



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA MECÁNICA E INDUSTRIAL

SISTEMA MECATRÓNICO PARA LA DETECCIÓN DE ERRORES EN PIELES CURTIDAS

T E S I S

PARA OBTENER EL TÍTULO DE

INGENIERO MECATRÓNICO

P R E S E N T A:

LUIS MIGUEL LEÓN HERNÁNDEZ



DIRECTORA DE TESIS: M.C. MARÍA DEL PILAR CORONA LIRA

CIUDAD UNIVERSITARIA, MÉXICO D.F.

AGOSTO 2012

Agradecimientos

Agradezco a la **vida** por haberme dado la fuerza para lograr una de mis más grandes metas en la vida, le agradezco también por haberme permitido compartir esto con las personas que amo y las cuales me brindaron su apoyo incondicional durante todo este periodo de formación profesional.

A mis **padres**, quienes me permitieron llegar hasta donde estoy, ya que son los pilares de mi vida, gracias por todo su amor y su cariño, y por hacer de mí la persona que soy, sin importar cual adversidad se antepusieron siempre confiando en mí, no tengo palabras para describir lo agradecido que estoy con ustedes, gracias por ser mis padres. Espero poder seguir dándoles satisfacciones y que siempre se sientan orgullosos de mí .

A mi **hermana**, mi gran maestra de la vida, por cada una de las palabras que me brindaste en esos momentos de obscuridad. Siempre serás mi admiración y mi ejemplo a seguir.

A mi fiel compañera **Yoana**: que me enseñó el valor de una simple caricia... gracias por todo tu amor y por hacerme tan feliz; eres mi mejor amiga, mi confidente, mi cómplice, eres lo que siempre soñé, este es el comienzo de una vida llena de alegría, dicha y felicidad. Gracias a tí aprendí que los sueños se pueden hacer realidad, y que no existen imposibles, me enseñaste a creer y a confiar en mí mismo. Siempre tuyo...

A **Susana** que a pesar de sacarme muchas canas verdes me enseñó el significado de la verdadera amistad, los dos maduramos juntos como unos verdaderos hermanos durante la facultad. No tengo palabras para agradecerte todo el apoyo que me brindaste.

A todos mis amigos en especial a **Rafael López** y **Sahel Flores**, gracias por estar a mi lado en los momentos más difíciles de mi vida, gracias por escucharme y siempre darme una palabra de consuelo cuando más la necesitaba, son y siempre serán mis mejores amigos, nunca voy a olvidar los momentos increíbles que pasamos dentro y fuera de la universidad.

A mi directora de tesis **Pili**, gracias por el apoyo brindado en cada etapa de la realización de la tesis, sin su apoyo esto no hubiera sido posible.

A los maestros **Patricio Zamudio** y **Serafín Castañeda** por brindarme el equipo necesario para el desarrollo de esta tesis.

Al maestro **Luis Cesar** que ha sido como un padre para mí. Sus palabras fueron como luces para esos momentos de obscuridad. Mil gracias.

A todos ustedes no me queda más que darles las gracias por ser parte de mis sueños, esta tesis se las dedico con todo mi cariño, respeto y admiración.

Luis León

«En un platillo de la balanza coloco mis odios; en el otro, mis amores. Y he llegado a la conclusión de que las cicatrices enseñan; las caricias, también.»

Mario Benedetti

Índice general

Índice de figuras	I
Índice de Tablas	IV
Glosario	V
Resumen	VI
Objetivo General	VII
Objetivos Particulares	VII
Justificación	VII
Introducción	VIII
Organización de la tesis	IX
1. Marco teórico	1
1.1. Representación digital de imágenes	1
1.1.1. Propiedades de una imagen digital	2
1.2. Etapas fundamentales del procesamiento de imágenes	3
1.2.1. Adquisición de imágenes	3
1.2.2. Pre-procesamiento	4
1.2.3. Segmentación	5
1.2.4. Extracción y selección de características	6
1.2.5. Reconocimiento e interpretación de formas	6
1.2.6. Conocimiento	7
1.3. Antecedentes sobre análisis de texturas	7
1.3.1. Definición de textura	7
1.3.2. Objetivos del estudio de las texturas	7
1.3.3. Campos de aplicación existentes en la actualidad	8
1.3.4. Métodos de caracterización de texturas	8
1.3.4.1. Métodos estadísticos	9
1.3.4.2. Métodos estructurales o geométricos	9
1.3.4.3. Métodos basados en modelos	9
1.3.4.4. Enfoque mediante transformadas	9
1.3.4.5. Comparación de los métodos de caracterización de texturas	10
1.4. Sistema de visión artificial	10
1.4.1. Arquitectura de un sistema de visión	11
1.4.2. Sistema de procesamiento o módulo de proceso	11
1.4.3. Sistema de adquisición de señales de video	12
1.4.4. Sistema de iluminación	12
1.4.5. Sistema de captación	14
1.4.6. Actuadores externos	14
2. Estado de arte	15
2.1. Introducción	15
2.2. Detección de defectos en texturas.	15
2.2.1. Métodos estadísticos	15
2.2.1.1. Propiedades del histograma	16
2.2.1.2. Matriz de co-ocurrencia	16
2.2.1.3. Autocorrelación	17
2.2.1.4. Patrones locales binarios	17
2.2.2. Enfoques estructurales	17

2.2.3.	Enfoques basado en filtros	18
2.2.3.1.	Filtrado en el dominio espacio y frecuencia	18
2.2.3.2.	Métodos espacial-frecuencial.	19
2.2.3.3.	Enfoque basado en modelos	20
2.3.	Enfoques aplicados a la piel curtida	21
3.	Caracterización de texturas	24
3.1.	Introducción	24
3.2.	Detección de defectos de textura por LabVIEW [®]	25
3.2.1.	Descomposición wavelet frame	25
3.2.1.1.	Tipos de <i>wavelet</i>	28
3.2.2.	Extracción de características estadísticas	28
3.2.2.1.	Matrices de co-ocurrencia de niveles de gris (GLCM)	28
3.2.2.2.	Extracción de las características de Haralick	29
3.3.	Clasificador de máquinas de vectores de soporte	30
3.3.1.	Máquinas de vectores de soporte	32
3.3.1.1.	Entrenamiento	33
3.3.1.2.	Clasificación	33
3.3.1.3.	SVM multiclase	33
3.3.1.4.	Modelos	33
3.3.1.5.	<i>Kernels</i>	34
3.3.1.6.	Elección de los parámetros correctos dentro del entrenador clasificador (<i>NI Texture Training Interface</i>)	35
4.	Diseño Mecatrónico	37
4.1.	Especificaciones del sistema	37
4.1.1.	Descripción del proceso	37
4.1.2.	Objeto a examinar	39
4.1.3.	Adquisición de imágenes	40
4.1.3.1.	Cámara y óptica	41
4.1.3.2.	Hardware IMAQ de adquisición de imágenes	44
4.1.4.	Iluminación y acondicionamiento de la imagen	45
4.1.4.1.	Sistema de iluminación implementado	46
4.1.5.	Manipulador cartesiano	47
4.1.5.1.	Motores	48
4.2.	Hardware de control	50
4.2.1.	Tipos de sistema de control	50
4.2.1.1.	Computadora	50
4.2.2.	Microcontrolador	51
4.2.2.1.	Características básicas	52
4.2.3.	Etapas de potencia	53
4.2.4.	Sensores	56
4.3.	Software incorporado	57
4.3.1.	Diagrama de bloques	58
4.3.1.1.	Lógica de programación	59
4.3.1.2.	Configuración de adquisición y envío de señales.	61
4.3.1.3.	Configuración de adquisición de imagen	63
4.3.1.4.	Configuración de la etapa de procesamiento	65
4.3.2.	Panel frontal	66
4.3.3.	Programación del controlador	67
5.	Resultados experimentales	71
5.1.	Entrenamiento del archivo clasificador de texturas	71
5.2.	Prueba de repetibilidad	77
5.3.	Resultados obtenidos	80

