como soporte los algoritmos de proceso establecidos.

Sensor.- Es el encargado de detectar la presencia del engranaje para comunicar a la computadora, vía interfase su presencia y esta a su vez enviar la orden de captura de imagen.

Interfase.- Es el circuito electrónico encargado de procesar la señal detectada por el sensor y comunicarse con la computadora. Este circuito se muestra en la figura 5.4.

Engranaje.- Es el objeto de estudio ó análisis dentro del sistema.

Circuito de Interfase:

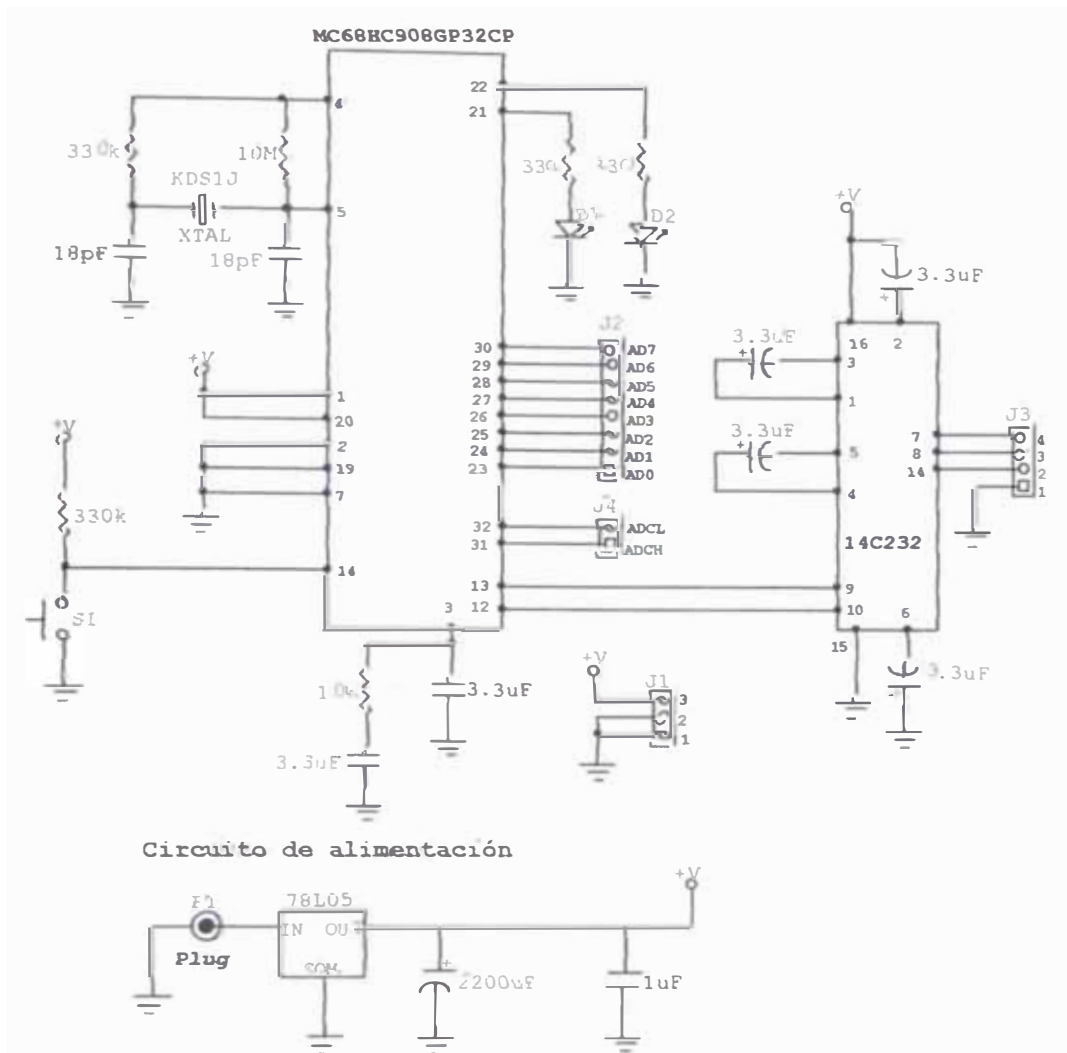


Figura 5.4: Circuito de interfase entre la cámara de adquisición y la computadora.

El circuito de interfase entre la cámara de adquisición de imagen y la computadora se encarga de detectar la presencia del engranaje, para esto se puede utilizar un interruptor-sensor (S1 en el circuito) ó cualquier otro tipo de sensor. El micro-controlador MC68HC908GP32CP está en comunicación permanente con la computadora y en el instante en que es notificado de la presencia del engranaje, envía un pulso mediante el C.I. 14C232 para que el programa realice la acción correspondiente. En este caso el programa transfiere el control de la imagen a la aplicación para su procesamiento.

El C.I. 14C232 acondiciona la señal digital del micro-controlador para enviarla a través del jumper J3 al puerto serial (COM 1) de la computadora. El jumper J2 es para conexión de visualización LCD, en forma opcional. El jumper J1 es para obtener fuente +Vcc, y el jumper J4 no se usa.

En la figura 5.5 se muestran algunas imágenes obtenidas por diferentes cámaras y bajo diferentes condiciones de luz, allí se puede apreciar claramente como varía la calidad de la información obtenida en la imagen.

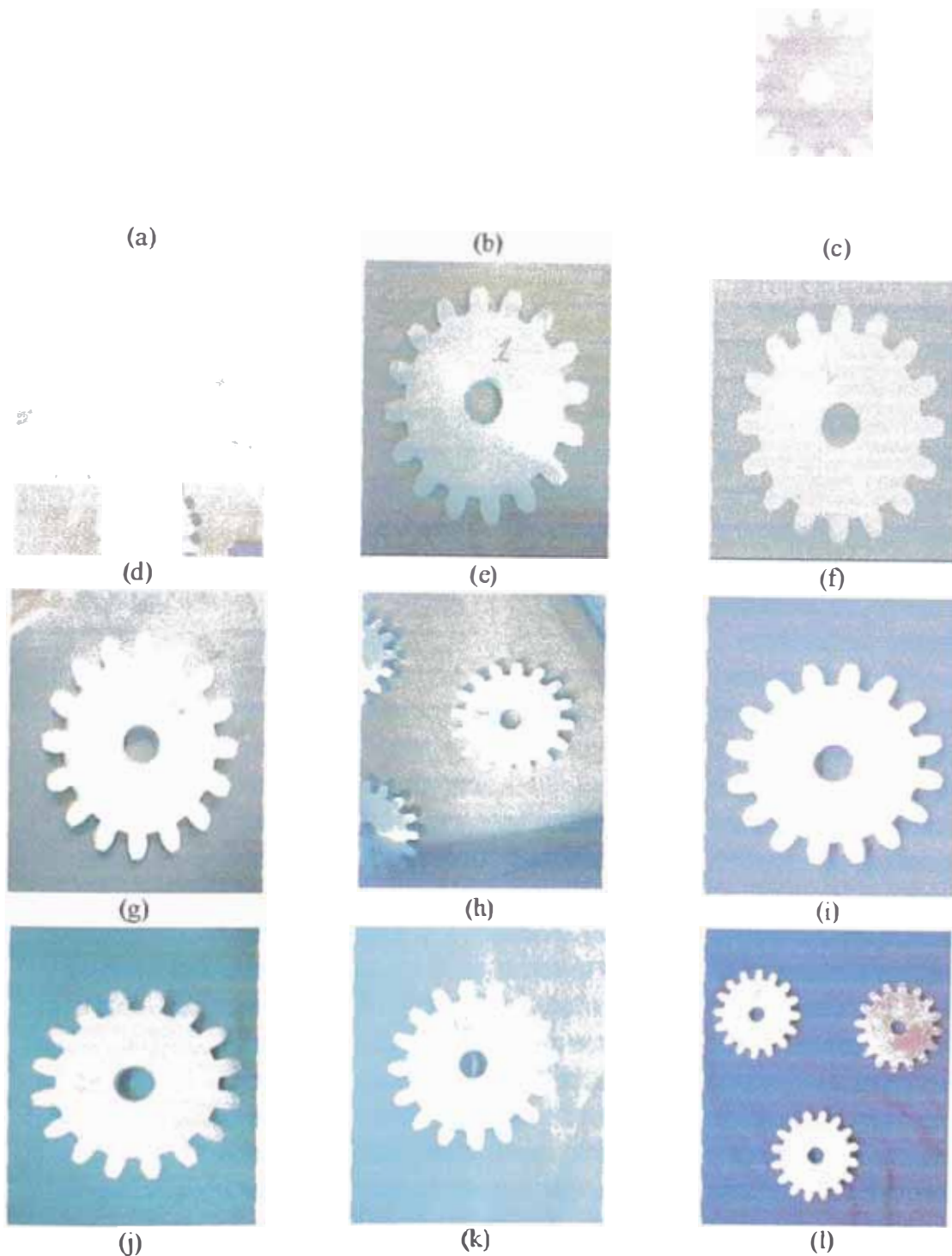


Figura 5.5: Imágenes obtenidas bajo diferentes condiciones.

(a)(b)(c)(d) Con cámara Kodak F350.

(e)(f)(g)(h) Con webcam Creative video Blaster Plus.

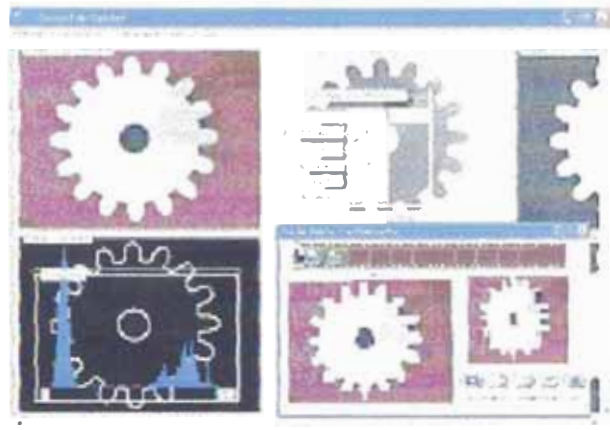
(i)(j)(k)(l) Con cámara digital Creative PC CAM 900.

En la figura 5.6 se muestran las fotografías los componentes del sistema de control de calidad (figura 5.6.a), así como de cada uno de los elementos que la conforman.

Figura 5.6.b, la interfase de software del usuario y en la figura 5.6.c la computadora. En la figura 5.6.d se muestra la cámara de visión, y el detalle de esta en las figuras 5.6.e y figura 5.6.f. Finalmente en la figura 5.6.g la cámara webcam y en figura 5.6.h la interfase de hardware entre la cámara de visión y la computadora.



(a) Sistema de control de calidad.



(b) Interfase de software del usuario.



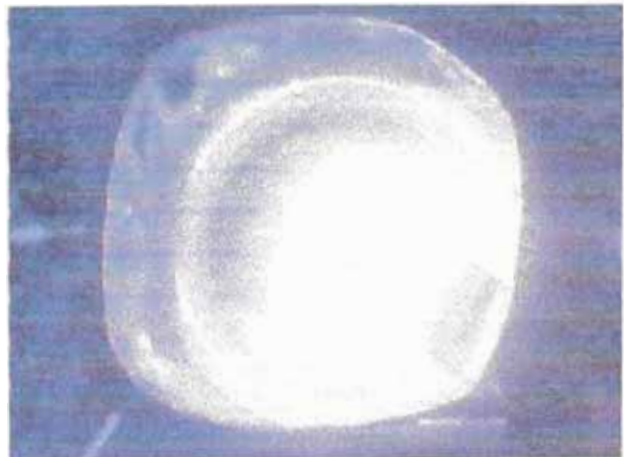
(c) Computadora.



(d) cámara de visión.



(e) cámara de visión.



(f) cámara de visión.



(g) WebCAM.



(h) Interfase de hardware.

Figura 5.6: fotografías de sistema de control de calidad y sus componentes.

A continuación se describen las características de los elementos del sistema:

Computadora

Sistema:

Microsoft Windows XP – Profesional
Versión 2002

Equipo:

Intel® Pentium® 4 CPU 1.70GHz
256 MB – RAM
40 GB – HD

Monitor:

Samsung SyncMaster 500b

Tarjeta de video:

Riva TNT Model 64 – 32 MB de Memoria

WebCam

Video BLASTER – Creative
WebCam Plus

cámara de visión

Caja forrada interiormente con papel aluminio para mejor iluminación
Fluorescente de 22 watts a 220 voltios

Sensor

Optico ON-OFF para detectar presencia de engranaje en la cámara de visión.

Interfase

Circuito electrónico basado en uControlador Motorola de 40 pines.

Engranaje

Objeto de análisis del sistema

5.2 Parámetros del engranaje patrón

5.2.1 Dimensiones físicas del engranaje

En la figura 5.7 se muestran los diámetros del engranaje patrón, los mismos que serán utilizados para calcular las áreas del engranaje.

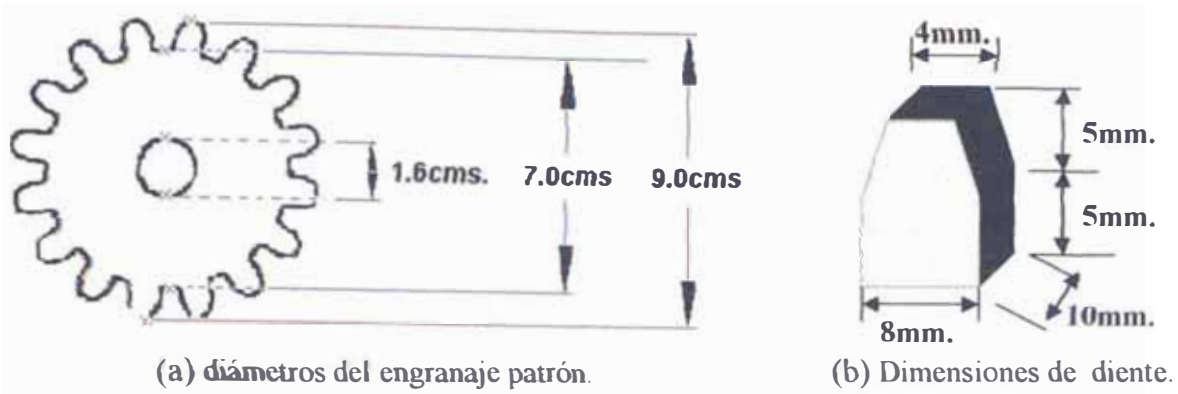


Figura 5.7: Dimensiones del engranaje patrón.

Hallando el área del engranaje patrón a partir de sus dimensiones:

$$Ad = Base \times h - 2 \times A\Delta = 0.8 \times 1 - 2 \left(\frac{0.5 \times 0.2}{2} \right) = 0.7cms.^2 \quad (5.1)$$

$$Ac = \pi(Rm - Rc)^2 = \pi(3.5 - 0.8)^2 = 22.90cms.^2 \quad (5.2)$$

$$Aa = \pi \times Rc^2 = 3.14159 \times 0.8^2 = 2.01cms.^2 \quad (5.3)$$

$$Ae = Ac + (Nd \times Ad) = 22.09 + 16 \times 0.7 = 33.29cms.^2 \quad (5.4)$$

- Donde:
- Ae = área del engranaje patrón
 - Aa = área del agujero
 - Ad = área de cada diente
 - Ac = area del cuerpo
 - Nd = Numero de dientes del engranaje

5.2.2 Coordenadas referenciales del engranaje

En la figura 5.8 se muestran las coordenadas relativas a puntos singulares del engranaje patrón. Dichas coordenadas permiten determinar la relación entre pixeles y unidades de longitud y área.

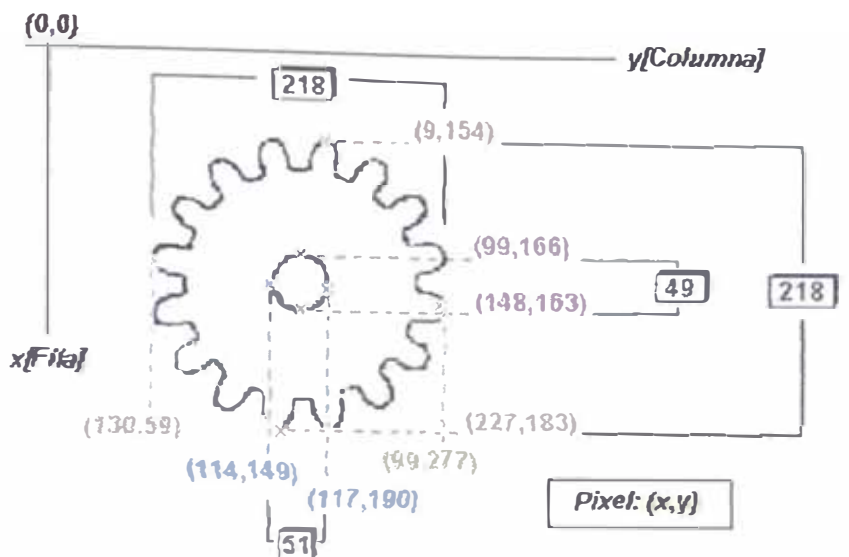


Figura 5.8: Coordenadas referenciales relativas de puntos singulares.

A partir de las dimensiones del engranaje patrón y el cálculo de áreas podemos resumir los siguientes parámetros para el engranaje patrón:

Diámetro Mayor	= 9 cms.
Diámetro Menor	= 7 cms.
Diámetro de agujero	= 1.6 cms.
área del engranaje	= 33.29 cms. ²
área del agujero	= 2.01 cms. ²
Numero de dientes	= 16

Relacionando los parámetros medidos y los obtenidos del sistema (ver figuras 5.7, 5.8), podemos determinar los parámetros característicos del engranaje patrón, como se muestra en la tabla 5.1.

Tabla 5.1 Dimensiones del engranaje patrón en cms. Y píxeles.

	Por medición	Del sistema
Diámetro mayor	9 cms.	218 píxeles
Diámetro menor	7 cms.	180 píxeles
Diámetro de agujero	1.6 cms.	50 píxeles
área engranaje	33.29 cms. ²	30000 píxeles
área agujero	2.01 cms. ²	1380 píxeles
Centro de masa		(120,170)*

(*) El centro de masa varía de acuerdo a la ubicación de la pieza en la cámara de V.A.