6. Conclusiones y trabajo a futuro	87
6.1. Conclusiones	87
6.2. Principales aportaciones	88
6.3. Lineas de trabajo a futuro	89
Referencias	89
A. Piel curtidas analizadas	95
A.1. Texturas muestras en buen estado	95
A.2. Piel No 1	96
A.3. Piel No 2	97
A.4. Piel No 3	98
A.5. Piel No 4	99
A.6. Piel No 5	100
A.7. Piel No 6	101
B. Hoja de datos	102
C. Caracterización del motor	106
D. Cálculo de repetibilidad	108

Índice de figuras

1.1. Convención de ejes utilizado para la representación digital de imágenes. *Elaboración propia (Julio-2012)	1
1.2. Representación y composición de una imagen digital. *Elaboración propia (Julio-2012)	2
1.3. Etapas fundamentales del procesamiento digital de imágenes. *Tomado de Haralick [1]	3
1.4. Digitalización de la imagen. Posterior a la captura de la imagen, esta se discretiza, mediante una forma de matriz numérica, la cual puede ser procesada por una computadora. *Elaboración propia (Julio-2012)	4
1.5. Fuentes de error en una imagen digital. a) Imagen Original. b) Imagen corrompida con ruido Gaussiano. c) Ruido Sal y Pimienta. *Tomado de http://www.slideshare.net/omarspp/imagen-filtrado-espacial	5
1.6. Elementos básicos de un sistema de Visión. *Elaboración propia (Julio-2012) . . .	12
1.7. De izquierda a derecha: a) luz radial Directa b) luz radial difusa c) proyector de luz directa d) proyector de luz difusa. *Tomado de http://www.dcmsistemas.com/index.php?sec=doc&sbsec=teo&pag=13	14
2.1. Ejemplo de defectos en diferentes tipos de superficies - De la izquierda a derecha: Acero [2], Piedra [3], textil [4], Madera [5] y Azulejo Cerámico [6].	16
2.2. Localización de defectos en madera. *Tomado de Niskanen et al.[7, 8]	17
2.3. Detección de grietas en azulejos de granito. *Tomado de Kittler et al.[9]	18
2.4. Detección de defectos en textiles. *Tomado de Tsai y Hsieh[10]	19
2.5. Localización de defectos en texturas aleatorias. *Tomado de Xie y Mirmehdi[11] . .	20
2.6. Detección de defectos en piel. *Tomado de Amorim et. al.[12]	21
2.7. Muestras de piel curtida y los resultados de inspección. *Tomado de Sobral[13] . .	22
2.8. Detección de defectos en piel <i>Wet Blue</i> . *Tomado de Villar et al.[14]	23
3.1. Entrada típica y sus imágenes de salida después de ser analizadas. *Tomado de <i>NI Vision Concepts Help</i> [15]	25
3.2. Esquema de la transformada de wavelet en dos dimensiones. *Tomado de <i>NI Vision Concepts Help</i> [15]	26
3.3. Esquema de la transformada discreta de <i>wavelet frame</i> de 2 dimensiones. *Tomado de <i>NI Vision Concepts Help</i>	27
3.4. La siguiente figura ilustra las imágenes derivadas de las subbandas de dos niveles de descomposición. *Elaboración propia (Julio-2012)	27
3.5. A) Textura muestra. B) Valores de los pixeles para la textura muestra. C) GLCM para el vector de desplazamiento. *Tomado de <i>NI Vision Concepts Help</i> [15]	29
3.6. Ejemplo de las características de Haralick para la sub-banda <i>low-high</i> . *Elaboración propia (Julio-2012)	30
3.7. Ejemplo de un SVM	31
3.8. Se muestra una aplicación que involucra dos clases linealmente separables representadas en un espacio bidimensional. A) Muestras de clase 1 B) Muestras de clase 2 C) Vectores de soporte D) Hiperplano E) Margen. *Elaboración propia (Julio-2012)	32

3.9.	<i>La Figura A) ilustra como un kernel polinomial separa información de características no lineales. La Figura B) ilustra como un kernel RBF separa la información de características no lineales. La Figura C) muestra la información de características no lineales claramente divisible obtenido después de usar un kernel no lineal para transformar la muestra a una una dimensión donde la información de características son linealmente separables. *Elaboración propia</i>	35
4.1.	División del espacio de trabajo del manipulador. *Elaboración propia (Julio-2012)	38
4.2.	Recorrido de la cámara en el espacio de trabajo del manipulador. *Elaboración propia (Julio-2012)	38
4.3.	Porción de piel curtida de bovino a analizar. *Elaboración propia (Julio-2012) . . .	39
4.5.	Sistema de adquisición de imágenes. *Elaboración propia (Julio-2012)	40
4.4.	Diferentes tipos de defectos en la piel curtida de bovino. *Elaboración propia (Julio-2012)	41
4.6.	Cámara CCD JAI CV-M50 E. *Tomado de http://www.pixequip.com/product_images/18550_jai_cv-m50_01.png	42
4.7.	Porción de piel curtida de bovino a examinar. *Elaboración propia (Julio-2012) . .	44
4.8.	Unión entre hardware y software. *Elaboración propia (Julio-2012)	45
4.9.	A) Estructura de la cámara de aislamiento. B) Vista de la estructura. C) Manipulador completamente aislado de luz ambiental.	46
4.10.	A) y B) iluminación implementada. C) Lampara Led GU10 con 60 Leds	47
4.11.	Manipulador cartesiano con 3 grados de libertad.	47
4.12.	Sistema de sensado para establecer el <i>home</i> del sistema.	48
4.13.	Motor Minebea 23KM-K.	49
4.14.	<i>Driver</i> para el motor y terminales de conexión. Véase Hoja de datos	49
4.15.	Diagrama de bloques del hardware de control. *Elaboración propia (Julio-2012) . .	51
4.16.	PXI y modulo de adquisición de datos con su correspondiente bloque conector. . .	52
4.17.	Microcontrolador utilizado.	52
4.18.	Diagrama de flujo del controlador. *Elaboración propia (Julio-2012)	53
4.19.	Diagrama de flujo de la etapa de potencia. *Elaboración propia (Julio-2012)	54
4.20.	Conexión entre la tarjeta de adquisición y el microcontrolador arduino.	55
4.21.	Esquemático del hardware para el controlador de motor a pasos.	55
4.22.	Diagrama de la etapa de potencia. *Elaboración propia (Julio-2012)	56
4.23.	Diagrama del Hardware total. *Elaboración propia (Julio-2012)	56
4.24.	Esquemático del sensor de herradura. *Tomado de Apéndice C	56
4.25.	Diagrama de conexión del sistema de sensado para el <i>home</i>	57
4.26.	Diagrama de conexión del sistema de sensado de emergencia.	57
4.27.	Ejes y motores del manipulador.	58
4.28.	Diagrama de máquina de estados.	61
4.29.	Señales de control.	61
4.30.	Envío de señales digitales al microcontrolador.	62
4.31.	A) Lectura proveniente del Arduino B) Lectura de sensores de <i>home</i>	63
4.32.	A) En el cuadro rojo se puede observar la parte donde se lleva a cabo la inicialización, asignación de memoria y adquisición de la imagen. B) En el cuadro amarillo se encuentra la parte del programa donde se cierra los recursos utilizados de hardware y libera espacio en memoria.	64
4.33.	Etapa de procesamiento y detección de defectos en pieles curtidas.	66
4.34.	Panel frontal del proyecto.	67
4.35.	Diagrama de señales de entrada y salida del arduino.	67
4.36.	Diagrama de estados para la parte manual del programa.	68
4.37.	Diagrama de estados para la parte automática del programa.	69
5.1.	Muestra 1 con defecto para el entrenamiento del <i>NI training texture</i>	72
5.2.	Sub-bandas del <i>wavelet</i> tipo Haar para la muestra 1. A) <i>Low-Low</i> . B) <i>Low-High</i> . C) <i>High-Low</i> . D) <i>High-High</i> . E) <i>Low-Low-Low</i> F) <i>Low-Low-High</i> G) <i>Low-High-Low</i> . H) <i>Low-High-High</i>	73
5.3.	Características de Haralick para la muestra 1. A) Contraste. B) Correlación. C) Disimilitud. D) Entropía. E) Homogeneidad.	73
5.4.	Muestra 2 con defecto para el entrenamiento del <i>NI training texture</i>	74