Para la obtención del diámetro menor en píxeles se realizó un barrido de puntos dentro de un ángulo $\alpha = [0^\circ - 20^\circ]$, usando la variable r , para detectar el cambio de gris y detectando el borde del engranaje. Se obtuvieron los siguientes resultados:

Tabla 5.2 Datos obtenidos de bordes exteriores del engranaje.

A	0	5	10	15	20
R	218	180	178	180	218
condición	borde	borde	borde	borde	borde

A partir de las medidas del engranaje patrón podemos llegar a la relación:

$$kp = \frac{33.29cms.^2}{30000pixeles} = 1.109666 \times 10^{-3} \quad \left[\frac{cms.^2}{pixeles} \right] \quad (5.5)$$

de donde para calcular el área en $cms.^2$ a partir de píxeles:

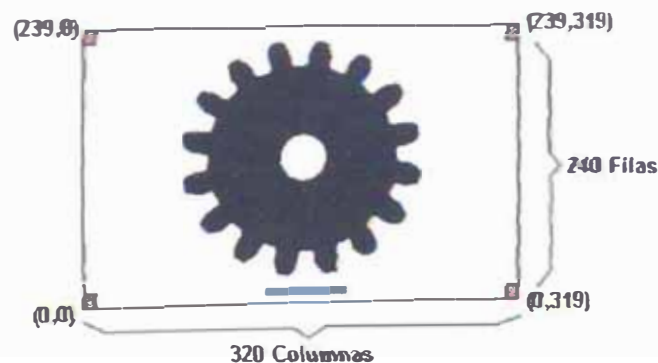
$$Aec = kp \times Aep \quad [cms.^2] \quad (5.6)$$

donde: $Aec = \text{área de engranaje en cms.}^2$

$Aep = \text{área de engranaje en píxeles}$

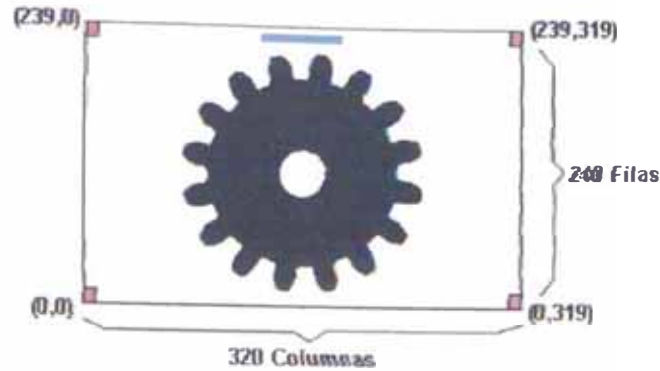
En la figura 5.9.a se muestra la referencia de coordenadas en memoria para la imagen capturada. Se debe tener en cuenta que posteriormente, la imagen llevada a una matriz de ceros y unos para su análisis posterior quedara reflejada respecto al eje x, como se muestra en la figura 5.9.b.

Referencia de coordenadas para imagen en resolución 240x320:



(a) Referencia de coordenadas de imagen.

Referencia de coordenadas en matriz de análisis:



(b) Referencia de coordenadas de matriz.

Figura 5.9: Cambio de referencia de coordenadas para el sistema.

Para determinar el número de dientes del engranaje, se realiza un barrido de 360 grados, con una variación de ángulo de 5° , manteniendo el valor de r fijo:

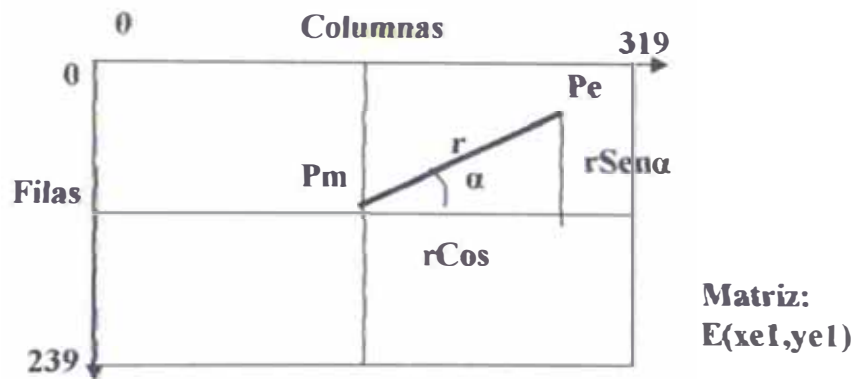


Figura 5.10: Barrido circular para determinar el número de dientes del engranaje.

Donde:

$E(xe1, ye1)$ y $E(xe2, ye2)$: Matriz que contiene el engranaje

$Pm = (xm, ym) = (116, 175)$: Centro de masa del engranaje

$Pe = (xe, ye)$: Punto de evaluación del engranaje.

$r = 102$: Distancia al punto medio de dientes.

$d\alpha = 5$: Incremento de α .

$xr1 = r \text{ Sen}(\alpha)$: Primer punto de evaluación

$yr1 = r \text{ Cos}(\alpha)$

$xe1 = xm - xr1$

$ye1 = ym + yr1$

$xr2 = r \text{ Sen}(\alpha + d\alpha)$: Segundo punto de evaluación

$$y_{r2} = r \cos(\alpha + \Delta\alpha)$$

$$x_{e2} = x_m - x_{r2}$$

$$y_{e2} = y_m + y_{r2}$$

Luego,

Si $E(x_{e1}, y_{e1})$ y $E(x_{e2}, y_{e2})$ son diferentes se encontró un diente

Si $E(x_{e1}, y_{e1})$ y $E(x_{e2}, y_{e2})$ son iguales hay un diente

En la tabla 5.3 se muestran los valores obtenidos por el programa para detectar el número de dientes, utilizando el engranaje patrón:

Tabla 5.3 Ubicando dientes en el primer cuadrante del engranaje.

A [°]	Pe1 [(xe1, ye1)]	Pe2 [(xe1, ye1)]	Pe1 ¿? Pe2	Dientes
0	116, 278	107, 277	diferentes	1
5	107, 277	98, 276	iguales	
10	98, 276	89, 274	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	
55	32, 234	27, 227	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	

La verificación se realizó solo en el primer cuadrante, debido a la simetría del engranaje patrón. Se observa que el número de dientes que aparece es el doble debido a que por cada verificación de condición se produce el evento 2 veces.

5.3 Desarrollo del software

5.3.1 Estructura de la aplicación

La figura 5.11 muestra el árbol de menús generada en la aplicación, donde cada opción realiza una tarea específica.

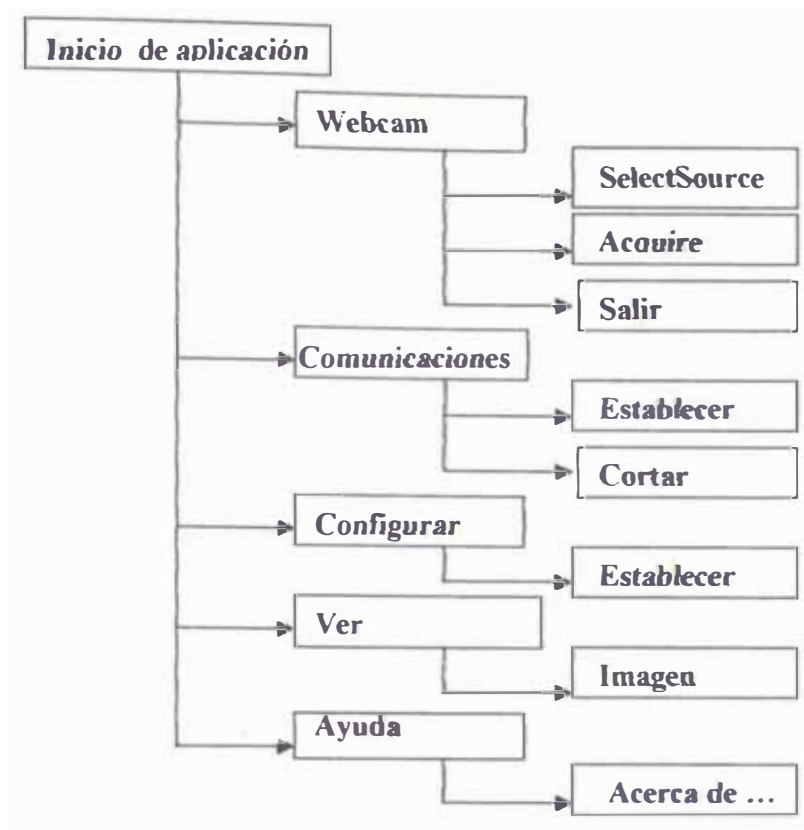


Figura 5.11: Estructura de menús de la aplicación.

Las acciones realizadas en cada opción son las descritas a continuación:

- **WebCam - SelectSource.**- Carga la ventana de selección de interfase de adquisición de imágenes.
- **WebCam - Acquire.**- Carga la interfase de adquisición de imágenes seleccionada previamente.
- **WebCam - Salir.**- Termina la ejecución del programa.
- **Comunicaciones - Establecer.**- Abre el puerto de comunicación serial para comunicación con la tarjeta de interfase.
- **Comunicaciones - Cortar.**- Corta la comunicación vía puerto serial.
- **Configurar - Establecer.**- Permite seleccionar los parámetros de: puerto de comunicación, resolución y el umbral para la binarización.
- **Ver - Imagen.**- Transfiere control de la imagen a la aplicación, realiza los procesos de control y displaya imágenes en pantalla.
- **Ayuda - Acerca de.**- Muestra información sobre versión y año del programa.

5.3.2 Estructura de control de la interfase de adquisición de imágenes

Una vez elegida la opción de adquirir imagen, se carga la interfase de adquisición

de imágenes correspondiente a la webcam que se está utilizando. En este caso dicha interfase cuenta con 5 opciones, las cuales se muestran en la figura 5.12.

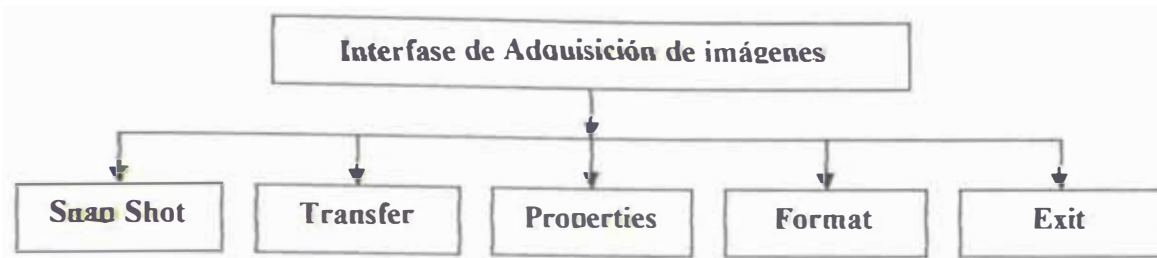


Figura 5.12: Estructura de control de interfase de adquisición de imagen.

Donde:

- Snap Shot (Instantánea)** = Mediante la clase CTwain captura imagen. El proceso es transparente para el programador.
- Transfer (Transferir)** = Transfiere imagen de aplicación a pantalla.
- Properties (Propiedades)** = Ajusta propiedades de webcam.
- Format (Formato)** = Selecciona formato de imagen
- Exit (Salida)** = Devuelve control del sistema a aplicación.

5.3.3 Diagrama de flujo del sistema de control de calidad

En la figura 5.14 se muestra el diagrama de flujo del sistema. El inicio del proceso es cuando la cámara de visión artificial está iluminada, se ha ejecutado el programa de control de calidad y se ha activado la opción [WebCam – Acquire ...], previa selección de la interfase de adquisición de imagen adecuada (cámara y programa monitor).

Aquí el sensor chequea si hay presencia de engranaje o no en la cámara de visión artificial, de no ser así el sistema sigue esperando, caso contrario; se realiza la captura de la imagen a memoria, luego transfiere dicha imagen a la aplicación como una matriz de 3D, configurada como se aprecia en la figura 5.13.

[R;C;B]0,0					[R;C;B]0,319
[R;C;B]239,0					[R;C;B]239,319

Figura 5.13: Representación de la imagen para el programa.

Después acondiciona la imagen mejorando sus características (monocromatizado,

filtrado, umbralizado, etc.) para poder realzar la información que permita obtener los datos específicos deseados sobre el engranaje.

Luego la información obtenida es procesada tomando como referencia los parámetros esperados (datos). Si los resultados son validos entonces el engranaje pasó el control de calidad, caso contrario será considerado como un engranaje fallido.

El programa finalmente emitirá un reporte sobre las características del engranaje evaluado.

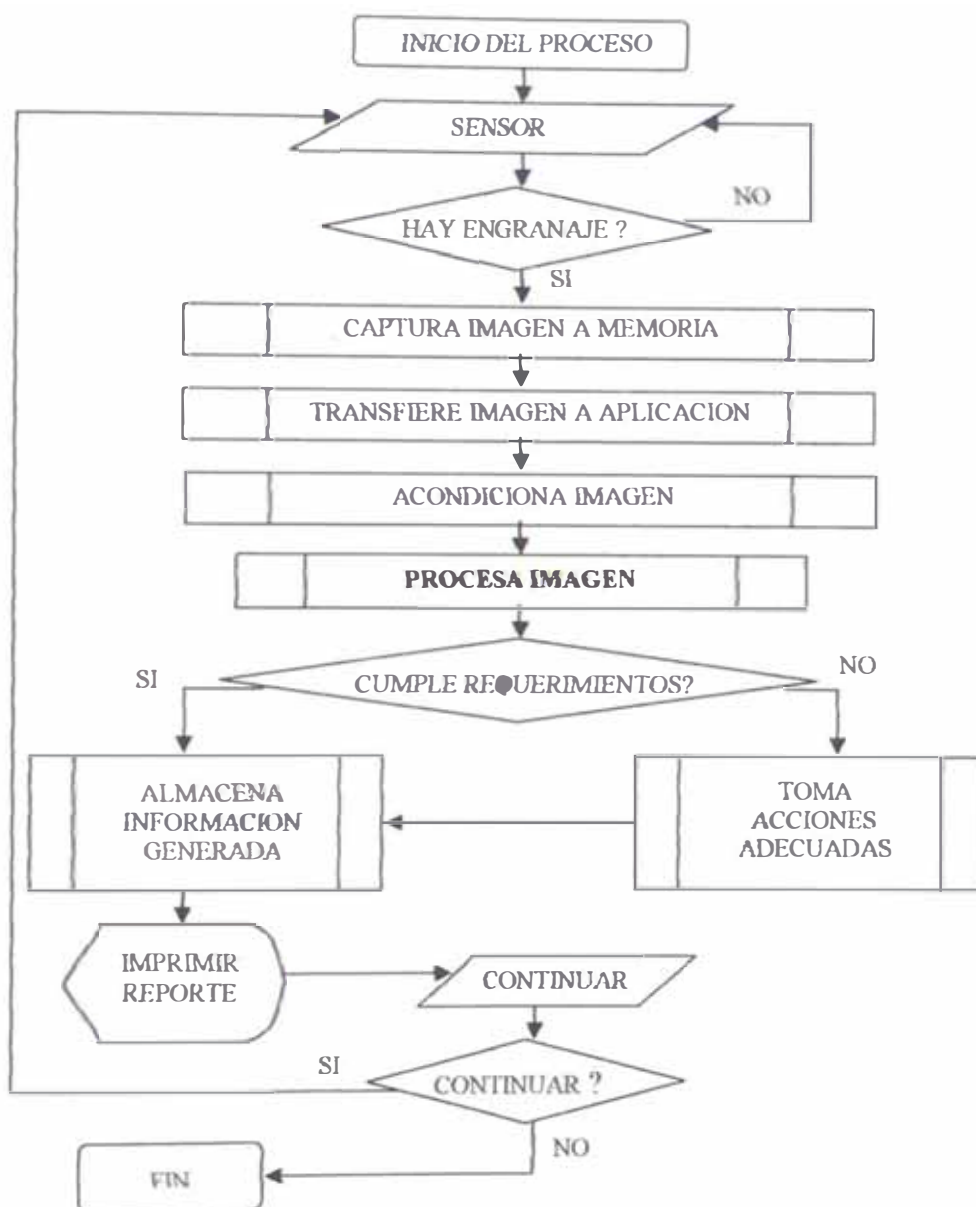


Figura 5.14: Diagrama de flujo del sistema.

5.3.4 Ejecución de la aplicación

En la figura 5.15 se muestran las funciones que son ejecutadas inicialmente por la aplicación. El programa antes de cargar la interfase del usuario y darle a este la posibilidad

de **control** del programa, previamente ejecuta las funciones que se encargan de **inicializar** el **entorno** de la aplicación. A continuación de muestran los prototipos de dichas funciones:

```

CTwain::CTwain(HWND hWnd)
BOOL CTwain::InitTwain(HWND hWnd)
CMainFrame::CMainFrame()
BOOL CMainFrame::PreCreateWindow(CREATESTRUCT& cs)
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
void CMainFrame::ActivateFrame(int nCmdShow)
▶ ▶ ▶ Luego queda cargada la interfase de la aplicación.
    
```

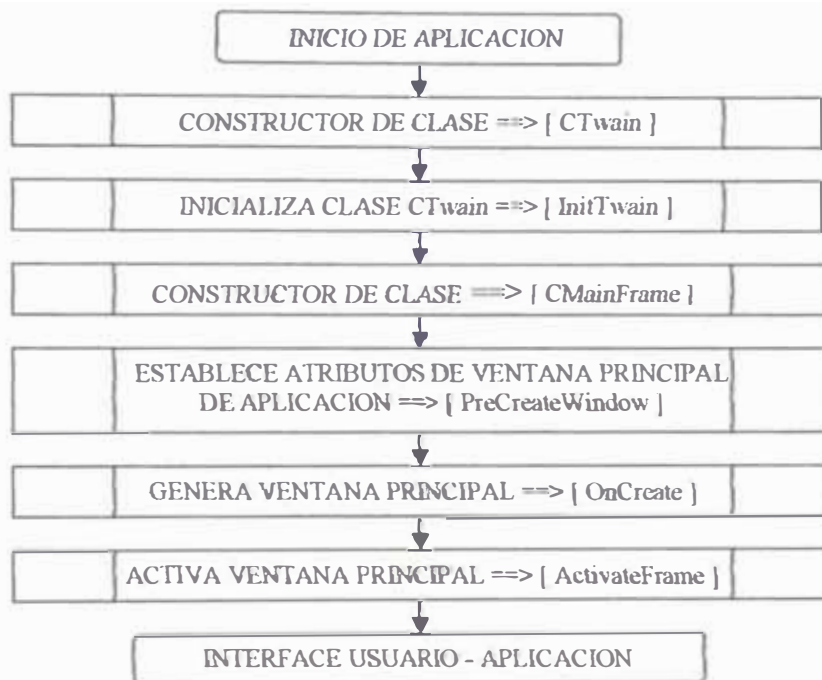


Figura 5.15: Diagrama de flujo al ejecutar el sistema.

El código fuente de estas funciones, se adjuntan en el apéndice A.

5.3.5 Ejecución del evento WebCam - SelectSource... en aplicación

Estando en la ventana principal y ejecutando el evento **WebCam - SelectSource...** , la aplicación ejecuta la secuencia de funciones mostrada en la figura 5.16.

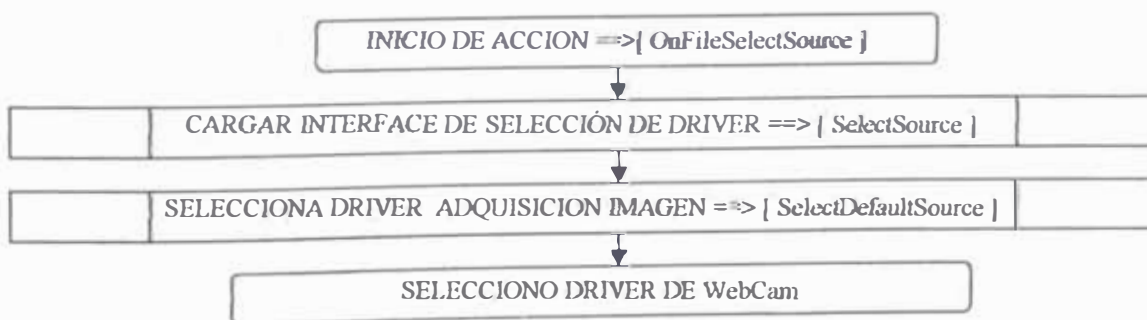


Figura 5.16: Diagrama de flujo al ejecutar WebCam - SelectSource... del sistema.

Esta opción permite al usuario seleccionar el driver y la interfase de adquisición de imágenes (programa monitor) proporcionado por el fabricante de la cámara con la que se desea trabajar. Esta interfase también condicionara el modo de operación del sistema, debido existen varias interfases para una misma cámara. Además de la cámara y la interfase, afectara también el comportamiento el tipo de computadora y el sistema operativo.

A continuación de muestran los prototipos de dichas funciones:

```
void CMainFrame::OnFileSelectsource()
BOOL CTwain::SelectSource()
BOOL CTwain::SelectDefaultSource()
▶ ▶ ▶ interfase de webcam queda seleccionada. Vuelve a interfase de
aplicación.
```

El código fuente de estas funciones, se adjuntan en el apéndice A.

5.3.6 Ejecución del evento WebCam - Acquire... en aplicación

Estando en la ventana principal de aplicación y ejecutando el evento **WebCam Acquire...**, la aplicación realiza la secuencia de funciones mostrada en la figura 5.17.

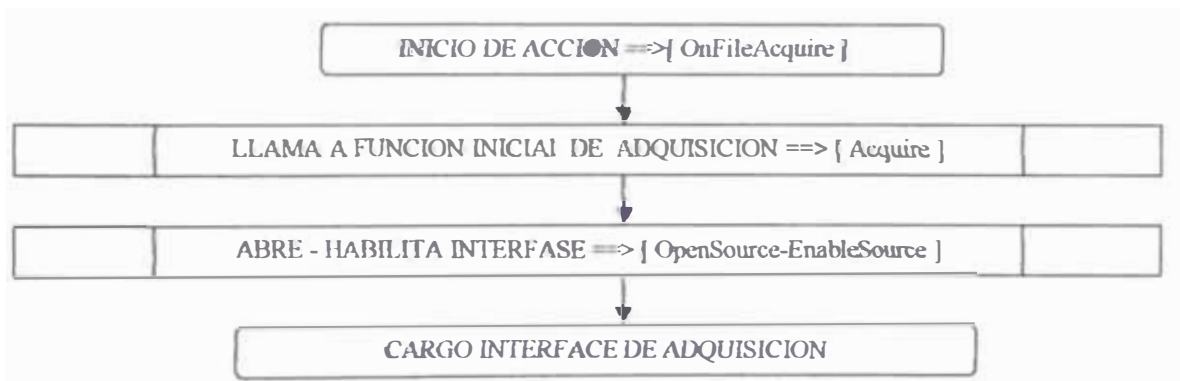


Figura 5.17: Diagrama de flujo al ejecutar WebCam - Acquire... del sistema.

Al ejecutar esta opción, en principio se ejecuta la función **OnFileAcquire**, la cual a su vez invoca a la función **Acquire**, que se encarga de cargar la interfase de adquisición de imágenes. Luego la cámara empieza a trabajar como video y está lista para la captura.

A continuación de muestran los prototipos de dichas funciones:

```
void CMainFrame::OnFileAcquire()
BOOL CTwain::Acquire(int numImages)
BOOL CTwain::OpenSource(TW_IDENTITY *pSource)
BOOL CTwain::EnableSource(BOOL showUI)
▶ ▶ ▶ Luego carga la interfase de adquisición de imagen.
```

El código fuente de estas funciones, se adjuntan en el apéndice A.

5.3.7 Ejecución del evento Transfer... en interfase de adquisición de imágenes

Estando en la interfase de adquisición de imágenes y ejecutando el evento **Transfer...**, se produce la secuencia de funciones mostrada en la figura 5.18.

Al ejecutar esta opción, la interfase envía un mensaje generado por el evento click. Luego dicho mensaje es traducido por la función **TranslateMessage**, la cual indica a la función **GetImageInfo** que obtenga información sobre la imagen y esta sea capturada. Enseguida se ejecutan los diferentes procesos con la imagen capturada para determinar los parámetros necesarios para el control de calidad.

Finalmente se emite reporte y displaya las diferentes imágenes obtenidas en el proceso.

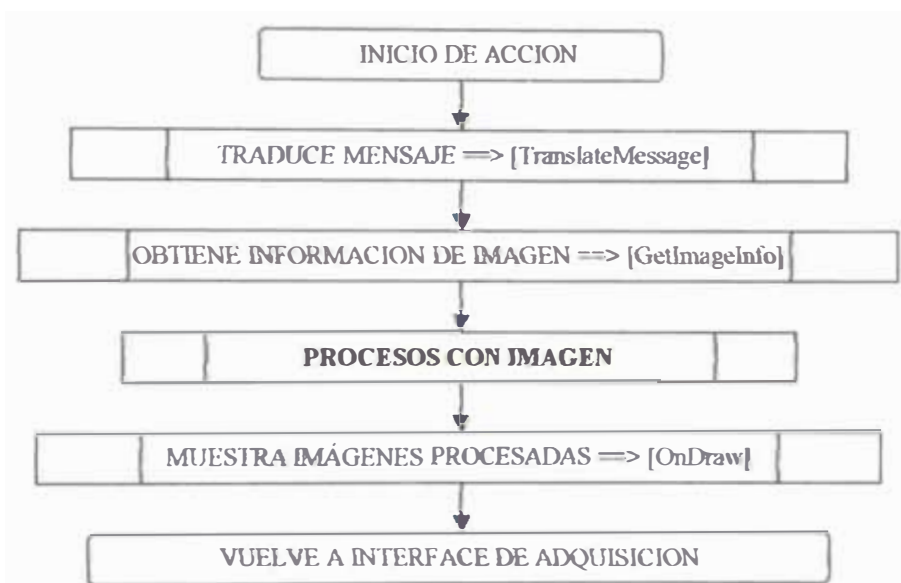


Figura 5.18: Ejecutando **Transfer ...**, en interfase de adquisición de imágenes.

A continuación se muestran los prototipos de dichas funciones:

```
void CTwain::TranslateMessage(TW_EVENT& twEvent)
void CTwain::TransferImage()
BOOL CTwain::GetImageInfo(TW_IMAGEINFO& info)
BOOL CTwain::GetImage(TW_IMAGEINFO& info)
▶ ▶ ▶ Aquí realiza procesos con imagen adquirida.....
BOOL CTwain::EndTransfer()
```

El código fuente de estas funciones, se adjuntan en el apéndice A.

5.3.8 Ejecución del evento Snap Shot... en interfase de adquisición de imágenes

Adquiere imagen a memoria. Depende de la interfase de la webcam.

Es transparente para el usuario.

5.3.9 Ejecución del evento Exit ... en interfase de adquisición de imágenes

Vuelve a la interfase de la aplicación.

5.3.10 Modulo de procesos

Al ejecutar la opción **Transfer...** en la interfase de adquisición de imágenes, como aparece en la figura 5.18; se realiza el proceso de control de calidad. allí se invoca a las funciones encargan de los diferentes procesos.

BOOL CTwain::GetImage(TW_IMAGEINFO& info)

▶ ▶ ▶ Aquí realiza procesos con imagen adquirida.....

void CMainFrame::CopyImage(HANDLE hBitmap,TW_IMAGEINFO& info)

void CMainFrame::SetImage(HANDLE hBitmap,int bits)

void CProyectoXView::Imagen()

▶ ▶ ▶ Aquí ubica las imágenes.

El código fuente de estas funciones, se adjuntan en el apéndice A.

En la figura 5.19 se muestra los módulos que conforman el proceso central del sistema de control de calidad.

Cada modulo realiza una tarea especifica, la cual es parte del proceso global de control de calidad. A continuación se describe brevemente a cada una de ellas:

Carga imagen de memoria.- Transfiere el control de la imagen al programa, para allí realizar los procesos.

Obtiene imagen monocromática.- Obtiene imagen monocromática usando 256 niveles de gris [0-255], a partir de la imagen a color RGB.

Filtra imagen monocromática.- Realiza filtrado para mejorar su calidad.

Obtiene negativo de imagen.- Genera el negativo de la imagen monocromática para mejorar su calidad.

Obtiene histograma.- Determina el histograma de la imagen a nivel de gris.

Binariza imagen.- Mediante el proceso de umbralización obtiene una imagen en B/N.

Detecta contornos de imagen.- Contornea la imagen en B/N.

Calcula descriptores de imagen.- Calcula descriptores de la imagen, tales como: áreas,

número de dientes, color, etc.

Evalúa resultados.- Analiza los resultados obtenidos con los valores patrón para realizar la discriminación o no.

Emite reporte.- Emite un resultado final sobre el engranaje analizado.

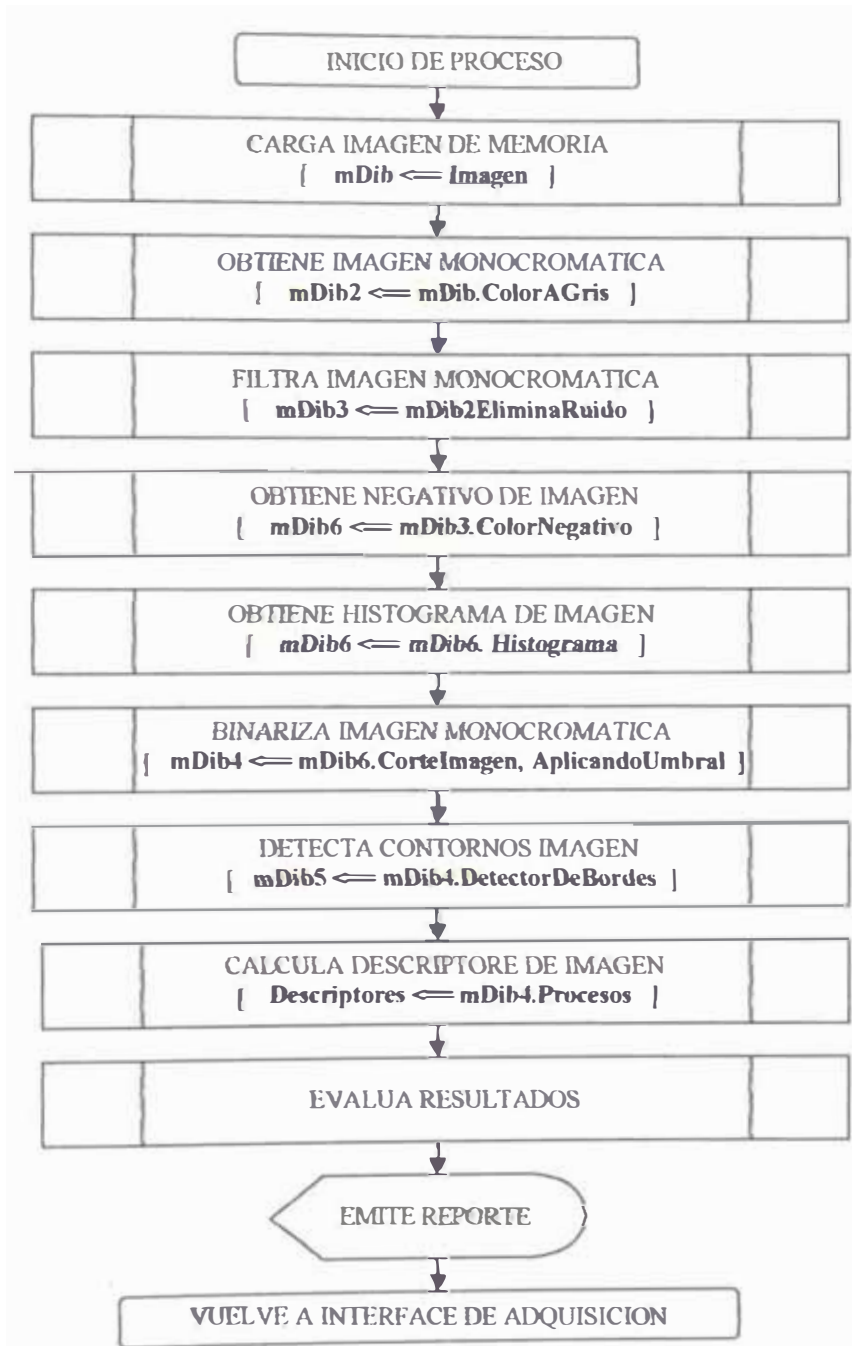


Figura 5.19- Modulo que se ejecutan al realizar el proceso de control de calidad.

5.4 Ejecución del sistema

5.4.1 Presentación de imágenes por el sistema

Al ejecutar el sistema, en la ventana principal aparecen 6 imágenes distribuidas como se muestra en la figura 5.20.

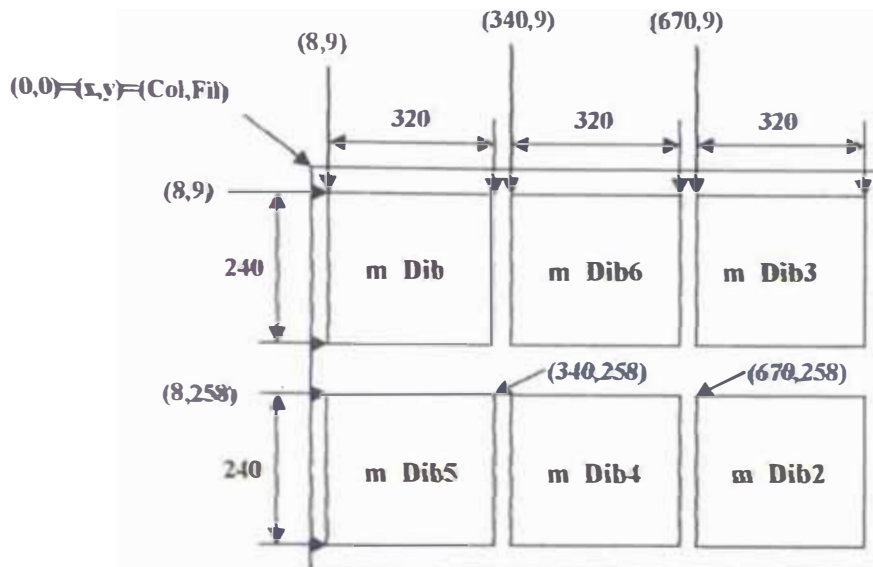


Figura 5.20: Distribución de imágenes en la ventana principal del sistema.

Donde:

- m Dib** = Imagen a color capturada
Coordenadas del vértice superior izquierdo = (8,9)
- m Dib2** = Imagen monocromática obtenida a partir de la imagen a color
Coordenadas del vértice superior izquierdo = (670,258)
- m Dib3** = Imagen monocromática filtrada
Coordenadas del vértice superior izquierdo = (670,9)
- m Dib6** = Imagen monocromática filtrada negativa
Coordenadas del vértice superior izquierdo = (340,9)
- m Dib4** = Imagen monocromática filtrada binarizada
Coordenadas del vértice superior izquierdo = (340,258)
- m Dib5** = Imagen monocromática filtrada binarizada contorneada
Coordenadas del vértice superior izquierdo = (8,258)

Luego de ejecutar un evento por la computadora, se deberá actualizar la imagen en pantalla; esto se realiza mediante el código

```
CRect rect;  
for (j = 0; j < 3; j++)  
    for (i = 0; i < 2; i++)  
        rect.left = 8 + j*(332);  
        rect.right = 328 + j*(332);
```



```
rect.top = 9 + i*(349);  
rect.bottom = 249 + i*(349);
```

La referencia de coordenadas para esta tarea se muestra en la figura 5.19.

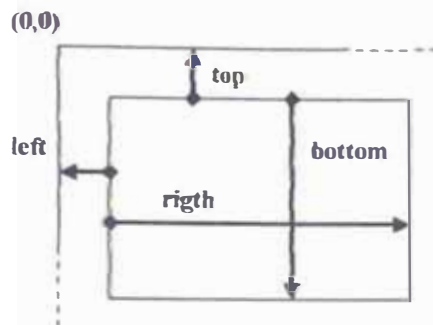


Figura 5.21: Referencia de coordenadas para el repintado.