5.5. Sub-bandas del wavelet tipo Haar para la muestra 2. A) <i>Low-Low</i> . B) <i>Low-High</i> . C) <i>High-Low</i> . D) <i>High-High</i> . E) <i>Low-Low-Low</i> F) <i>Low-Low-High</i> G) <i>Low-High-Low</i> . H) <i>Low-High-High</i>	74
5.6. Características de Haralick para la muestra 2. A) Contraste. B) Correlación. C) Disimilitud. D) Entropía. E) Homogeneidad.	75
5.7. Sub-bandas. A) <i>Low-Low</i> . B) <i>Low-High</i> . C) <i>High-Low</i> . D) <i>High-High</i> . E) <i>Low-Low-Low</i> F) <i>Low-Low-High</i> G) <i>Low-High-Low</i> . H) <i>Low-High-High</i>	75
5.8. Tercera muestra con defecto para el entrenamiento del <i>NI training texture</i>	76
5.9. Características de Haralick. A) Contraste. B) Correlación. C) Disimilitud. D) Entropía. E) Homogeneidad	76
5.10. Texturas muestra sin defecto.	77
5.11. Imágenes binarias de salida del clasificador <i>NI training interface</i>	77
5.12. Posiciones seleccionada para la prueba de la repetibilidad. A) Posición 1 B) Posición 2 C) Posición 3 D) Posición 4.	78
5.13. Resultado repetibilidad Posición 1.	78
5.14. Resultado repetibilidad Posición 2.	79
5.15. Resultado repetibilidad Posición 3.	79
5.16. Resultado repetibilidad Posición 4.	80
5.17. Defecto encontrado al analizar la Piel curtida No1.	80
5.18. Defecto encontrado al analizar la Piel curtida No2.	81
5.19. Defectos encontrado al analizar la Piel curtida No3.	81
5.20. Defectos encontrados al analizar la Piel curtida No4.	81
5.21. Defectos encontrados al analizar la Piel curtida No5.	81
5.22. Defectos encontrados al analizar la Piel curtida No6.	82
5.23. Piel curtida No 1 antes y después de ser analizada.	82
5.24. Panel Frontal después de terminar el análisis de la Piel curtida No1.	82
5.25. Piel curtida No2 después de ser analizada.	83
5.26. Panel Frontal después de terminar el análisis de la Piel curtida No2.	83
5.27. Piel curtida No3 después de ser analizada.	83
5.28. Panel Frontal después de terminar el análisis de la Piel curtida No3.	84
5.29. Piel curtida No4 antes y después de ser analizada.	84
5.30. Panel Frontal después de terminar el análisis de la Piel curtida No4.	84
5.31. Piel curtida No5 antes y después de ser analizada.	85
5.32. Panel Frontal después de terminar el análisis de la Piel curtida No5.	85
5.33. Piel curtida No6 antes y después de ser analizada..	85
5.34. Panel Frontal después de terminar el análisis de la Piel curtida No6.	86
5.35. Defecto proveniente de la piel curtida No5 que no fue detectado por el algoritmo.	86
5.36. Características de Haralick. A) Contraste. B) Correlación. C) Disimilitud. D) Entropía. E) Homogeneidad.	86
C.1. Fragmento de la cuadrícula diseñada para el cálculo del número de pasos.	106
C.2. Definición del número de secuencias para el avance de los motores.	107

Índice de Tablas

1.1. Estudio comparativo de los métodos de caracterización de texturas. *Elaboración propia (Julio-2012)	10
3.1. Tipos de <i>wavelets</i> . *Tomado de <i>NI Vision Concepts Help</i> [15]	28
4.1. Siete tipos de defectos en pieles curtidas de bovino. Tomado de Yeh y Perng [16]. .	40
4.2. Especificaciones de la cámara CCD. Véase Apéndice B	42
4.3. Requerimientos de dominio para la inspección de defectos en piles curtidas de bovino.	43
4.4. Secuencia normal de pasos para motores unipolares.	49
4.5. Especificaciones básicas del Arduino mega.	53
4.6. Señales digitales enviadas al arduino.	62
4.7. Señales digitales recibidas.	63
4.8. Tabla de verdad del modo manual.	68
4.9. Tabla de los estados con sus correspondientes salidas y lecturas.	68
4.10. Tabla de verdad del modo automático.	69
4.11. Tabla de los estados con sus correspondientes salidas y lecturas del modo automático.	70

Glosario

Black-propagation	Entrenamiento hacia atrás
CCD	Charge coupled devices o dispositivos de acoplamiento de carga
CCR	Correct classification ratio
DWFT	Discrete Wavelet Frame Transform (Transformada discreta de wavelet frame)
FNN	Red neuronal con propagación hacia delante del inglés (Feed-forward Neural Network)
LBP	Patrones binarios locales (del inglés Local Binary Pattern)
MCNG	Matriz de Co-ocurrencia de Niveles de Gris
MRF	Markov Random Field
SVM	Máquinas de vectores de soporte (Support Vector Machine, por sus siglas en inglés SVM)

Resumen

Este proyecto plantea el desarrollo de un sistema funcional en tiempo real, para la detección de defectos en pieles curtidas de bovino. El cual permite la inspección de determinados tipos de defectos, conociendo que este tipo de textura no tiene una forma periódica o regular.

El sistema se integra por el software LabVIEW[®], el cual mediante un chasis PXI con un controlador embebido NI PXI-8195 y dos módulos periféricos de adquisición (uno de E/S analógica-digital y otro de adquisición de imagen), realizan la comunicación entre las etapas de adquisición y actuación del sistema. La etapa de adquisición esta constituida por una cámara monocromática CCD marca JAI[®] y la etapa de actuación se realiza utilizando un manipulador cartesiano de tres grados de libertad. Tanto la adquisición como el procesamiento digital de imágenes se desarrollan mediante los Módulos *NI Vision Development*[®] y *IMAQ Detect Texture Defect*[®].

El espacio de trabajo del manipulador fue dividido en espacios más pequeños con el objetivo de obtener imágenes más detalladas, la adquisición de cada una de ellas se desarrolla mediante la ejecución de un evento definido que al ser tomada es procesada señalando si la porción analizada tiene o no tiene defectos, esta imagen es guardada en un archivo jpg y eliminada de la memoria para ayudar a optimizar recursos en el controlador embebido, al finalizar las tomas del recorrido de todas las porciones, éstas son unidas para formar una imagen general de la piel resaltando los defectos hallados en dicho recorrido. En el procesamiento se hace uso de imágenes en escala de grises de 8 bits.

Para el análisis de defectos se utiliza un algoritmo de visión por computadora implementado por LabVIEW[®] el cual extrae las características de estadística local de la descomposición de imágenes en escala de grises de la textura con *wavelet frames* en sub-bandas de varias orientaciones y escalas. Las características locales extraídas son derivados de estadística de segundo orden de la Matriz de Co-ocurrencia de Niveles de Grises. Posteriormente, una máquina clasificadora de vectores de soporte (SVM) es entrenada para aprender la descripción general de una textura normal en muestras sin defectos [15]. Este algoritmo es implementado en LabVIEW[®] y es capaz de procesar texturas naturales en tiempo real.

El método utilizado combina la transformada de *wavelet* con los descriptores estadísticos. Estos descriptores estadísticos son sencillos y permiten el análisis de texturas heterogéneas. Además, se combina con la transformada de *wavelet* que permite trabajar a varias escalas (con texturas de distintos tamaños) realizando un análisis multiresolución y dando una visión más cercana a la proporcionada por el ojo humano ya que separa los detalles de la textura según su orientación.

Introducción

Objetivo general

Realizar un sistema Mecatrónico que detecte defectos en las pieles curtidas de vaca a través de visión artificial, aportando así nuevas tecnologías para que este proceso que se hace por inspección humana ahora se realice, de una forma automatizada.