Luego, de acuerdo a la configuración indicada, al ejecutar la aplicación se obtienen las imágenes que aparecen en la figura 5.22.



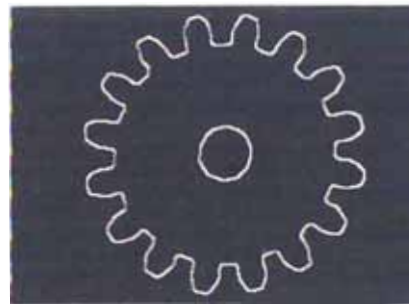
(a) **Imagen color**



(b) **Imagen negativa**



(c) **Imagen monocromática**



(d) **Imagen contorneada**



(e) **Imagen binarizada**



(f) **Imagen filtrada**

Figura 5.22: Gráficas mostradas por el sistema.

5.4.2 Determinación de áreas de referencia en el sistema

Se puede manipular el umbral, junto con el contraste ó dando valores apropiados en la interfase de configuración de la cámara para poder obtener imágenes apropiadas que permitan determinar el área total de la imagen (figura 5.23.a) y el área del agujero del engranaje (figura 5.23.b).

REPORTE FINAL DEL PROCESO:	
Color de Engrane:	179 Grado d
Tipo de Engrane:	E. Negro
Numero Dientes:	0 Dientes
Area de Engrane:	75200 mm
Area de Agujero:	
Coef. de Calidad:	

(a) Reporte del proceso



Imagen binarizada

REPORTE FINAL DEL PROCESO:	
Color de Engrane:	220 Grado d
Tipo de Engrane:	E. Negro
Numero Dientes:	0 Dientes
Area de Engrane:	73822 mm
Area de Agujero:	
Coef. de Calidad:	

(b) Reporte del proceso

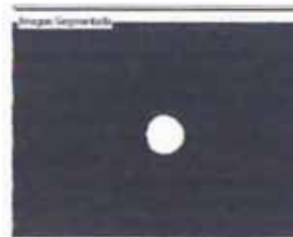


Imagen binarizada

REPORTE FINAL DEL PROCESO:	
Color de Engrane:	30 Grado
Tipo de Engrane:	E. Blanco
Numero Dientes:	16 Diente
Area de Engrane:	30140 m
Area de Agujero:	
Coef. de Calidad:	

(c) Reporte del proceso



Imagen binarizada

Figura 5.23: Determinando parámetros de áreas. (a) área total de la imagen. (b) área del agujero. (c) área del engranaje.

De la figura 5.23 podemos concluir:

área total de la imagen	≈ 75200 píxeles
área del agujero	≈ 75200 – 73820 ≈ 1380 píxeles
área del engranaje	≈ 30000 píxeles

5.4.3 análisis del color en el sistema

Una vez obtenida la imagen monocromática a partir de la imagen a color, tanto el fondo como el engranaje adquirirán para el sistema niveles de gris entre [0 - 255]. Debido a la relación de áreas entre el fondo y el engranaje, esto bajo condiciones adecuadas de adquisición, se tendrá un histograma como se muestra en la figura 5.24.

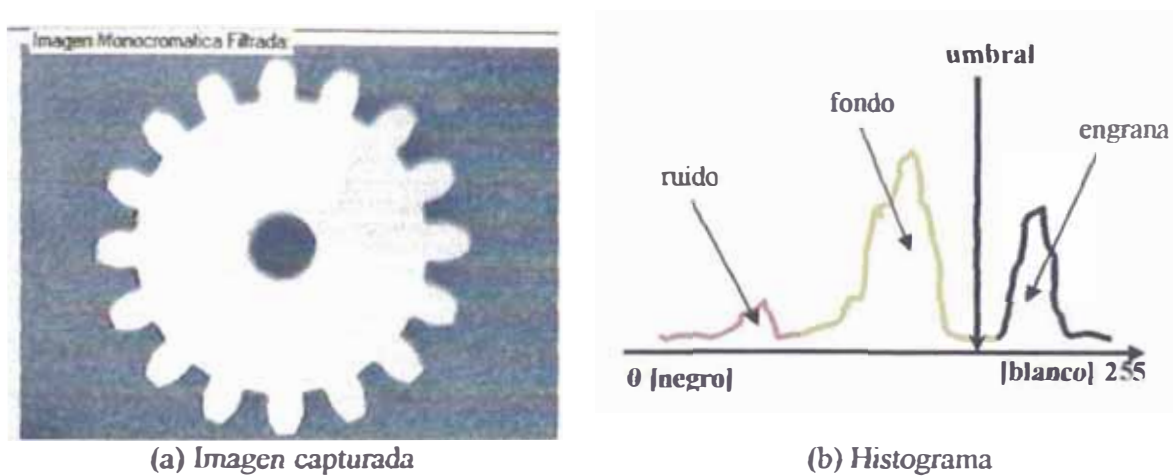


Figura 5.24: Imagen capturada y su histograma respectivo.

El histograma corresponde a la imagen m_Dib3 , la cual es monocromática filtrada. En la figura 5.24.b, el segundo mayor área del histograma corresponde al engranaje, el cual también ocupa el segundo área dentro de la imagen. La mayor área del histograma corresponde a la mayor área dentro de la imagen, es decir; el fondo. El área menor del histograma corresponde a puntos no deseados que aparecen sobre el fondo de la imagen. La correspondencia entre área del histograma y área de la imagen se puede ver en las imágenes (a) y (b) de la figura 5.24.

En la imagen de la figura 5.24 se puede observar que hay un buen contraste entre el engranaje y el fondo, lo cual implica que se puede realizar un análisis sobre el objeto de interés, es decir; el engranaje.

Así mismo, el contraste que se obtenga entre el engranaje y el fondo dependerá de:

- Los colores de ambos elementos.
- La pureza del color sobre el objeto.

- Sistema de iluminación.
- Calidad del dispositivo de adquisición.
- Configuración del dispositivo de adquisición.

Además de la calidad de la imagen obtenida por el sistema, el algoritmo que se utilice para el procesamiento, también será determinante en la precisión de los resultados.

A continuación se muestran los diferentes algoritmos de detección de color utilizados y los resultados obtenidos.

1. **Algoritmo del color dominante**, aquí se verifica la presencia del mayor nivel de gris en la imagen. Los resultados obtenidos se muestran en la figura 5.25 y en la tabla 5.4.

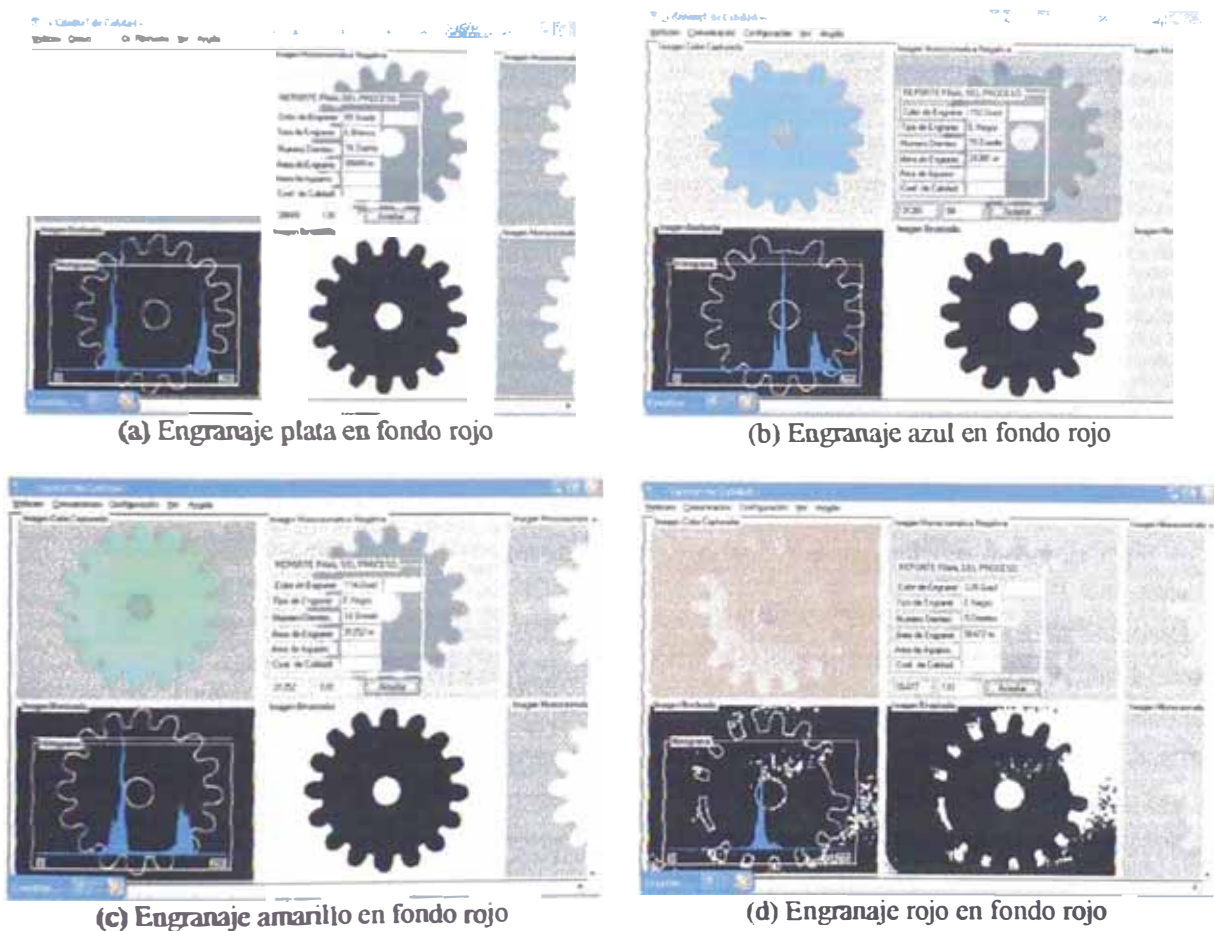


Figura 5.25: Resultados gráficos de aplicar el algoritmo del color dominante.

Tabla 5.4 Resultados de aplicar el algoritmo del color dominante.

Engranaje	Fondo	Umbral	BackLigth	Color	Imagen	Figura
Plata	rojo	120	1	83	ok	2.a
Azul	rojo	90	1	152	ok	2.b
Amarillo	rojo	120	1	114	ok	2.c
Rojo	rojo	130	1	129	xx	2.d

2. **Algoritmo del segundo color dominante**, aquí se verifica la presencia del segundo mayor nivel de gris en la imagen. Los resultados obtenidos se muestran en la figura 5.26 y en la tabla 5.5.

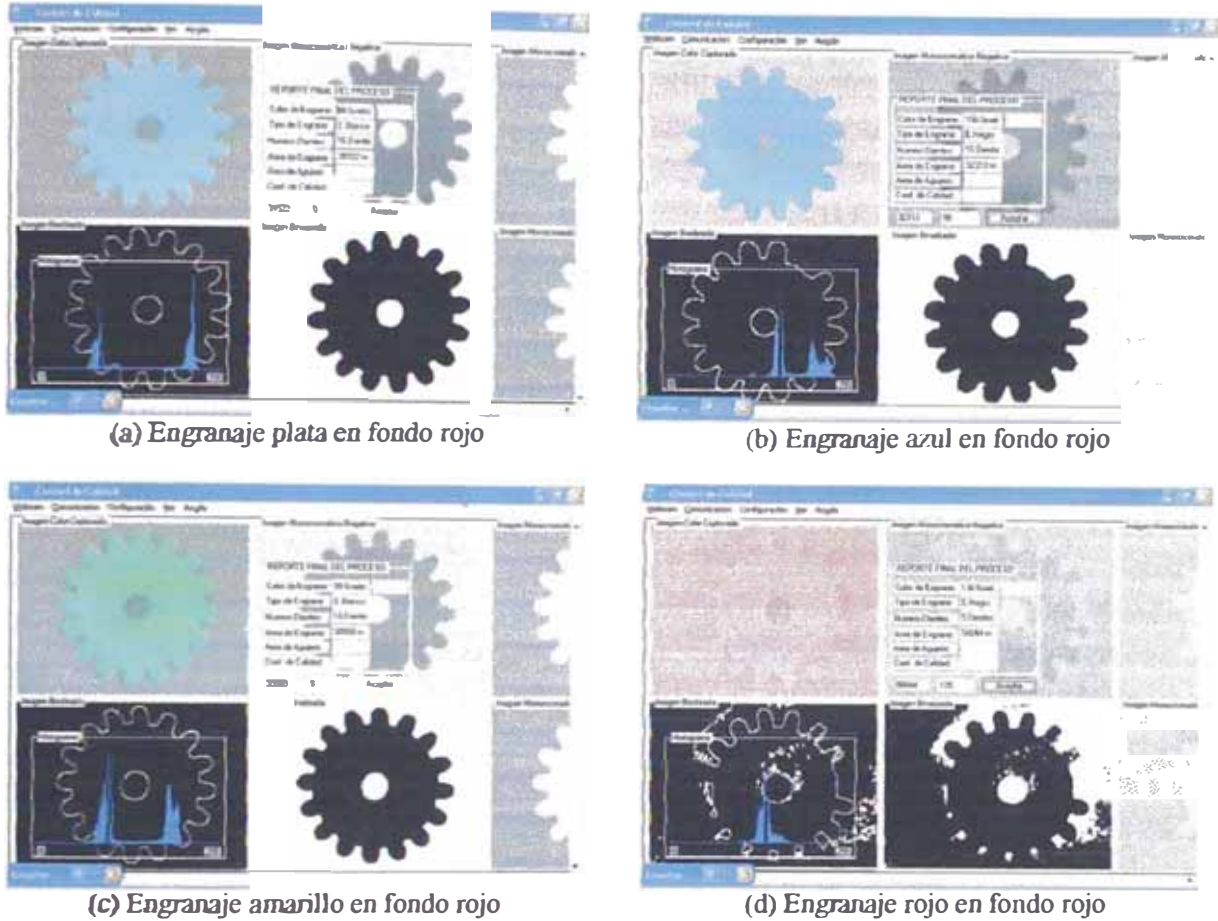


Figura 5.26: Resultados gráficos de aplicar el algoritmo del segundo color dominante.

Tabla 5.5 Resultados de aplicar el algoritmo del segundo color dominante

Engranaje	Fondo	Umbral	Background	Color	Imagen	Figura
plata	Rojo	120	1	88	ok	3.a
azul	Rojo	90	1	156	ok	3.b
amarillo	Rojo	120	1	99	ok	3.c
rojo	Rojo	125	1	136	xx	3.d

3. **Algoritmo del promedio global**, se verifica el promedio de nivel de gris en toda la imagen. Los resultados obtenidos se muestran en la figura 5.27 y en la tabla 5.6.

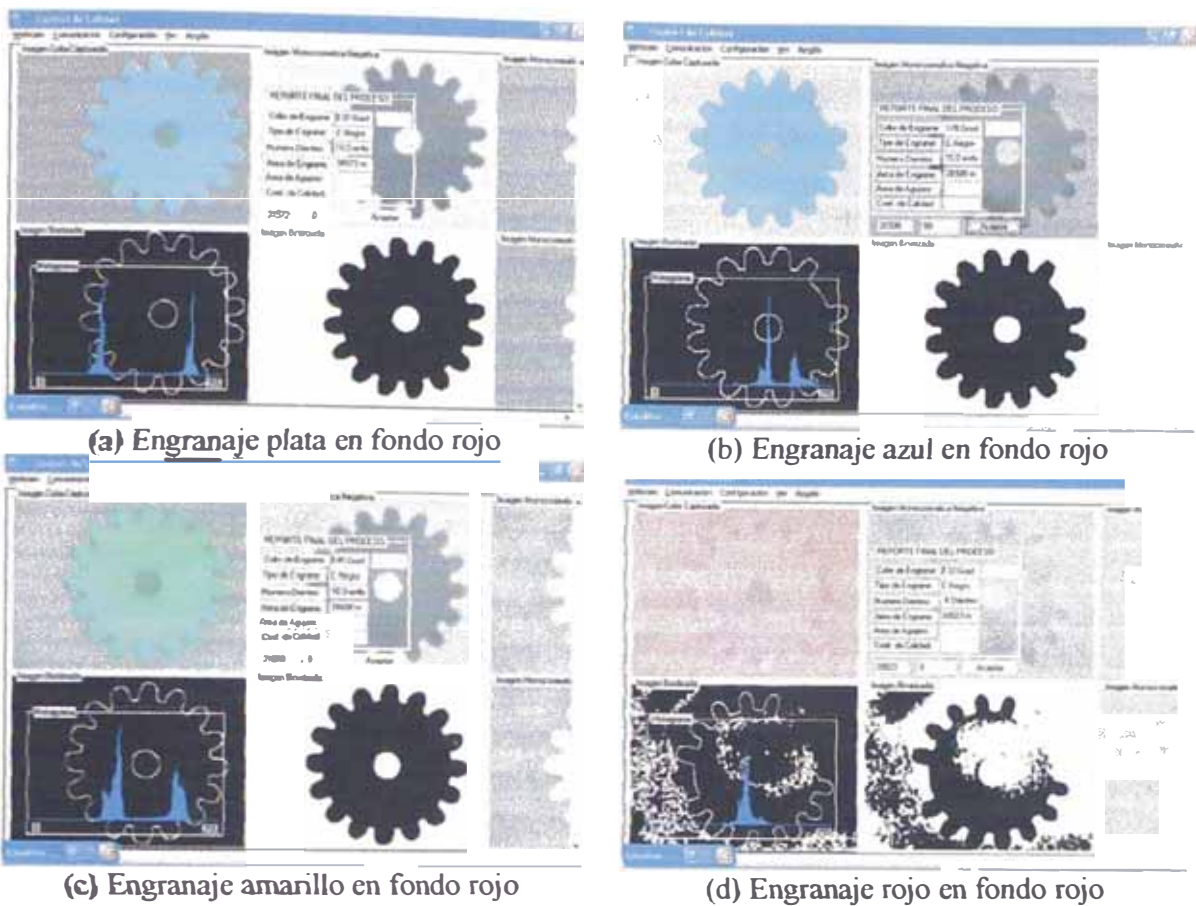
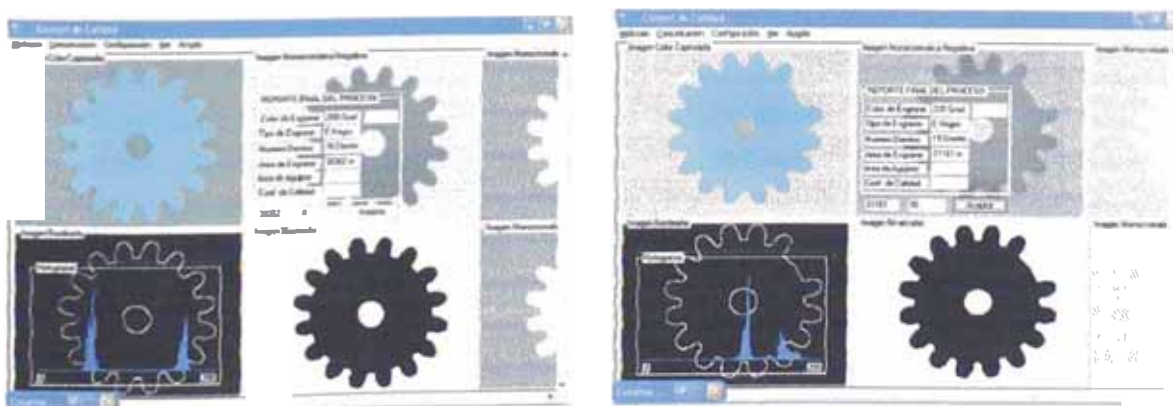


Figura 5.27: Resultados gráficos de aplicar el algoritmo del promedio global.