Objetivos particulares

- Generar el algoritmo de adquisición y procesamiento digital de imágenes mediante el uso del lenguaje gráfico LabVIEW[®].
- Seleccionar las técnicas de visión adecuadas para la detección de defectos en pieles curtidas.
- Determinar una iluminación adecuada y controlada.
- Construir una interfaz de comunicación entre el usuario y el manipulador.
- Desarrollar un prototipo funcional que se puede realizar e implementar.
- Entrega y despliegue de información en pantalla de partes con defectos encontrados en la piel curtida de bovino.

Justificación

La inspección visual y clasificación de defectos superficiales en pieles curtidas son muy importantes en la fabricación de productos de cuero donde se requiere usualmente alta calidad. Esas operaciones se realizan actualmente por inspectores humanos que suelen dejar pasar un número considerable de defectos ya que los seres humanos son inconsistentes e inapropiados para esas tareas sencillas y repetitivas. Además, dado que la inspección y clasificación visual son tareas lentas y requieren mucho trabajo, esto puede convertirse en un cuello de botella crítico en el proceso de producción. La inspección automática puede reducir las cargas de trabajo humano [17].

De hecho, mucha investigación se ha llevado a cabo en la inspección automática de superficies como metal, madera y textiles, mientras que se ha hecho poco trabajo en la clasificación automatizada de defectos en texturas naturales, debido a la dificultad de la naturaleza del problema. Es prácticamente imposible construir modelos exactos de defectos para la clasificación ya que su aspecto y tamaño varían mucho y es difícil encontrar dos defectos con la misma forma y tamaño, incluso si pertenecen a la misma clase de defecto. No obstante, este proceso de clasificación es necesario debido a que juega un papel importante en el suministro de información para la prevención de defectos.

El desarrollo de sistemas para la detección de defectos en pieles curtidas, es un problema no resuelto en cuanto a su aplicación general, ya que sólo existen muy poca literatura científica que tratan de solucionar este tipo de problema. El presente proyecto, surge de la necesidad de desarrollar un sistema para llevar a cabo una inspección visual. Mediante la implementación del sistema de inspección automático, se pretende primeramente, evitar problemas de salud en los operarios humanos; tales como: estrés, cansancio visual y fatiga originados por la realización de una tarea repetitiva, repercutiendo en la inspección del producto.

Introducción

La automatización de muchos de los procesos de fabricación, como el ensamblaje mecánico de piezas, el transporte de mercancías, o la manipulación de productos peligrosos, entre otros, ha constituido un gran avance en la industria. Sin embargo en la mayoría de los casos, el proceso de inspección visual de productos sigue siendo realizado hoy en día por numeroso personal capacitado.

Una de las mayores dificultades en la inspección de un producto, es la detección de defectos, realizada en la mayoría de las ocasiones por inspectores humanos. Para cualquier inspector calificado, la detección de defectos resultan labores sencillas y rutinarias. Por lo general, la complejidad de estos trabajos residen en conseguir un alto nivel de atención visual de manera prolongada y proceder objetivamente a la hora de discernir sobre la calidad de los productos.

Aun cuando es alta la necesidad de precisión y exactitud, las personas tienen un periodo limitado de atención, lo cual las hace susceptibles a distraerse. Además, presentan ciertas inconsistencias en su sensibilidad visual en el transcurso de un día a otro. Por el contrario los sistemas de inspección automatizados, pueden mantener de manera incansable toda su atención sobre la tarea que deben desarrollar y reproducir siempre, de manera objetiva y constante los criterios de selección o catalogación para los que han sido entrenados.

Hay un sin número de aplicaciones de visión artificial en la industria manufacturera. Estas técnicas prometen un gran avance económico y buenos niveles de precisión sin la presencia humana; por lo que es bastante obvio suponer que en los próximos años la mayoría de las técnicas de inspección de calidad para aplicaciones industriales, se llevaran a cabo por medio de sistema de visión por computadora.

Los sistemas de inspección visual para superficies texturizadas han sido utilizado en diferentes áreas para el análisis de defectos, como son entre otros textiles, madera y pintura de automóvil. Las pieles curtidas como un material natural poseen características como apariencia visual no homogénea en color, grosor, brillo y arrugas, lo cual lo hace un objeto complejo para el control y análisis. Los defectos superficiales en las pieles curtidas afectan el aspecto estético de los artículos de cuero y la cantidad de superficie usable, porque la existencia de áreas con defectos puede ser considerado como no utilizable o útil para fines particulares.

Realmente esta aplicación particular de visión por computadora es bastante difícil; esto es debido principalmente a las características particulares de algunos defectos que para ser vistos requieren un estiramiento mecánico en la piel; obviamente con este tipo de defectos el riesgo concreto es que solo se hacen evidentes en el producto terminado. Además, el color de los cueros curtidos pueden variar de blanco a negro y esto plantea problemas en la elección correcta de la fuente de luz para la iluminación correcta de la escena.

Por otra parte, a pesar de que los sistemas de inspección manuales han sido remplazados con éxito por inspecciones automáticas, en muchos sectores es necesaria una mejora en la precisión de los procesos para reducir la tasa de falsos positivos¹ y falsos negativos² así como la reducción del tiempo de procesamiento.

El siguiente trabajo presenta un intento de abordar los problemas implicados en la detección de imperfecciones de las pieles curtidas utilizadas en diversos segmentos de la industria. En la literatura científica solo hay un número limitado de obras que tratan de solucionar este tipo de problemas y es aquí donde se encuentra la dificultad de este trabajo.

¹Productos clasificados como buenos cuando son defectuosos.

²Productos clasificados como defectos cuando ellos son buenos.

Organización de la tesis

El presente trabajo de tesis se encuentra dividido en los siguientes capítulos:

- Capítulo 1 titulado "Marco teórico" se describe de manera general los conceptos básicos referidos al procesamiento digital de imágenes así como antecedentes sobre análisis de texturas. Se hace un breve resumen sobre los métodos utilizados para el análisis de texturas. Por último se describe de manera detallada los elementos que conforman un sistema de visión artificial.
- Capítulo 2, "Estado de arte" se examinan los avances recientes en la inspección de superficies donde se utilizan técnicas de visión por computadora y procesamiento digital, especialmente aquellos métodos basados en análisis de texturas aleatorias. También se revisan las técnicas aplicadas exclusivamente a las pieles curtidas de bovino. Finalmente se comparan los enfoques de clasificación con la detección novedosa en la etapa de toma de decisiones.
- Capítulo 3, "Caracterización de texturas" se describe detalladamente el algoritmo utilizado para la detección de defectos en pieles curtidas de bovino, así como sus principales parámetros con los cuales se configurará.
- Capítulo 4, "Diseño mecatrónico" se especifican los dispositivos utilizados y sus principales características para la realización de esta tesis. Se describe paso a paso el diseño y la implementación del hardware y software del sistema para el funcionamiento del proyecto.
- Capítulo 5, "Resultados experimentales" se presentan los resultados experimentales que se han considerado de más valor académico para ilustrar el funcionamiento del algoritmo propuesto para este proyecto de tesis.
- Capítulo 6, "Conclusiones y trabajo a futuro" se da a conocer las conclusiones del trabajo y se proponen algunas direcciones futuras de investigación.

Capítulo 1

Marco teórico

El procesamiento digital de imágenes es un campo de investigación que sigue en un constante crecimiento. El progreso en esta área no ha sido por si sola, sino en conjunto con otras áreas con las cuales esta relacionada como las matemáticas, la computación, el conocimiento de órganos del cuerpo que intervienen en la percepción, entre otras.

El interés en el procesamiento digital de imágenes proviene de dos áreas principales de aplicación: La mejora de la información en la imagen para la interpretación humana y el procesamiento de los datos obtenidos de la imagen para la percepción autónoma por una máquina.

Los problemas típicos de la percepción automatizada que utilizan rutinariamente técnicas de procesamiento de imágenes son el reconocimiento automático de caracteres, la visión industrial mecanizada para el ensamblado, inspección de productos, reconocimientos militares, tratamiento automático de huellas digitales, muestras de sangre, imágenes de rayos x, procesamiento automático de imágenes aéreas y de satélite para la predicción del tiempo y evaluación de cultivos [18].

1.1. Representación digital de imágenes

El término imagen se refiere a una función bidimensional $f(x, y)$ que representa la intensidad de luz, donde x y y denotan las coordenadas espaciales y el valor de f en cualquier punto (x, y) es proporcional al brillo (o nivel de gris) de la imagen en ese punto [19] (Figura 1.1).