Tabla 5.6 Resultados de aplicar el algoritmo del promedio global.

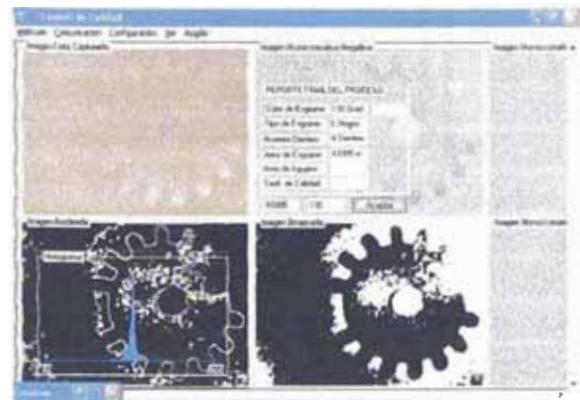
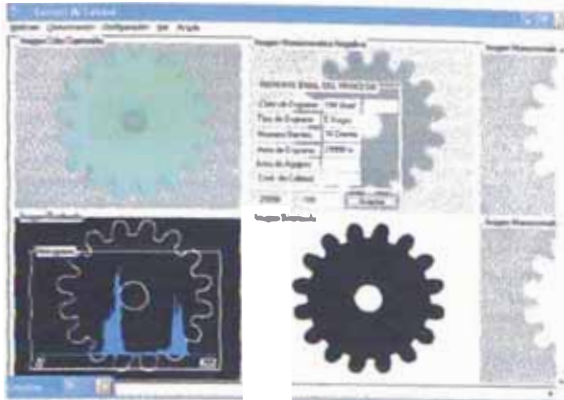
Engranaje	Fondo	Umbral	BackLigth	Color	Imagen	Figura
Plata	rojo	120	1	137	ok	4.a
azul	rojo	80	1	179	ok	4.b
amarillo	rojo	120	1	145	ok	4.c
rojo	rojo	120	1	133	xx	4.d

4. **Algoritmo del promedio local (en el engranaje)**, aquí se verifica el promedio de nivel de gris solo en el rango correspondiente a la presencia de engranaje. Los resultados obtenidos se muestran en la figura 5.28 y en la tabla 5.7.



(a) Engranaje plata en fondo rojo

(b) Engranaje azul en fondo rojo



(c) Engranaje amarillo en fondo rojo

(d) Engranaje rojo en fondo rojo

Figura 5.28: Resultados gráficos de aplicar el algoritmo del promedio local.

Tabla 5.7 Resultados de aplicar el algoritmo del promedio local.

Engranaje	Fondo	Umbral	Backl.isth	Color	Imagen	Figura
Plata	rojo	120	1	200	ok	5.a
Azul	rojo	80	1	200	ok	5.b
Amarillo	rojo	100	1	194	ok	5.c
Rojo	rojo	130	1	130	xx	5.d

Viendo el detalle de los histogramas (ver figura 5.29) podemos observar que para los engranajes color plata, azul y amarillo se distingue claramente el contraste, es decir; se puede realizar el proceso obteniéndose resultados apropiados. En el caso del engranaje rojo, para las condiciones de prueba (fondo rojo), se observa que no se puede obtener contraste, por lo tanto, los resultados del proceso en este caso serán erróneos.



(a) Engranaje plata - fondo rojo



(b) Engranaje azul - fondo rojo

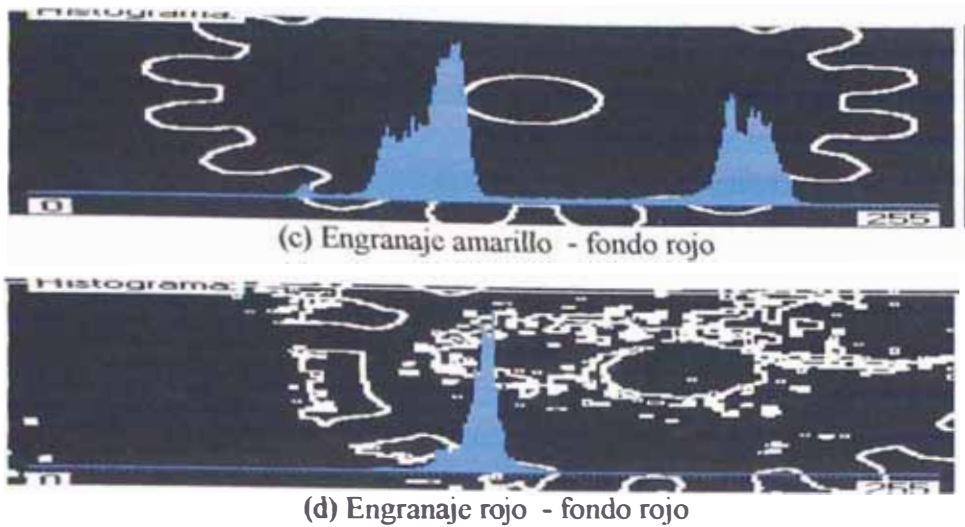


Figura 5.29: Histograma de los diferentes engranajes.

Podemos concluir entonces, que bajo condiciones apropiadas de adquisición de imagen el **contraste** entre el histograma y el fondo de la imagen es notorio, en este caso fondo rojo; por lo tanto se puede realizar un buen trabajo de análisis del engranaje. Obviamente esto no es posible para el engranaje rojo.

Como existen problemas para identificar la información del color rojo, en este caso se puede verificar utilizando diferentes fondos para ver la capacidad de discriminación del sistema. En la figura 5.30 se muestra como utilizando fondo aluminio (figura 5.30.a) y fondo blanco (figura 5.30.b) se puede discriminar perfectamente el engranaje del fondo y de esa manera poder procesar la información en forma adecuada.

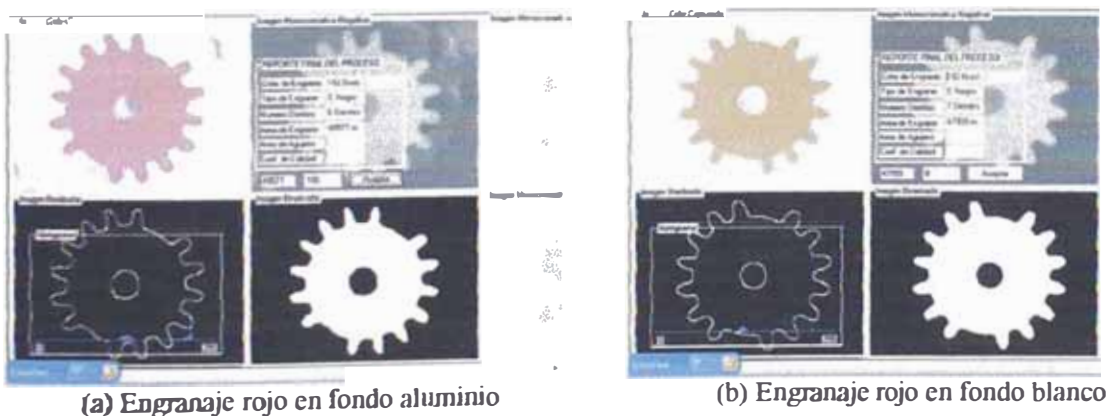


Figura 5.30: Resultados gráficos al cambiar el fondo de la imagen.

Además como ya se menciona anteriormente, se deberá trabajar con una configuración apropiada para la de interfase de adquisición de imágenes.

En todos los casos descritos para la evaluación del color, se utilizo la interfase de adquisición de imágenes con la configuración mostrada en la figura 5.31.

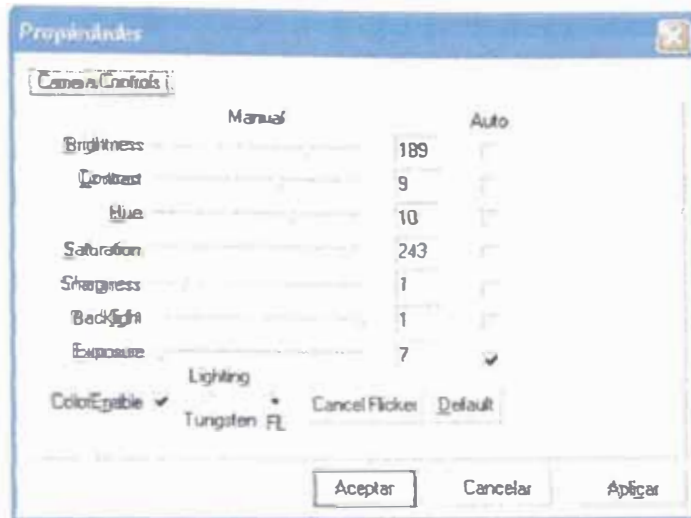


Figura 5.31: Interfase de adquisición de imágenes.

Para poder distinguir el objeto de análisis dentro de la imagen, se debe buscar que el **contraste** sea lo máximo posible, es decir que el color del engranaje y de fondo traten de ser **negativos** entre si. En la figura 5.32 se muestran algunos colores y sus respectivos negativos.

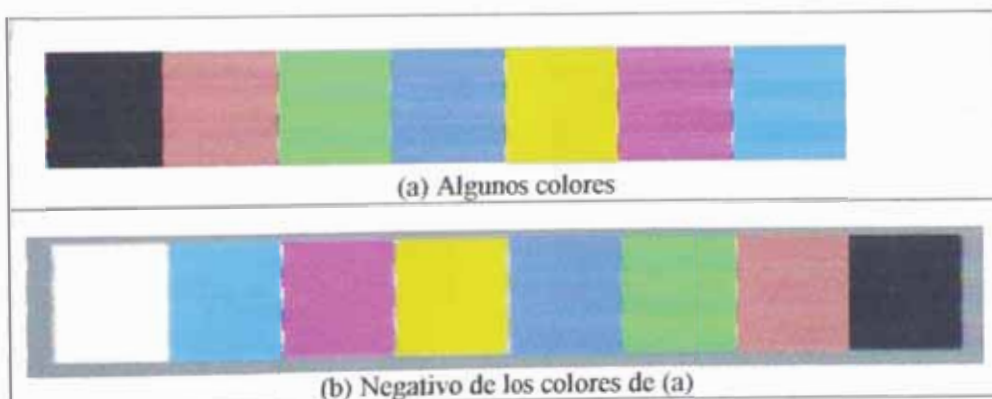


Figura 5.32: Algunos colores y sus respectivos negativos.

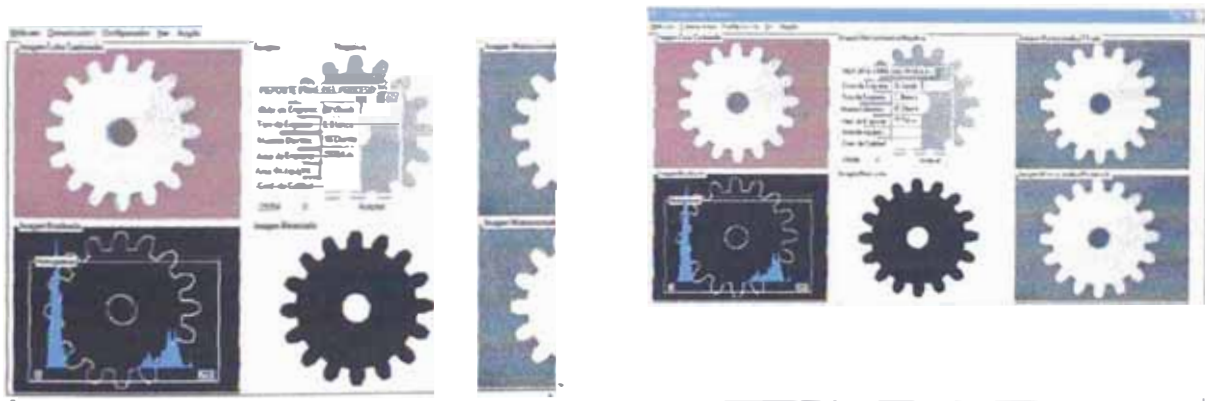
CAPITULO 6

Resultados y Conclusiones

6.1 Configuración del monitor

Al ejecutar el programa, el tamaño de la ventana principal y del número de imágenes que pueda mostrar depende de la configuración del monitor. En la figura 6.1 se muestran 2 configuraciones diferentes para un mismo monitor.

El área de trabajo del programa (ventana principal), puede albergar cuando esta maximizado inclusive mas de 6 imágenes, dependiendo de la configuración de la pantalla de la computadora y la distribución de las mismas.



Ejecutando el sistema en computadora:
Procesador : PC Pentium IV – INTEL
Velocidad : 1.70 GHz
Memoria : 256 MB
Monitor : Samsung 17"
Configuración : Resolución de 800 x 600

(a)

Ejecutando el sistema en computadora:
Procesador : PC Pentium IV – INTEL
Velocidad : 1.70 GHz
Memoria : 256 MB
Monitor : Samsung 17"
Configuración : resolución de 1024 x 768

(b)

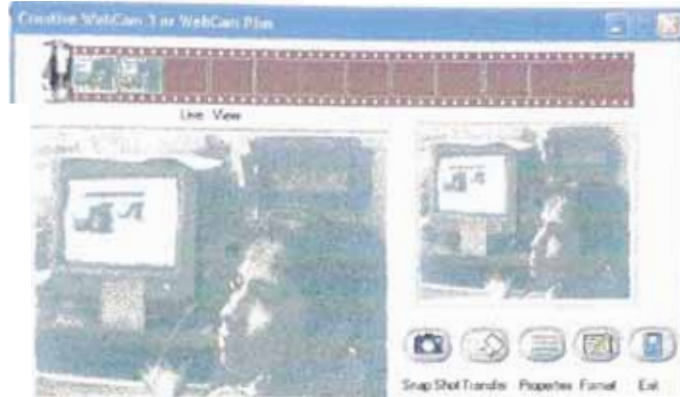
Figura 6.1: Presentación del sistema para diferentes configuraciones.

6.2 Interfases de captura de imágenes

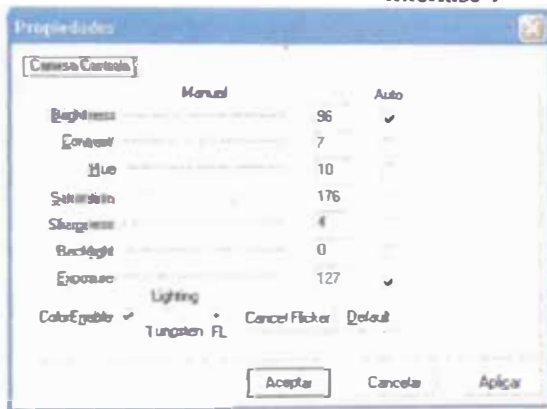
De acuerdo a la cámara, la interfase de adquisición y su configuración, variará la calidad de la imagen. La interfase de adquisición dependerá de los drivers y del software

de control, además del soporte de hardware (computadora). En la figura 6.2 se muestran 3 interfaces de control para 3 cámaras diferentes, utilizadas en las pruebas.