Figura 1.1: Convención de ejes utilizado para la representación digital de imágenes. *Elaboración propia (Julio-2012)

Por convención, la referencia espacial del pixel con coordenadas $(0,0)$ se encuentran en la esquina superior izquierda de la imagen. Esta imagen puede considerarse como una matriz cuyos

índices de fila y columna identifican un punto de la imagen, y el valor correspondiente de la matriz indica el nivel de gris en ese punto. Nótese en la Figura (1.2) que los valores de x incrementan moviéndose de izquierda a derecha, y los valores de y incrementan de arriba a abajo.

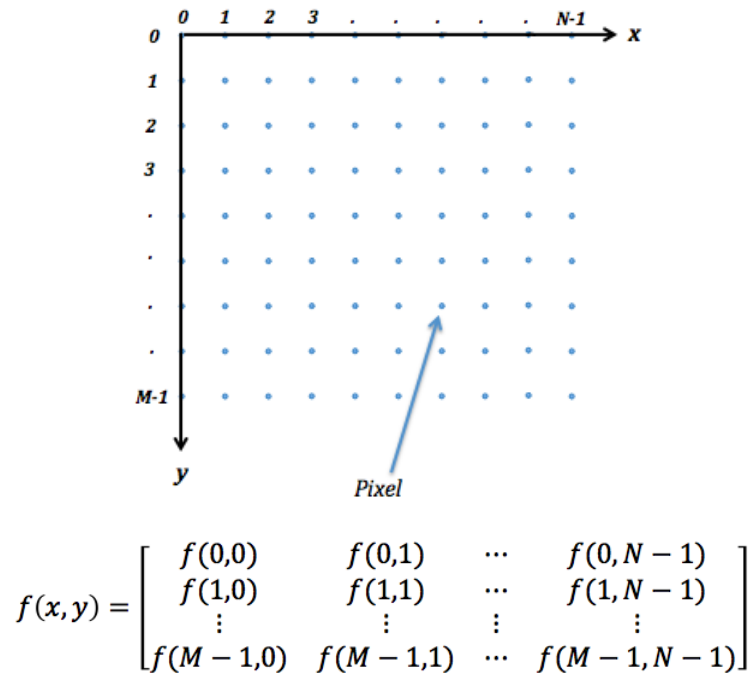


Figura 1.2: Representación y composición de una imagen digital. *Elaboración propia (Julio-2012)

1.1.1. Propiedades de una imagen digital

Las imágenes digitales, tienen tres propiedades básicas: resolución, definición y número de planos [15].

- Resolución: La resolución de una imagen es el número de filas y columnas que forman los píxeles. Una imagen compuesta por M filas y N columnas tiene una resolución $M \times N$. Esta imagen tiene m píxeles a lo largo del eje horizontal y n píxeles a lo largo del eje vertical.
- Definición: La definición de una imagen, también llamada profundidad de pixel, indica el número de colores que se pueden ver en la imagen. La profundidad de modulación es el número de bits usado para codificar la intensidad del pixel.
- Número de planos: Es el número de arreglos de píxeles que componen la imagen. Una imagen en escala de grises está compuesta por un solo plano, mientras que una imagen de color verdadero está formada por tres planos. (Rojo, Verde, Azul).

De acuerdo a la definición de profundidad de pixel, para trabajar con números en la computadora, el nivel de brillo, o valor de cada pixel, es cuantizado¹ a códigos binarios enteros positivos². El número de niveles de cuantización está determinado por la relación:

$$L = 2^n \tag{1.1}$$

Donde n representa el número de bits necesarios para cada muestra. Esto es, que con 5 bits de cuantización en la intensidad se pueden representar 32 niveles de grises ($2^5 = 32$). Para tener una imagen de buena calidad para el ojo humano es necesario tener como mínimo 64 niveles de gris [19]. Para una apreciación fina se usa una cuantización estándar a 8 bits³, esto es, 256 niveles de

¹Conversión que sufre la amplitud de la señal analógica (imagen natural, capturada con una cámara, telescopio, etc) generándose el concepto de nivel de gris o intensidad.

²El brillo no puede ser negativo.

³La cuantización a 8 bits (1 byte) se dice estándar porque es el mínimo número direccionable directamente por la mayoría de los microprocesadores.

gris en un rango de $[0, 255]$, en donde a la escala de gris 0 es asignada al color negro y la escala de gris 255 al color blanco. A una imagen cuantizada de esta manera se le conoce como imagen en escala de grises.

Una imagen binaria es una imagen monocromática cuantizada a 1 bit por pixel, esto es, dos niveles, blanco y negro. Para el caso de imágenes a color, suelen usarse 256 niveles para representar la intensidad de color de cada uno de los tres colores primarios RGB (rojo, verde y azul). De esta forma se obtienen 16 millones de colores aproximadamente ($256 \times 256 \times 256$) y se habla de imágenes en color real. En algunos casos puede necesitarse mayor resolución radiométrica y se usan 4096 niveles por banda de color en vez de 256 o incluso más.

Se ha de tener en cuenta, que si el muestreo consiste en un mallado de M por N cuadrados y el número de niveles de gris permitido $L = 2^n$, entonces el número de bits totales necesarios para almacenar una imagen digitalizada es:

$$M \times N \times (\text{No. Bits pixel}) = 2^n \quad (1.2)$$

1.2. Etapas fundamentales del procesamiento de imágenes

Gonzalez y Woods [18] modelan los pasos fundamentales que se siguen en el procesamiento de imágenes (Figura 1.3).

Aunque cada aplicación de visión por computadora, tiene sus propias especificaciones para realizar una tarea en específico, se puede decir que existe un tronco común de etapas entre ellas, mismas que están agrupadas en 3 tipos de niveles: visión de bajo nivel, visión intermedia y visión de alto nivel.

Los niveles y etapas que conforman el procesamiento de imágenes, tienen como objetivo mejorar la calidad de las imágenes, con la finalidad de realizar una interpretación y por consiguiente obtener un resultado. Cabe mencionar que dichos niveles y etapas varían de acuerdo al entorno donde se pretenda implementar el sistema de visión; dichos entornos se dividen en: industriales y científicos. Destacando que en aplicaciones industriales, la velocidad a la que se realizan las operaciones es fundamental ya que se deben evaluar todos los productos o piezas producidas en tiempo real, en tanto que las aplicaciones científicas, se busca la determinación de los resultados en imágenes más complejas.

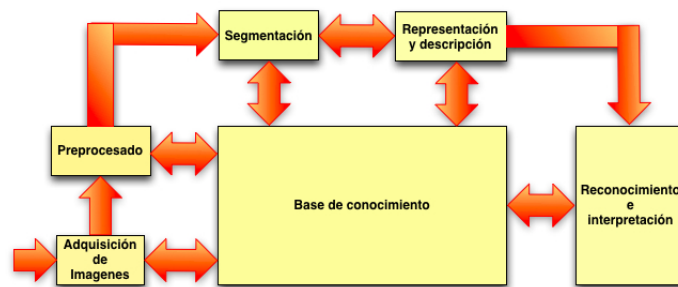


Figura 1.3: Etapas fundamentales del procesamiento digital de imágenes. *Tomado de Haralick [1]

1.2.1. Adquisición de imágenes

El primer paso del proceso de un sistema de visión, es la adquisición de la imagen, esto es, digitalizarla.

En esta etapa se construye el sistema de información de las imágenes, basado en la captura o adquisición de la imagen. Su objetivo es realzar, mediante técnicas fotográficas (iluminación, óptica,

cámaras, filtros, pantallas, etc.), las características visuales de los objetos (formas, texturas, colores, sombras, etc) Figura (1.4).

La imagen debe obtenerse utilizando alguna fuente de energía que en interacción con el objeto de análisis, brinde información sobre su estado. Dependiendo del tipo de análisis que se desee hacer sobre el elemento en estudio, se deberá seleccionar la energía necesaria para poder tomar una imagen de él. Los tipos de energía utilizados son: luz (visible) para la fotografía; rayos X y rayos γ para radiografías y tomografías, ultrasonido para ecografías, campos magnéticos para resonancias magnéticas, calor para termografías, etc.

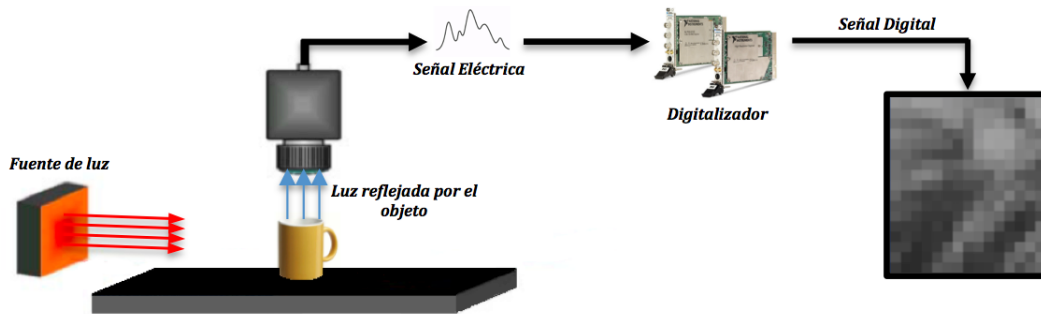


Figura 1.4: Digitalización de la imagen. Posterior a la captura de la imagen, esta se discretiza, mediante una forma de matriz numérica, la cual puede ser procesada por una computadora. *Elaboración propia (Julio-2012)

1.2.2. Pre-procesamiento

Posterior a la obtención de una imagen digital, el siguiente paso es el pre-procesamiento. Su objetivo es mejorar la calidad informativa de la imagen adquirida, y transformarla para conseguir una manera más adecuada de interpretar la información que en ella se refleja; de manera que se incremente la oportunidad de éxito de los siguientes procesos. Cuando se adquiere una imagen mediante cualquier sistema de captura, por lo general esta no es directamente utilizable por el sistema de visión. La aparición de variaciones en intensidad debidas a ruido⁴, o deficiencias en la iluminación, o a la obtención de imágenes de bajo contraste, hace necesario un pre-procesamiento de la imagen con el objetivo fundamental de corregir estos problemas, además de aplicar aquellas transformaciones a la imagen que acentúen las características que se deseen extraer de las mismas, de manera que se faciliten las operaciones de las etapas posteriores.

Como técnicas de pre-procesamiento, resaltan dos tipos: las que buscan eliminar características no deseadas (ruido) (Figura 1.5) y las que examinan resaltar determinadas características de la imagen.

Algunas de las técnicas de pre-procesamiento más habituales son:

- Conversión del color o de los niveles de gris: su objetivo es la mejora de la calidad de la imagen.
- Transformaciones geométricas: no modifican la información, solo el aspecto visual. El objetivo fundamental de una operación geométrica, es transformar los valores de una imagen, tal y como podría observarse desde otro punto de vista. Algunas de las transformaciones son: rotar, trasladar y amplificar.
- Transformación del histograma: Las transformaciones del histograma pueden facilitar la segmentación de objetos de la imagen, aunque habitualmente sólo sirve para modificar el contraste de la imagen y el rango dinámico de los niveles de gris.

⁴El proceso de adquisición y digitalización de una imagen conlleva a introducir datos adicionales, conocidos como ruido. El ruido por sí mismo no puede considerarse información útil de la imagen, puesto que se trata de datos que aparecen y no deberían estar ahí; influyendo en los posteriores procesamientos que se hagan a la imagen.

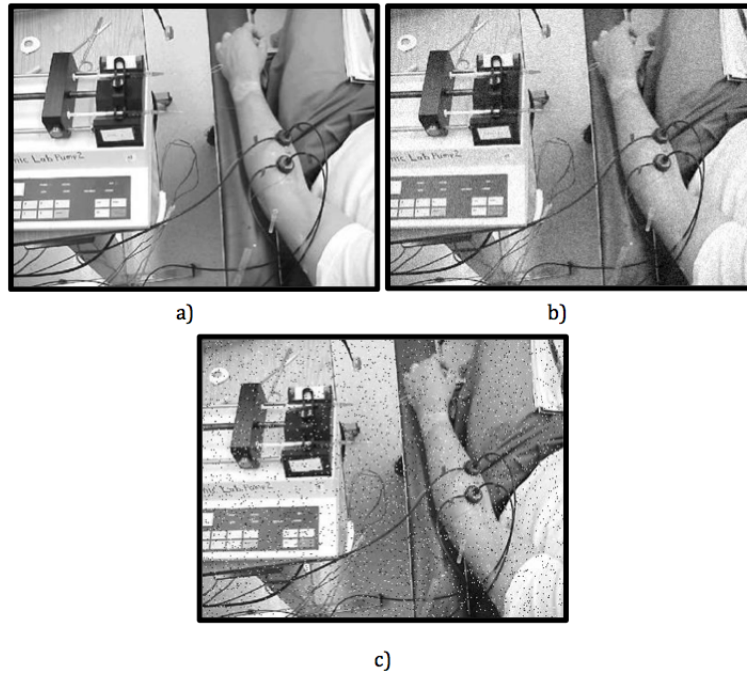


Figura 1.5: Fuentes de error en una imagen digital. a) Imagen Original. b) Imagen corrompida con ruido Gaussiano. c) Ruido Sal y Pimienta. *Tomado de <http://www.slideshare.net/omarspp/imagen-filtrado-espacial>

- Filtrado espacial y frecuencial: dado que pueden suponer una importante modificación de la imagen original, generalmente solo se consideran filtrados aquellas operaciones realizadas en el dominio espacial o frecuencial que suponen transformaciones pequeñas de la imagen original.

1.2.3. Segmentación

Es el proceso mediante el cual se divide la imagen en regiones con significado, diferenciando los objetos que aparecen y separándolos del entorno en el que se encuentran. Dicho de otra forma, esta etapa trata de detectar automáticamente los bordes o contornos entre los elementos o regiones en una escena. La segmentación es uno de los procesos más importantes en la localización de objetos en un entorno, siendo el proceso previo al de extracción y reconocimiento de objetos.

Para la segmentación de las imágenes, se usan tres conceptos básicos:

- Similitud: Los pixeles agrupados del objeto deben ser similares respecto a algún criterio (nivel de gris, color, borde, textura, etc.).
- Conectividad: Los objetos corresponden a áreas de pixeles con conectividad. Las particiones corresponden con regiones continuas de pixeles.
- Discontinuidad: Los objetos tiene formas geométricas que definen unos contornos. Estos bordes delimitan unos objetos de otros.

Partiendo de los conceptos antes expuestos, se dice, que la segmentación se basa en dos principios fundamentales: discontinuidad y similitud. Por tanto, cabe mencionar que la segmentación tiene dos distintos enfoques; aquel orientado a bordes o contornos (discontinuidad) y otro orientado a regiones (similitud).

Tanto la detección de bordes como la de regiones implican una manipulación de la imagen original, que supone en definitiva una transformación de la misma, de forma que los valores de los pixeles originales son modificados mediante ciertas funciones de transformación u operadores.

Para la detección de bordes, basados en la propiedad de discontinuidad, se proponen regularmente, los siguientes operadores:

- Operadores primera derivada
- Operadores segunda derivada
- Operadores morfológicos

Mientras que para la detección de regiones se aplican técnicas basadas en la propiedad de similitud:

- Binarización basada en el uso de umbrales
- Crecimiento de regiones mediante la adición de píxeles
- División de regiones
- Similitud de textura, color o nivel de gris.

1.2.4. Extracción y selección de características

Con los objetos a estudiar ya definidos, es necesario extraer una serie de características que los representen adecuadamente en forma de vector de características.

La descripción y representación, llamadas también selección de características, tiene como objetivos resolver el problema del reconocimiento. Tratan con la extracción de los rasgos que representan alguna información cuantitativa de interés o características que sean básicas para diferenciar una clase de objetos de otra. Los rasgos usados por el sistema dependen del tipo de objetos a ser reconocidos.