Interfase Creative WebCAM 3 or WebCam Plus



Interfase 1



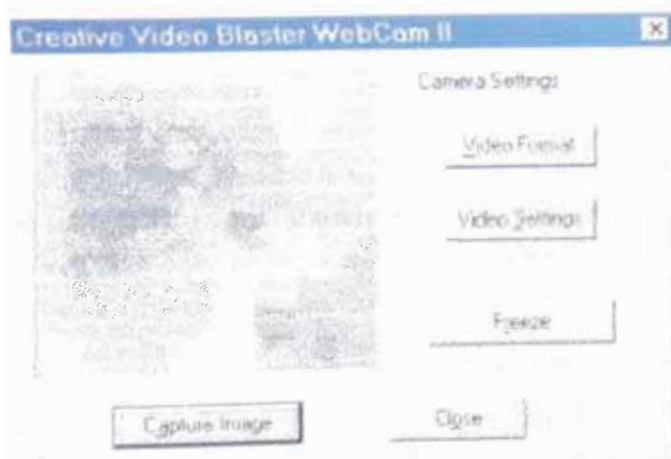
Properties



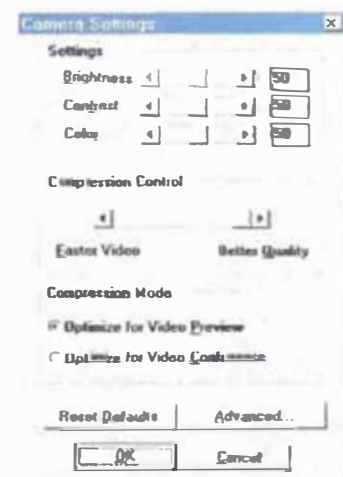
Format

(a)

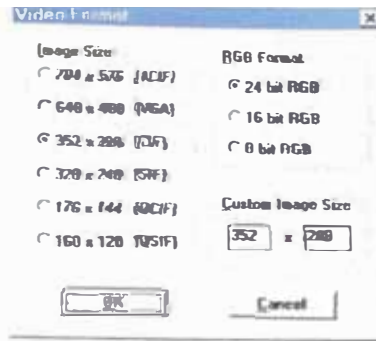
Interfase Creative WebCAM II



Interfase 2

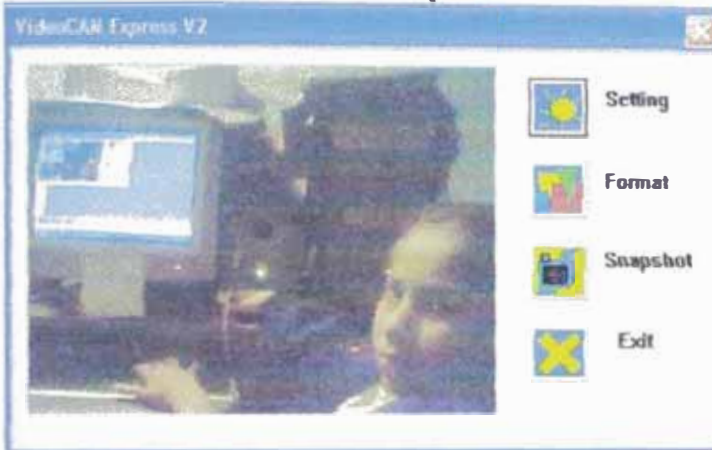


Camera Setting



Video Format
(b)

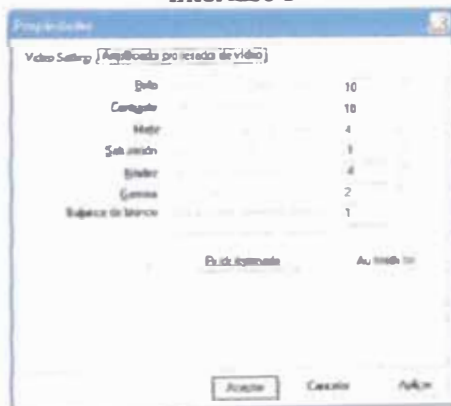
Interface Genius VideoCAM Express V2



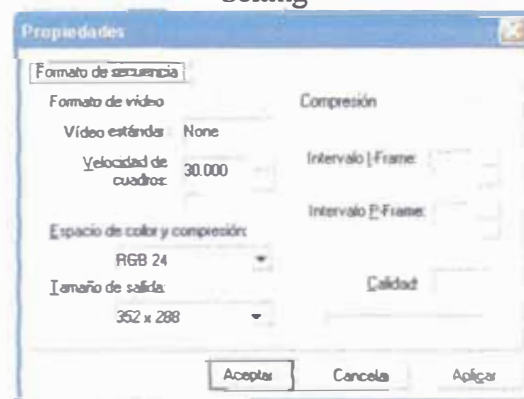
Interfase 3



Setting



Setting



Format

(c)

Figura 6.2: Tres tipos de cámaras con sus respectivas interfaces de configuración.
(a) Creative WebCam Plus. (b) Creative WebCAM II. (c) Genius VideoCAM Express V2

El tipo de drivers y software de control son las que definen la interfase y la forma de adquisición de la imagen, al igual que las funciones que manejan los eventos de la interfase.

Es decir, el tipo de interfase de software y el hardware de algún modo, condicionara la respuesta de la aplicación desarrollada.

Durante el desarrollo del sistema se hicieron pruebas sobre el soporte de software y

hardware y se obtuvieron los siguientes resultados:

Tabla 6.1: Respuesta de las diferentes interfasas en diferentes PC's.

	Interfase 1	Interfase 2	Interfase 3
Pentium IV	Trabaja en la PC	No reconoce la PC	No es reconocida por la clase CTwain
Pentium III	Trabaja en la PC	Trabaja en la PC	No es reconocida por la clase CTwain
Pentium I	No soporta la PC	Trabaja en la PC	No es reconocida por la clase CTwain

6.3 Engranajes de diferentes colores sobre fondos diferentes

El color del fondo de la imagen, deberá ser elegido de tal forma que puede contrastar con el engranaje, caso contrario será imposible analizar el objeto.



(a) rojo - blanco



(b) amarillo - blanco



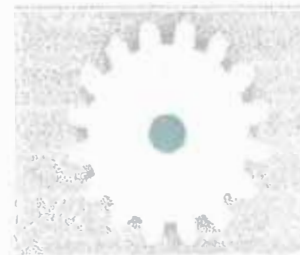
(c) azul - blanco



(d) plata - blanco



(e) rojo - rojo



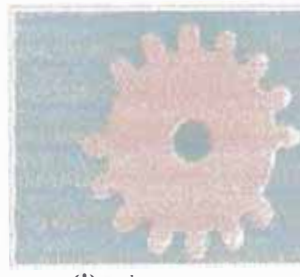
(f) amarillo - rojo



(g) azul - rojo



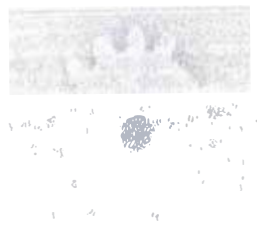
(h) plata - rojo



(i) rojo - negro



(j) amarillo - negro



(k) azul - negro



(l) plata - negro

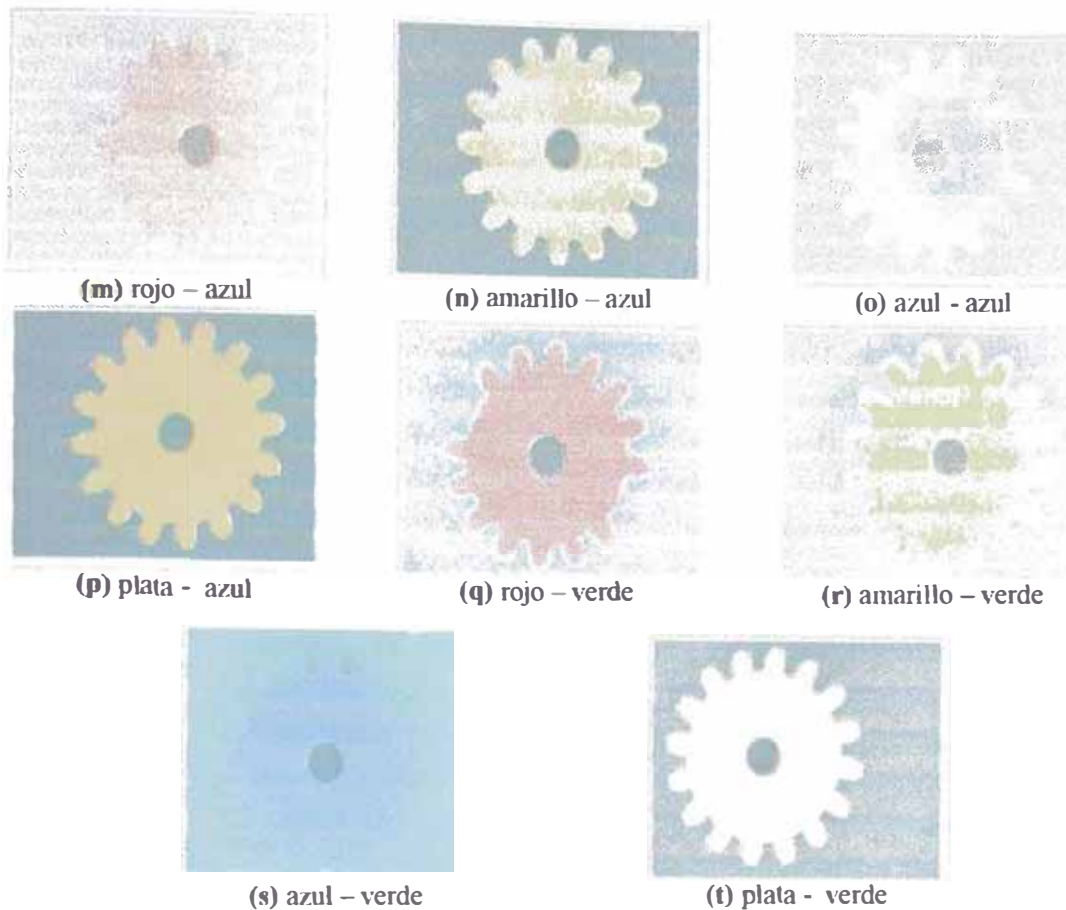


Figura 6.3: Diferentes combinaciones de color de engranaje y color de fondo.

En la figura 6.3 se muestran algunas combinaciones de engranajes y fondos. Inclusive se puede apreciar visualmente que de acuerdo al color del engranaje y el color de fondo, en algunos casos es difícil de discriminar el objeto de interés (ver figuras 6.3.d, e, k, o), por lo tanto no se podrá realizar una evaluación adecuada. En otros casos es notorio el contraste por lo que se podrá evaluar el engranaje dentro de la imagen capturada (ver figuras 6.3.a, f, h, l).

6.4 Efecto de la configuración de la interfase

Además de la iluminación y los colores tanto del engranaje como del fondo, el efecto de la configuración de la interfase de adquisición de imagen ó programa monitor también determinara la calidad de la imagen obtenida. Dicho efecto se puede apreciar en la figura 6.4; allí se aprecia como conservando el color del engranaje y el fondo cambia la imagen adquirida de acuerdo a la configuración. Para la figura 6.4 en todos los casos se mantuvo el fondo rojo.

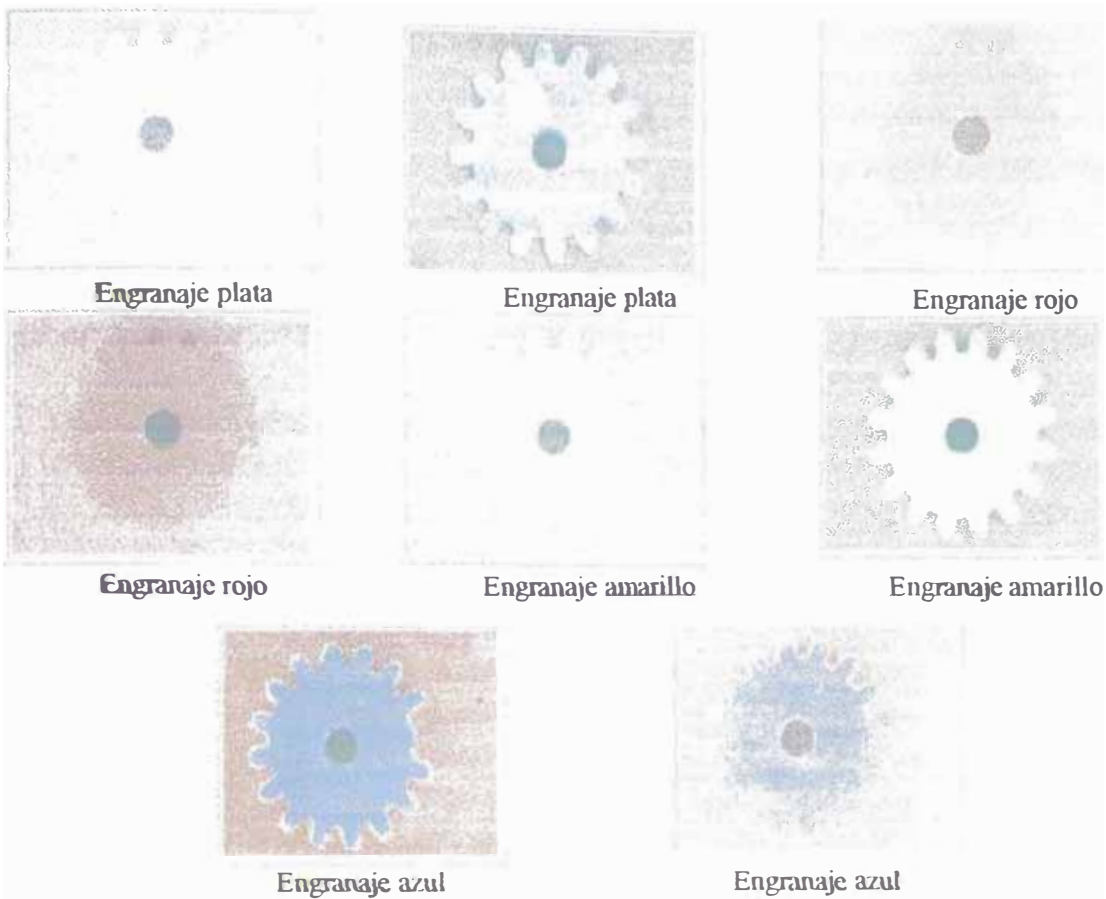


Figura 6.4: Efecto de la configuración en la imagen. En todas se mantiene el fondo rojo.

El mismo efecto se muestra en la figura 6.5, pero esta vez se muestra la interfase de configuración para obtener un buen contraste en 2 casos particulares.

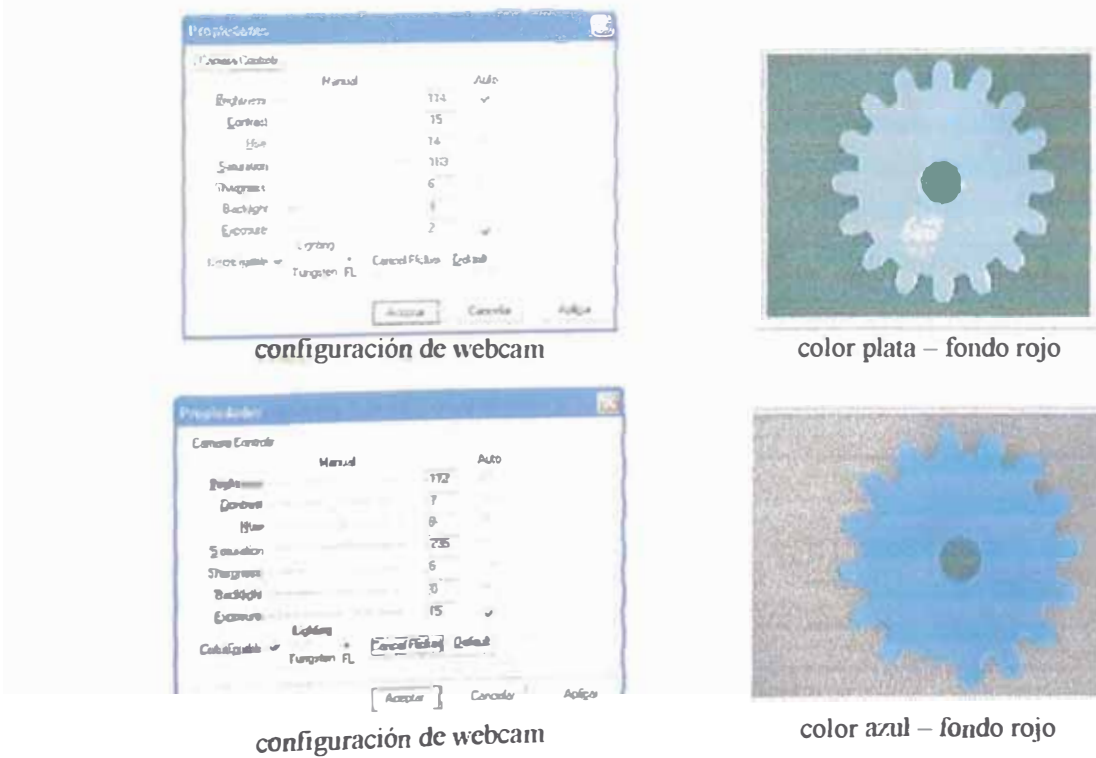
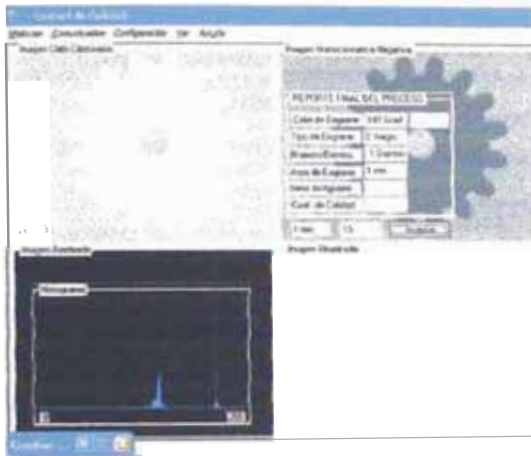


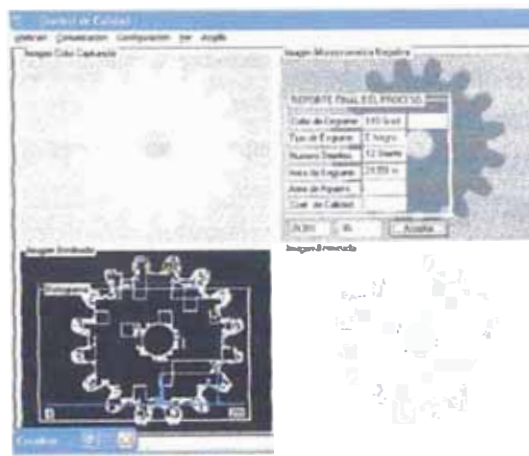
Figura 6.5: Configuración interfase para 2 casos particulares.

6.5 Efecto del valor del umbral

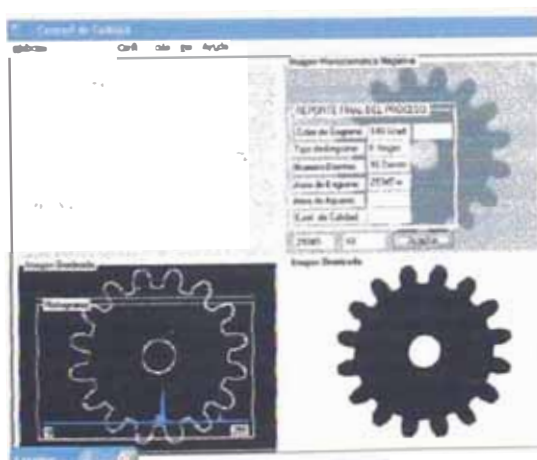
El valor del umbral será determinante en la respuesta del sistema, debido a que si este valor **no separa** el objeto bajo análisis (engranaje) de la información **no deseada** (fondo), los resultados obtenidos serán erróneos. Este efecto se puede apreciar en las figuras 6.6.c, 6.6.d, 6.6.e, donde el engranaje queda separado del resto de la escena, para un valor de umbral dentro del un rango [40, 95].