Los rasgos o características pueden clasificarse en geométricos y topológicos. Un rasgo geométrico permite, describir una propiedad geométrica del objeto; el área y el perímetro de un objeto son dos rasgos de este tipo. Un rasgo topológico permite describir una propiedad que tiene que ver con la estructura del objeto; el número de hoyos, descriptores de Fourier, el número de Euler son ejemplos de estos rasgos.

Como norma general, para la elección de características o rasgos se intenta que:

- Sean discriminantes: Tengan valores numéricos diferentes para clases diferentes.
- Sean fiables: Tengan valores parecidos dentro de una misma clase, permitiendo diferenciar objetos similares sin ser iguales.
- La descripción del objeto tiene que ser lo más completa que se pueda y no presentar ambigüedades.
- Se calculen en un tiempo aceptable, de manera que sean utilizables en problemas de tiempo real.

1.2.5. Reconocimiento e interpretación de formas

La última etapa en el procesamiento digital de imágenes, es la que comprende al reconocimiento e interpretación de formas, también denominado en conjunto como reconocimiento de patrones.

El reconocimiento es el proceso que etiqueta o asigna un nombre, a un objeto basándose en la información que proveen sus descriptores o patrones. La interpretación involucra la asignación de significado a un conjunto de objetos reconocidos.

En esta etapa del procesamiento digital de imágenes, se desarrollan algoritmos computacionales orientados a la emulación del proceso de visión humana, que permiten reconocer objetos y tomar decisiones o efectuar tareas. Para lo cual se utilizan técnicas de reconocimiento geométrico de formas, como el aprendizaje supervisado en condiciones estadísticas; algoritmos de clasificación no supervisados o *clustering*, redes neuronales, *pattern matching*, solo por mencionar algunas.

1.2.6. Conocimiento

La base de conocimiento mantiene todo el conocimiento que se tiene acerca del problema tratado. Este conocimiento puede ser simple, teniendo solo las regiones de la imagen donde la información de interés se conoce que está localizada, con lo que se limita la búsqueda que conduce a esa información. Dicha base, también puede ser compleja, tal como una lista interrelacionada de todos los defectos mayores posibles en un problema de inspección de materiales, o una base de datos de imágenes conteniendo imágenes de satélite en alta resolución, de una región en conexión con aplicaciones de detección de cambios.

Además de guiar la operación de cada etapa del proceso, la base de conocimientos también controla la interacción entre módulos. Esta comunicación entre los módulos del procesamiento digital de imágenes, generalmente se encuentra basada en el conocimiento previo de los resultados que se pueden esperar.

1.3. Antecedentes sobre análisis de texturas

El análisis de textura se basa empleando diversos métodos como lo son estadísticos, filtros, transformadas, etc. para poder calcular índices cuantitativos que describan la distribución de niveles de grises o niveles digitales en una imagen, para proporcionar información útil para cualquier tarea posterior de clasificación, segmentación, síntesis de texturas, etc.

1.3.1. Definición de textura

Para poder realizar el análisis de textura es de vital importancia entender en que consiste el estudio de la textura, aunque se puedan presentar pequeñas diferencias en su definición dependiendo del campo de estudio en el que se emplea. En el procesamiento de imágenes, son muchas las definiciones acogidas. Algunas de las más comunes son las siguientes:

”Textura es un atributo que presenta la distribución espacial de los niveles de gris de determinada región”[20].

”El término textura se refiere a la repartición de elementos básicos llamados texels. Cada texel contiene varios pixeles, cuya ubicación puede ser periódica, quasi-periódica o aleatoria. Las texturas naturales son generalmente aleatorias mientras que las artificiales son a menudo determinísticas o periódicas”[21].

”Se entiende por textura un conjunto de una escena como son su rugosidad, homogeneidad, contraste, regularidad, direccionalidad o granulosidad, entre otras, las cuales se conocen como propiedades texturales”[1].

”De manera intuitiva, puede decirse que es un descriptor que da una medida de propiedades como suavidad, heterogeneidad o regularidad”[18].

1.3.2. Objetivos del estudio de las texturas

El objetivo primordial en el estudio de las texturas presentes en una imagen es extraer las características de manera que se puedan identificar de una manera precisa y así proceder a sus respectiva segmentación o clasificación. Este objetivo final puede ser ratificado por medio de diversos autores, como lo son Haralick y Shapiro[1] que establecen que son tres los objetivos primordiales en el análisis de texturas.

- Clasificación de texturas: Para determinar a cual de un número determinado de clases físicamente definidas pertenece una región de texturas homogéneas. Se debe de tener en cuenta que las texturas homogéneas en imágenes naturales no siempre son frecuentes, por lo que la dificultad consiste en clasificar una imagen o un fragmento de ella dentro de un numero determinado de clases cualitativas conocidas previamente.

- Generar modelos de texturas a partir de otra original mediante técnicas de generación o síntesis de modelos de textura que emulan a las naturales, y que puedan servir de referencia para otras y ser empleadas en procesos de modelización gráfica por ordenador.
- Delimitar en una imagen aquellas zonas que posean texturas diferentes. Se trataría de un problema de segmentación de imágenes o también llamado de discriminación de texturas, en el que se segmentaría una imagen de texturas en regiones, cada una correspondientes a una textura perceptualmente homogénea.

Otros autores establecen objetivos complementarios a los anteriores en un proceso de análisis de textura [22].

- Extracción de características que permitan describir numéricamente sus propiedades de textura. Al conjunto de descriptores extraído de una textura dada se le denomina vector de características de textura. Se emplean generalmente los estadísticos de primer orden y los estadísticos de segundo orden. Si bien se pueden obtener un número considerable de dichos descriptores, en la mayoría de las aplicaciones es muy interesante el obtener un vector con aquellas características más discriminantes para diferenciar texturas.

otro objetivo podría ser el descrito por [23].

- Intentar reconocer de que imagen inicial proviene un fragmento de imagen dado, tarea designada como recuperación de texturas, o bien encontrar una textura dada en una base de imágenes.

1.3.3. Campos de aplicación existentes en la actualidad

Por lo que respecta a los campos donde se puede aplicar el análisis de texturas, se podría decir que las principales aplicaciones son [24] :

1. Teledetección
2. Tratamiento de imágenes médicas
3. Procesos industriales de control de producción
4. Segmentación de documentos para separar áreas de texto de aquellas que incorporan imágenes o gráficos
5. Generación y síntesis de formas a partir de texturas
6. Detección de defectos en texturas

1.3.4. Métodos de caracterización de texturas

Para llevar a cabo el procesamiento digital de imágenes es de gran utilidad, para ciertas aplicaciones tener información sobre la textura. Este problema se puede abordar desde diferentes puntos de vista, dependiendo de la técnica específica utilizada para extraer de forma cuantitativa la información contenida en la textura de la imagen.

Son muchas las propuestas hechas para el análisis de texturas [25, 26, 27]. Algunos autores difieren en cuanto a la relación de métodos para estudiar las texturas, pero la mayoría de los autores los agrupan en cuatro grupos principales:

- Estadísticos
- Estructurales
- Basado en modelos
- Basados en transformadas

1.3.4.1. Métodos estadísticos

Este Método se basa en el cálculo de descriptores de la distribución de intensidades en la imagen, los cuales son casi constantes para cada textura. Estos descriptores estadísticos miden variaciones de niveles de gris en un vecindario y se puede hablar de modelos estadísticos de 1er orden, o derivados del histograma y de los estadísticos de 2do orden de los cuales más información aportan sobre una textura, es decir los extraídos de la Matriz de Co-ocurrencia de Niveles de Gris (MCNG) [28] .

En [29] detallan otros métodos estadísticos como son:

- Gráficos polares (polar *plots*) y polarogramas.
- Estadísticas texturales en los bordes.
- Expansión armónica.
- Métodos ópticos-digitales.
- Estadísticos de alto orden.
- Momentos invariantes.
- Métodos de distribución de características.

1.3.4.2. Métodos estructurales o geométricos

Consiste en entender una textura como una serie de primitivas (microtexturas) y una jerarquía de orden espacial (macrotextura) de esas primitivas. Para describir la textura se deben definir por tanto las primitivas, su tamaño y sus formas (o calcular sus propiedades estadísticas), así como sus reglas de localización espacial [22, 30]. La ventaja de este método es que proporciona una buena descripción simbólica de la imagen, indicando si una imagen presenta o no textura, si bien es un método más útil para procesos de síntesis que para tareas de análisis.