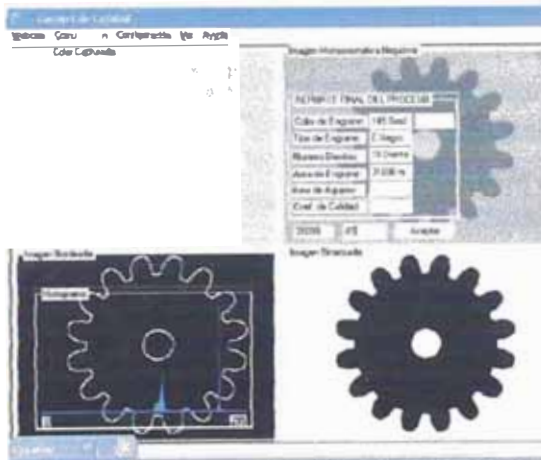
(a) Umbral 15



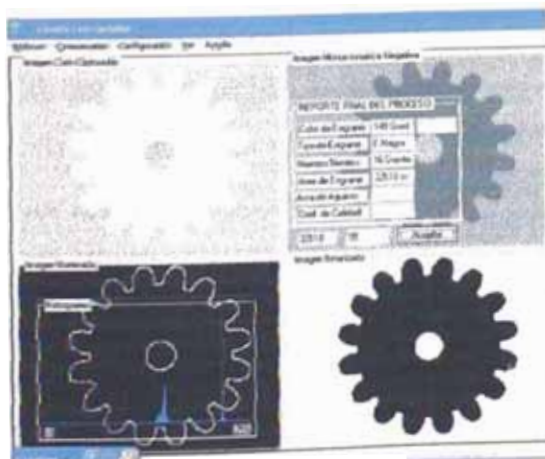
(b) Umbral 35



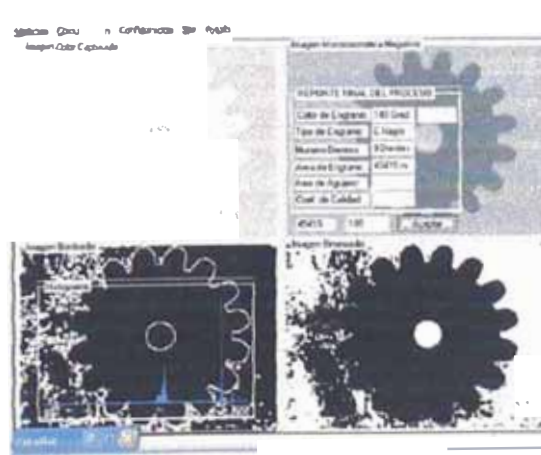
(c) Umbral 40



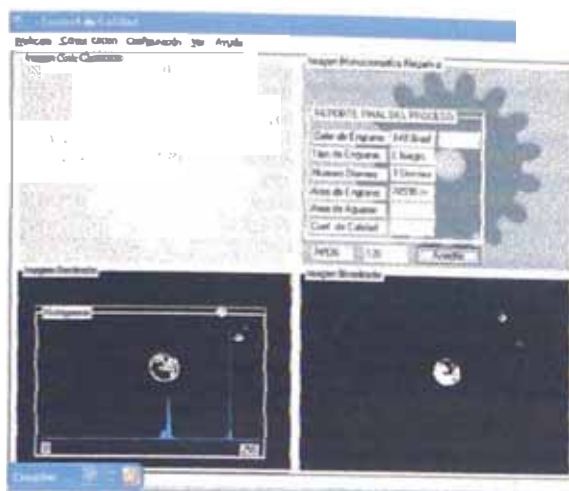
(d) Umbral 85



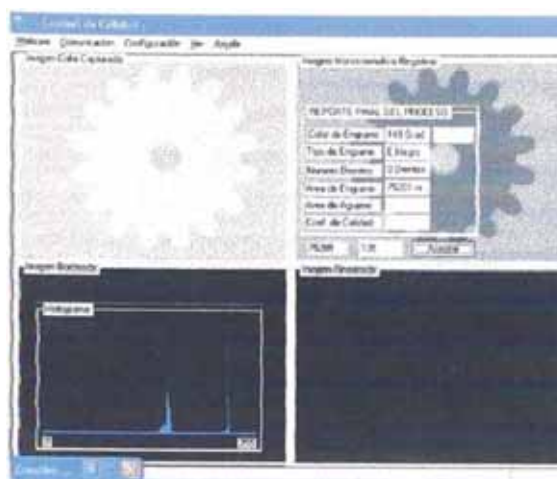
(e) Umbral 95



(f) Umbral 105



(g) Umbral 120



(h) Umbral 135

Figura 6.6: Efecto de valor de umbral en la binarización de la imagen.

A partir de las pruebas realizadas, para nuestro sistema para los valores de umbral mostrados en la tabla 6.2 se obtiene una respuesta adecuada.

Tabla 6.2: Valores de umbralización adecuados para el proceso.

Engranaje	Fondo	Umbral	Color	Imagen	BackLight
plata	rojo	80	98	ok	1
azul	rojo	80	151	ok	1
amarillo	rojo	120	111	ok	1
rojo	rojo	80	185	xx	1

(*) Reconocimiento de color por promedio local fondo.

El sistema por defecto se puede umbralizar para un valor entre [75,95]. Específicamente en este caso para 85.

6.6 Efecto de movimiento del engranaje

Otro parámetro importante a considerar en un proceso de tratamiento digital de imágenes mediante visión artificial es el efecto de movimiento de la imagen que se desea capturar. Mientras más rápido se mueva la imagen, mayor será la distorsión; esto además de las características de la cámara.

Si la distorsión es mínima, podría ser corregido con algoritmos adecuados, de ser mayor la distorsión será imposible realizar el proceso.

En la figura 6.7.a se muestra como el proceso se realiza sin inconvenientes para una imagen en reposo; mientras que en la figura 6.7.g, se aprecia que cuando la velocidad es mayor será imposible detectar el objeto de interés.

Para demostrar el efecto en las figuras 6.7.b y 6.7.c se movió el engranaje en forma manual variando su velocidad. En las figuras 6.7.d, 6.7.e, 6.7.f, 6.7.g se utilizó a una mano en movimiento, aumentando su velocidad en forma gradual. En la figura 6.7.g se puede apreciar que la mano prácticamente no aparece en la fotografía.

6.7 Efecto de iluminación

Otro parámetro a considerar también es la iluminación que se realice sobre el objeto de interés. En la figura 6.8 se puede apreciar 4 diferentes condiciones de iluminación y el resultado sobre el proceso de análisis del engranaje. En la figura 6.8.b no se puede separar el engranaje del fondo, por lo tanto el proceso no se puede realizar, mientras que en los otros casos se logra separar el engranaje.

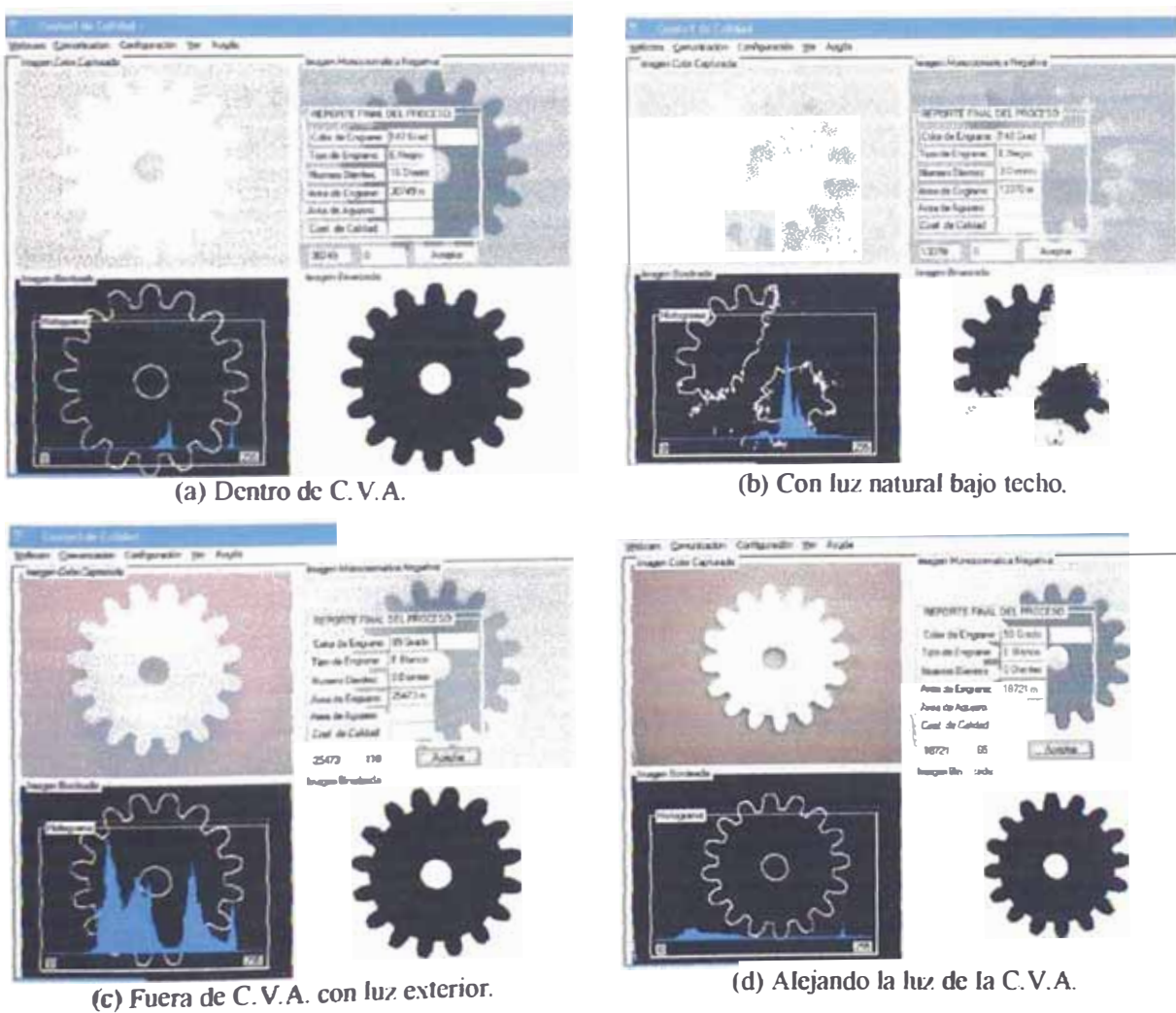


Figura 6.8: Efecto de iluminación.

6.8 Cambio de ubicación de pieza dentro de la cámara de visión artificial (CVA)

Se debe tener en cuenta que la ubicación de la pieza dentro de la C.V.A. modificara los **parámetros** de la imagen. Este efecto se puede apreciar en la figura 6.9, allí se aprecia como la figura 6.9.b al ser corrida hacia la derecha figura 6.9.c ó hacia la izquierda figura 6.9.a **distorsiona** su forma, y esto según sea el caso inducirá a error en los resultados.

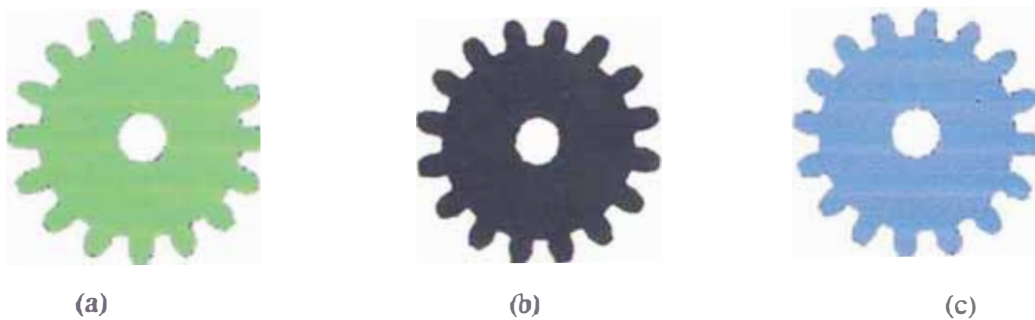


Figura 6.9: Efecto de la ubicación del engranaje dentro de la cámara de visión.

El mismo efecto se ilustra en la figura 6.10, donde los vértices de los ‘cuadrados’ que **contienen** al engranaje central (rojo) se observa que no coinciden en dimensión con los **vértices** de los ‘cuadrados’ azules (arriba y a la derecha) y los vértices de los ‘cuadrados’ verdes (abajo y a la izquierda), esto demuestra que el engranaje se ha distorsionado en la **imagen**

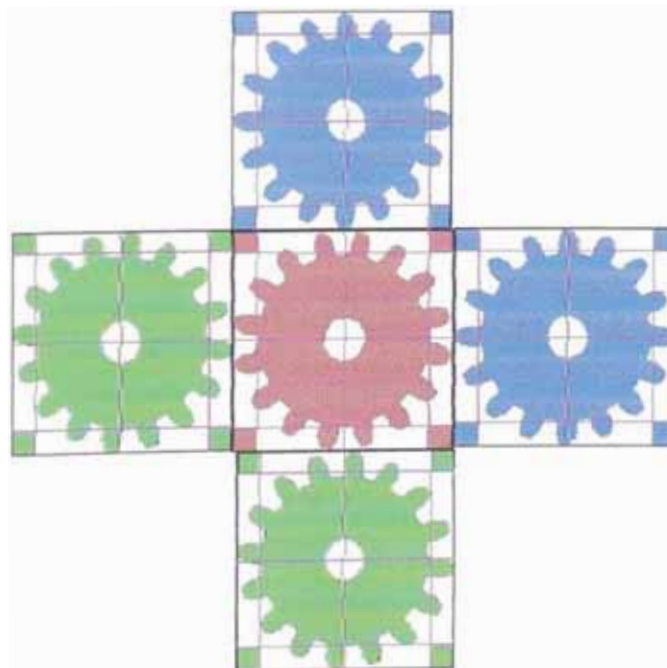


Figura 6.10: Efecto de la ubicación del engranaje dentro de la cámara de visión.

6.9 Pruebas de áreas y numero de dientes

Tomando en cuenta los diferentes criterios desarrollados anteriormente se realizaron pruebas y se obtuvieron los resultados mostrados en la tabla 6.3. La figura 6.10 muestra los valores correspondientes al engranaje patrón, obtenidos luego de las diferentes pruebas realizadas.

Tabla 6.3: Resultados de algunas pruebas realizadas con el sistema.

N°	Pieza	Fondo	Area[píxel]	Agujero[píxel]	Color	Dientes	Observaciones
01	EP	Rojo	20797	10583	31		Imagen no ok
02	E2	Rojo	8880		222		Engranaje diferente
03	E3	Rojo	9485		72		Engranaje diferente
04	EAM	Rojo	21135	10245	41		
05	ER	Rojo	67940		177		
07	ER	Blanco	61541		220		

N°	Pieza	Fondo	Area[píxel]	Agujero[píxel]	Color	Dientes	Observaciones
09	EP	rojo	34151	2779	221	16	Distorsionada
10	EP	rojo	30101	1279	43	16	Imagen ok
11	ER	rojo	68200		141	5	Todo casi negro
12	EAZ	rojo	28867	2513	42	16	Imagen casi ok

N°	Pieza	Fondo	Area[píxel]	Agujero[píxel]	Color	Dientes	Observaciones
13	EAM	amarillo	28958	2422	42	16	Imagen casi ok
14	EAZ	amarillo	54164		147	9	Todo casi negro
15	ER	amarillo	75199		211	0	TUDO NEGRO
16	EAM	amarillo	2		54	1	TUDO BLANCO
17	ER	blanco	46057		222	3	Imagen s/agujero
18	EAM	blanco	73822		220	0	SOLO AGUJERO

N°	Pieza	Fondo	Area[píxel]	Agujero[píxel]	Color	Dientes	Observaciones
19	EP	rojo	74223		229	0	Para B.L.= 0
20	EAZ	rojo	73776		219	1	Para B.L.= 0
22	EP	negro	29340	2040	60	16	Casi rojo B.L.= 1
23	ER	negro	73683		184	0	B.L.= 0
24	EAZ	negro	28865	2515	54	16	B.L.= 1
25	E1	azul	29659	1721	85	16	B.L.= 0
27	ER	verde	44818		102	8	B.L.= 0

N°	Pieza	Fondo	Area[píxel]	Agujero[píxel]	Color	Dientes	Observaciones
28	E1	rojo	30304	1076	41	16	B.L.= 1
29	EAZ	rojo	29665	1715	45	16	B.L.= 1

(*) Notas:

- Todas las pruebas se realizaron usando un valor de umbral de 120.
- B.L. se refiere a la opción de BackLigth de configuración de la cámara.
- Donde:
 - EP = Engranaje color aluminio
 - E2 = Engranaje color aluminio
 - E3 = Engranaje color aluminio
 - EAM = Engranaje color amarillo
 - ER = Engranaje color rojo
 - EAZ = Engranaje color azul
 - E1 = Engranaje color aluminio referencial

La figura 6.11 muestra en un cuadro los resultados de las pruebas realizadas.

Los parámetros de referencia del engranaje patrón se muestran en la figura 6.10.

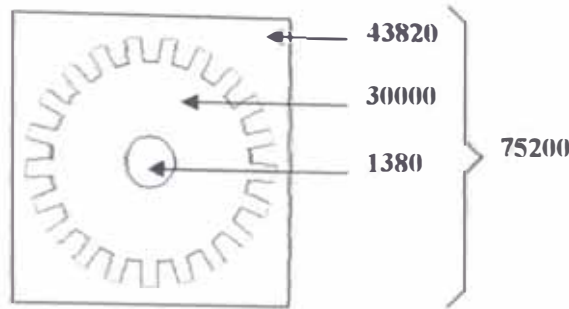


Figura 6.11: Parámetros de área del engranaje patrón.

6.10 Tiempos de ejecución

Se puede calcular teóricamente el tiempo de proceso, mediante los ciclos de ejecución de las instrucciones utilizadas en los diferentes procesos.

En la tabla 6.4 se muestra el código de la función: `void CDIB::Dientes()` y los tiempos consumidos durante la ejecución.

Tabla 6.4: Ciclos de instrucción para un proceso determinado.

Código	Nº de Veces	Ciclos de Instrucción	Total Ciclos de Instrucción
<code>int i, j, x, y, ii=0, jj=0, xe1, ye1, xe2, ye2, cont=0;</code>	1	aT	aT
<code>long int xx=0, ax=0, cx, xm, yy=0, ay=0, cy, ym;</code>	1	aT	aT
<code>float alfa, teta, xr1, yr1, xr2, yr2, r=75;</code>	1	aT	aT
<code>unsigned char *ptr;</code>	1	aT	aT
<code>if(m pInfo->bmiHeader.biBitCount == 24)</code>	1	bT	bT
<code>int E[240][320];</code>	1	aT	aT
<code>for(i=0; i<240; i++)</code>	1	cT	cT
<code>for(j=0; j<320; j++)</code>	1	cT	cT
<code>E[i][j]=5;</code>	240x320	dT	76800dT
<code>for(i=0, ptr = m pBits; l < height; i++)</code>	1	cT	1cT
<code>{ jj=0;</code>	240	dT	240dT
<code>ptr = m pBits + i*bytes;</code>	240	dT	240dT
<code>for(i=0; j < width ; j++, ptr+=3)</code>	240	cT	240cT
<code>{ ptr[j] = (BYTE)(ptr[j]);</code>	240x320	dT	76800dT
<code>if(ptr[0]==255)</code>	240x320	bT	76800bT
<code>E[ii][jj]=1;</code>	240x320	dT	76800dT
<code>else</code>			
<code>E[ii][jj]=0;</code>	240x320/2	dT	38400dT
<code>Jj=jj+1;</code>	240x320	dT	76800dT
<code>ii=ii+1;</code>	240x320	dT	76800dT
<code>for(y=0; y<320; y++)</code>	1	cT	1cT
<code>{ cx=0;</code>	320	dT	320dT
<code>for(x=0; x<240; x++)</code>	320	cT	320cT

$$T_{\text{proceso}} = 4520543.1 \times 10^{-9} = 4.5205431 \text{ mSg}$$

De forma similar se puede calcular el tiempo de los diferentes procesos del sistema.

6.11 Estadísticas

Para poder llegar al resultado final del sistema se tuvieron que realizar muchas pruebas y modificaciones en forma reiterada, tanto en el entorno de captura de imagen, así como; en los valores de los parámetros del sistema de software. En la tabla 6.5 se muestra un resumen de los resultados obtenidos durante el desarrollo del sistema.

Tabla 6.5: Resumen de las diferentes pruebas realizadas.

Color	Fondo	Número de Pruebas [%]	1ª Fase		2ª Fase		3ª Fase	
			Validos Color [%]	Validos Dientes [%]	Validos Color [%]	Validos Dientes [%]	Validos Color [%]	Validos Dientes [%]
Aluminio	Rojo	30	15	20	60	80	99	99
Aluminio	Amarillo	30	10	15	50	60	99	99
Aluminio	Negro	30	10	15	50	60	99	99
Aluminio	Azul	30	10	15	50	60	99	99
Aluminio	Verde	30	12	17	55	70	99	99
Aluminio	Blanco	----	----	----	----	----	----	----
Azul	Rojo	30	15	15	50	80	98	98
Azul	Amarillo	30	15	15	50	80	90	90
Azul	Negro	----	----	----	----	----	----	----
Azul	Azul	----	----	----	----	----	----	----
Azul	Verde	30	13	13	40	80	90	90
Azul	Blanco	30	10	10	50	80	95	95
Amarillo	Rojo	30	15	20	60	80	98	98
Amarillo	Amarillo	----	----	----	----	----	----	----
Amarillo	Negro	30	10	10	50	80	95	95
Amarillo	Azul	30	10	10	50	80	95	95
Amarillo	Verde	30	10	10	50	80	95	95
Amarillo	Blanco	----	----	----	----	----	----	----
Rojo	Rojo	----	----	----	----	----	----	----
Rojo	Amarillo	30	15	20	25	25	50	40
Rojo	Negro	30	15	20	25	25	50	40
Rojo	Azul	30	15	20	25	25	50	40
Rojo	Verde	30	15	20	25	25	50	40
Rojo	Blanco	30	15	20	60	80	99	99

El proceso de desarrollo del sistema de software pasó por 3 fases que se hacen referencia en tabla 6.4. Dichas fases son:

1ª Fase: Corresponde a la codificación de los algoritmos iniciales.

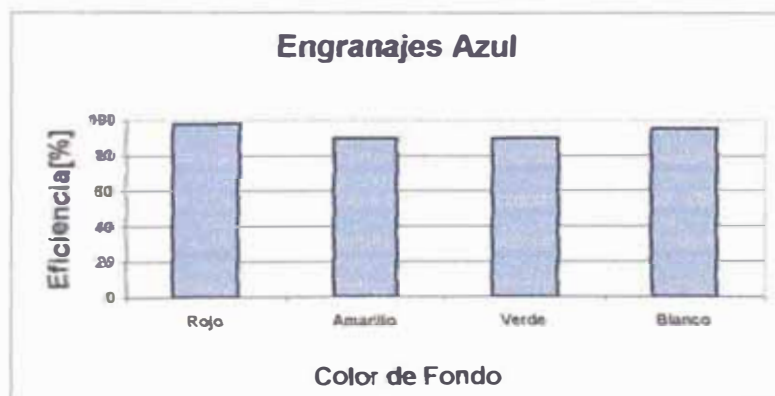
2ª Fase: Corresponde a la modificación de los algoritmos iniciales, en base a los resultados obtenidos en la 1ª fase.

3ª Fase: Son las pruebas que llevaron a la codificación final del sistema. Dichas modificaciones se hicieron considerando los resultados de las fases 1 y 2.

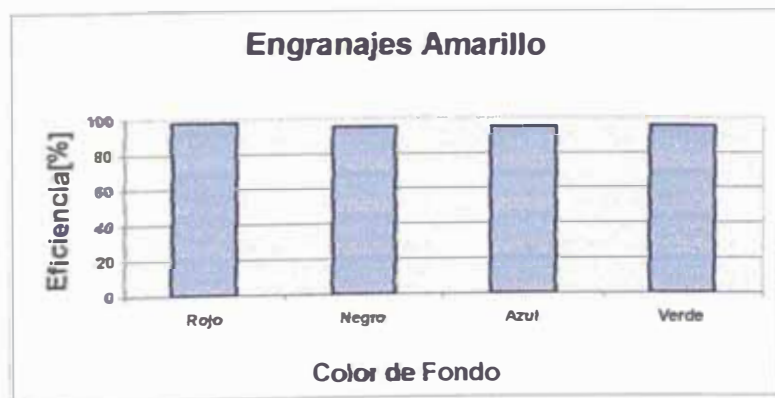
Los resultados de la tabla 6.5 se muestran en forma de grafica en la figura 6.12.



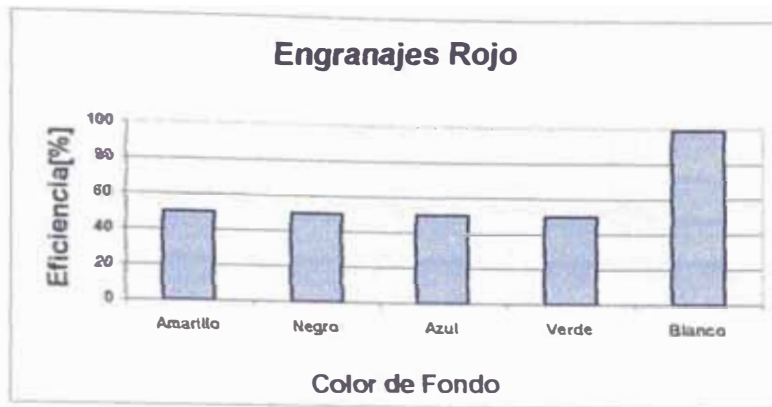
(a)



(b)



(c)



(d)

Figura 6.12: Resultados finales del sistema.

6.12 Observaciones

- La clase CTwain permite la adquisición de imágenes a partir de dispositivos externos de captura de imágenes o desde memoria.
- Existen muchos sistemas comerciales que realizan tareas mediante el Procesamiento Digital de Imágenes y vienen funcionando adecuadamente en diversos campos de la actividad industrial y humana.
- La adquisición de la imagen para su procesamiento se puede realizar de 2 maneras:
 - Con el objeto en reposo. ó
 - Con el objeto en movimiento.

La primera capturar una buena imagen con una cámara estándar, mientras que la segunda exigirá una cámara con mejores características, para minimizar o eliminar el efecto de movimiento.

- Muchas veces donde el objeto de interés es luminoso, la iluminación del sistema debe ser cero, esto para obtener un contraste adecuado.
- Durante el desarrollo de nuestro sistema se trabajo con las imágenes siguientes:

Imagen	Tamaño	Capacidad	Total
Estándar gris	240 x 320 con 256 niveles	240 x 320 = 76800 Bytes	75 KB
Estándar color	240 x 320 con 256 niveles/plano	240 x 320 x 3 = 230400 Bytes	225 KB

- Debido a las mejoras en la calidad de las cámaras digitales y simultáneamente a la rebaja de sus precios, se hace muy conveniente utilizarlas en la adquisición para el Procesamiento Digital de Imágenes.

- **Será importante** elegir la técnica de iluminación adecuada para el proceso de **captura de imagen** de acuerdo a las características que se deseen evaluar del sistema.
- **Para** realizar la separación de objetos dentro de la imagen se debe buscar que el **objeto a separar** tenga un rango grises diferente a su vecindad, de ser así; se lograra su separación utilizando para la umbralización los valores adecuados.
- Un requerimiento inicial en un sistema de control de calidad como en cualquier proceso industrial es el tiempo de respuesta del sistema. Este parámetro es **fundamental** en la etapa de elección de los componentes del sistema.
- Otra característica importante del sistema es determinar la velocidad del objeto a **capturar**. Esto será un limitante del sistema y básicamente definirá las características del dispositivo de captura de imagen (cámara de video ó fotográfica)
- Dependiendo del tipo de sistema de **transporte** que se utilice, si varía el ángulo de **captura de imagen**, entonces también se modificaran los parámetros del objeto. Esto se **podría** resolver de 2 formas: compensando la deformación en el programa, ó eligiendo un sistema de transporte que **no permita** que cambie la posición del objeto dentro de la imagen capturada.

6.13 Recomendaciones

- **Por** ser la visión artificial un campo de investigación y aplicación en pleno crecimiento, se **hace imprescindible** difundir y dedicarle mayor tiempo en nuestro **medio** y en especial dentro de la universidad.
- **En** muchas aplicaciones donde se **requiere** menor tiempo de proceso, se hace imprescindible el uso de tarjetas de procesamiento (procesadores) adicionales a la computadora.
- **La gran mayoría** de dispositivos de adquisición de imágenes a color trabajan con el **formato RGB**. Se debe tener cuidado que de no ser así, realizar las transformaciones respectivas.
- **La calidad** de la imagen adquirida definirá la estrategia de proceso. De acuerdo a **ello se podrá realizar** preprocesado en el dominio del espacio o de la frecuencia. La **segunda** alternativa requerirá mayor tiempo de proceso, pues implica llevar la imagen al dominio de la frecuencia, realizar el preproceso y devolverla al dominio del espacio.

- Se debe tener mucho cuidado de **utilizar** una matriz auxiliar cuando se realiza transformaciones que no sean en el ámbito puntual, pues caso contrario se estará modificando la información bajo análisis.
- En muchos casos es conveniente procesar la imagen en negativo. Esto es recomendable cuando así se logra **disminuir** la cantidad de información a procesar.
- Antes del procesamiento de una imagen se recomienda minimizar el tamaño efectivo de la misma, recortando en forma total o parcial las zonas que no contienen información de interés.
- Para fines prácticos y concretos sería conveniente verificar por medición que la iluminación sea homogénea dentro de la cámara de adquisición.
- Muy importante en cualquier sistema es el buen acabado y presentación. Además es conveniente que a estos sistemas de control de calidad se les agregue seguridad por software, es decir; una clave de acceso (PassWord).

6.14 Conclusiones

- Cualquier proceso o elemento que permita adquirir imágenes por algún medio, llámese; foto, video, rayos-x, satélite, tomografía, mamografía, etc.; podrá ser sometido al análisis por medio del Procesamiento Digital de Imágenes.
- La calidad de la imagen adquirida dependerá directamente de:
 - La iluminación del sistema
 - Las características de la cámara de video o fotográfica
 Y la eficiencia del sistema del control de calidad de:
 - El hardware del sistema
 - Las técnicas de procesamiento utilizadas
 - Los algoritmos respectivos
 - Los drivers y el programa monitor utilizado
- Todos los procedimientos de ingreso de parámetros de referencia para los procesos, tales como nivel de umbral; se podrían detectar en forma automática enseñándole al sistema a que los determine. Pero se debe tener en consideración que esto hará que el sistema se haga mas pesado y su tiempo de respuesta se incremente.
- El MatLab es una herramienta muy útil y practica en la etapa de simulación del proceso. Pero la implementación resulta demasiado lenta, por lo que en esta etapa se debe trabajar en otra plataforma de aplicación tal como el Visual C⁺⁺.

- De acuerdo al desarrollo de las técnicas de Procesamiento Digital de Imágenes y las diferentes aplicaciones que existen hoy en día se puede predecir que a futuro inclusive posibilitaran la supervisión total por medio de visión artificial de procesos completos y además permitirán la visión en personas ciegas.
- El Procesamiento Digital de Imágenes puede ser utilizado en sistemas industriales complejos o en procesos parciales, para tareas precisas y específicas.
- Por lo general en la industria los sistemas de visión artificial son sistemas empotrados que forman parte de un proceso mucho mayor y además trabajan con sistemas adicionales de soporte tales como posicionamiento y/o sujeción de objetos, control de velocidad, control de iluminación, etc.
- Las tareas de preprocesado y la elección de las respectivas técnicas se determinan a partir de la calidad de la imagen obtenida. En muchos casos cuando la adquisición se realiza en forma óptima, se puede obviar el preprocesado.
- Un sistema comercial de control de botellas procesa aproximadamente 72000 botellas/hr., es decir; 20 imágenes/seg. Para implementar un sistema así se debe elegir adecuadamente:
 - El dispositivo de adquisición de imágenes.
 - Los algoritmos.
 - El hardware de soporte.
- El procesamiento de imágenes se puede realizar tanto a nivel de gris ó con imágenes a color. Por lo general se trabaja en nivel de grises debido a que así se maneja una menor cantidad de datos (la tercera parte).
- La calidad de la imagen obtenida, la técnica del procesamiento digital y además los algoritmos; determinaran la eficiencia del sistema.
- Para la determinación de parámetros característicos ó descriptores de un objeto en una imagen, se puede trabajar utilizando como referencias longitudinales el número de píxel.
- En el caso del análisis de engranajes, se busca que la intensidad de luz sea homogénea en la superficie (iluminación difusa), y no se produzca sombra. De esta manera el proceso será más eficiente.
- Por lo general existe más de una forma (algoritmo) de determinar alguna característica en un objeto. La elección de dicho algoritmo se realizara tomando en

cuenta, el tiempo de proceso, simplicidad del algoritmo y respuesta frente a situaciones diferentes (bajo ciertos condicionamientos).

- En el caso de un engranaje rojo, se logra obtener información adecuada usando un fondo que permita obtener el contraste adecuado. En este caso se utilizó el blanco y el aluminio. Para un contraste adecuado en la imagen a procesar, se busca que los colores del engranaje y del fondo sean negativos entre sí.

BIBLIOGRAFIA

- [01] Rafael C. González, Richard E. Woods. “**Tratamiento Digital de Imágenes**”, Addison-Wesley / Díaz de Santos, 1996.
- [02] Javier González Jiménez. “**Visión por Computador**”, Paraninfo, 2000
- [03] Anil K. Jain. “**Fundamentals of Digital Image Processing**”, Prentice-Hall, 1989.
- [04] Wayne Brown; Barry J. Shepherd. “**Graphics File Format. Reference and Guide C**”, Prentice-Hall, 1995.
- [05] Gonzalo Pajares; Jesús M de la Cruz. “**Visión por Computador. Imágenes Digitales y Aplicaciones**”, Ra-Ma, 2001.
- [06] Arturo de la Escalera. “**Visión por Computador. Fundamentos y Métodos**”, Prentice-Hall, 2001.
- [07] Francisco Javier Ceballos. “**Microsoft Visual C++. Aplicaciones para Win32. 2ª Ed.**”, Ra-Ma, 1999.
- [08] K. S. Fu. “**Robótica: Control, Detección, Visión e Inteligencia**”.
- [09] Alonso, M. y Matilla, L. “**Imágenes en acción, análisis y práctica de la expresión audiovisual**”, Madrid: Akal, 1990.
- [10] ilches, L. “**La lectura de la imagen. Prensa, cine, televisión**”, Barcelona: Paidós, 1995.
- [11] Villafañe, J. y Mínguez, N. “**Principios de Teoría General de la Imagen**”, Madrid: Pirámide, 1996.
- [12] **The Graphics File Format Page**

<http://www.dcs.ed.ac.uk/home/mxr/gfx/>

SITIOS WEB

[13] <http://www.maps.com>

[14] <http://www.heuft.com>

[15] <http://www.urp.edu.pe>

[16] <http://www.cs.columbia.edu/botics>

[17] **Correcciones: Rafael C. González. Tratamiento Digital de Imágenes**

http://server.ee.utk.edu/~gonzalez/ipweb/correc_spa.htm
http://server.ee.utk.edu/~gonzalez/ipweb/correc_spa.htm

[18] <http://www.twain.org>

[19] <http://www.sony.com.pe>

[20] <http://www.nikon.com>

[21] <http://www.olympusamerica.com>

[22] <http://www.kodak.com>

[23] <http://www.nikom.com>