1.3.4.3. Métodos basados en modelos

Estos métodos [31] se basan en la construcción de un modelo, cuyos parámetros estimados sobre unas sub-imagenes tipo describan las cualidades de la textura. Se emplean características derivadas de un modelo multiresolución autoregresivo [32, 29], fractales y modelos estocásticos como los campos aleatorio de Markov, Gibbs o Wold [29] para interpretar una imagen de textura mediante modelos de generación de imágenes y modelos estocásticos respectivamente.

1.3.4.4. Enfoque mediante transformadas

Los métodos anteriores operan sobre la señal definida en el dominio espacial, si bien pueden resultar más conveniente representar la imagen en otro dominio de forma que en el nuevo dominio pueda facilitar la extracción de determinadas características[31]. Para ello se aplican procesos de filtrado o de transformación a otro espacio de trabajo, pudiendo hablar de varios métodos[28, 23].

- Filtros en el dominio espacial, como densidad de bordes, filtros de energía, operadores no lineales y métodos basados en momentos espaciales
- Transformada de Fourier, basada en el análisis en el dominio de la frecuencia.
- Filtro de Gabor, partiendo de la transformada de Fourier ventana cuando la función ventana es gaussiana.
- Transformada de *wavelet*, que permite analizar la frecuencia de una imagen en función de la escala.

De todas la transformaciones, las más conocidas son las de Fourier, Gabor y *wavelet* y la idea básica en la que se apoyan es la de representar una imagen en el espacio cuyo sistema de coordenadas esta relacionado con las características de una textura, como pueden ser la frecuencia de repetición o tamaño de la textura, si bien el comportamiento de estas tres transformadas no son similares.

La transformada de Fourier genera resultados pobres debido a su limitación en la localización espacial. Los filtros de Gabor realizan mejor localización espacial, pero su empleo está limitado por la resolución en la cual se pueden localizar las estructuras espaciales en las texturas naturales.

Este último contratiempo lo solventan la transformada de *wavelet* al presentar dos rasgos importantes en su aplicación:

1. Permite variar la resolución espacial y representar texturas de diversas naturalezas.
2. Hay un amplio abanico de funciones base *wavelet* que a priori generará mejor resultado.

1.3.4.5. Comparación de los métodos de caracterización de texturas

La siguiente Tabla (1.1) muestra un estudio comparativo de los distintos métodos vistos en el apartado anterior. De este modo se podrá justificar que método es el más adecuado para este trabajo.

Métodos	Ventajas	Inconvenientes
Descriptores espectrales	Fourier	<ul style="list-style-type: none"> • Buenos resultados para imágenes homogéneas • Sólo es útil para texturas con patrones periódicos • Sólo realiza análisis en frecuencias, no análisis temporal
	Wavelet	<ul style="list-style-type: none"> • Única que permite el análisis multiresolución • Realiza análisis multiresolución que permite localizar en espacio y frecuencia • No incrementa el número de datos • Elevado coste computacional debido a que tiene varios niveles de descomposición
Descriptores estructurales	<ul style="list-style-type: none"> • Resultados rápidos si la textura está formada por un elemento estructural reconocible por el sistema 	<ul style="list-style-type: none"> • No todas las texturas están formadas por elementos repetitivos • Necesita memoria donde almacenar posibles elementos estructurales
Descriptores basados en modelos	Cadenas de Markov	<ul style="list-style-type: none"> • Resultados muy buenos para microtexturas • Interrelaciona píxeles vecinos • Alta complejidad computacional • Necesidad de definir máscara adecuada
	Fractales	<ul style="list-style-type: none"> • Buenos resultados para superficies rugosas • Solo es útil para texturas rugosas
Descriptores estadísticos	1er orden – Energía e Histogramas	<ul style="list-style-type: none"> • Sencillos y rápidos • Necesitan poca memoria • No tienen en cuenta la información dada por la posición relativa de los píxeles
	2do orden – Matriz de Co-ocurrencia	<ul style="list-style-type: none"> • Adecuada para texturas de origen aleatorio y que no respondan a un patrón determinado • Tiene en cuenta la posición relativa de los píxeles entre sí • Alta carga computacional

tabla 1.1: Estudio comparativo de los métodos de caracterización de texturas. *Elaboración propia (Julio-2012)

1.4. Sistema de visión artificial

La visión artificial, también conocida como visión por computador⁵, tiene sus orígenes en la inteligencia artificial, aunque en la actualidad se le considera una disciplina con entidad propia, dado el enorme progreso que ha experimentado en las últimas décadas.

En la literatura es posible encontrar numerosas definiciones del término Visión Artificial [33, 34, 35]. Salvo por pequeñas matices, todas ellas coinciden esencialmente en lo siguiente:

⁵del ingles *Computer Vision*

”La visión artificial comprende un conjunto de técnicas computacionales destinadas a descubrir la estructura y las propiedades del mundo tridimensional o dinámico que nos rodea, a partir de una o varias imágenes bidimensionales obtenidas de él”.

Entre las propiedades que la visión artificial trata de descubrir acerca del mundo para reconocerlo, interpretarlo o reconstruirlo, se pueden incluir las siguientes:

- Propiedades de los objetos que aparecen en la imagen, relacionadas con:
 - Su geometría: forma, tamaño, posición, etc.
 - Su movimiento: Dirección, velocidad, aceleración, etc.
 - El material del que están hechos: color, textura, brillo, etc.
- Propiedades globales de la escena recogida en una imagen:
 - Iluminación: color, intensidad, dirección, etc. de las luces.
 - Movimiento: cambio de posición del observador (cámara que captura las imágenes).

1.4.1. Arquitectura de un sistema de visión

Los pilares del sistema físico de visión por computadora, son dos: el sistema de percepción y el sistema de actuación. El primero está constituido por los sistemas de iluminación, de captación de la imagen y de adquisición de señal en la computadora. Una vez introducida la señal en la computadora, esta es procesada mediante el o los algoritmos para transformarla en información útil, para la toma de decisiones, acerca del objeto que se está analizando. Esta información es utilizada por el sistema de actuación o actuador externo del proceso (Manipulador, actuador final, autómatas programables, etc), siendo innumerables los múltiples periféricos que pueden ser receptores de la información proveniente del sistema de percepción [36] (Figura 1.6).

Un sistema de visión artificial se compone básicamente de los siguientes elementos:

- Fuente de luz: es un aspecto de vital importancia ya que debe proporcionar unas condiciones de iluminación uniformes e independientes del entorno, facilitando además, si es posible, la extracción de los rasgos de interés para una determinada aplicación.
- Sensor de imagen: es el encargado de recoger las características del objeto bajo estudio
- Tarjeta de captura o adquisición de imágenes: es la interfaz entre el sensor y la computadora o módulo de proceso que permite al mismo disponer de la información capturada por el sensor de imagen.
- Algoritmos de análisis de imagen: es la parte inteligente del sistema. Su misión consiste en aplicar las transformaciones necesarias y extracciones de información de las imágenes capturadas, con el fin de obtener los resultados para los que haya sido diseñado.
- Computadora o módulo de proceso: Es el sistema que analiza las imágenes recibidas por el sensor para extraer la información de interés en cada uno de los casos implementando y ejecutando los algoritmos diseñados para la obtención de los objetivos.
- Sistemas de respuesta en tiempo real: Con la información extraída, los sistemas de visión artificial pueden tomar decisiones que afecten al sistema productivo con el fin de mejorar la calidad global de producción.

1.4.2. Sistema de procesamiento o módulo de proceso

El sistema de procesamiento suele ser una computadora o un *cluster* de computadoras, dependiendo de las necesidades de los algoritmos de visión a utilizar; y con las cuales se aplican las herramientas de visión adecuadas, a imágenes provenientes del sensor de visión. Se trata del sistema encargado de recibir y analizar las imágenes de los objetos a inspeccionar, a una velocidad necesaria para poder interactuar con otros dispositivos en tiempo real.

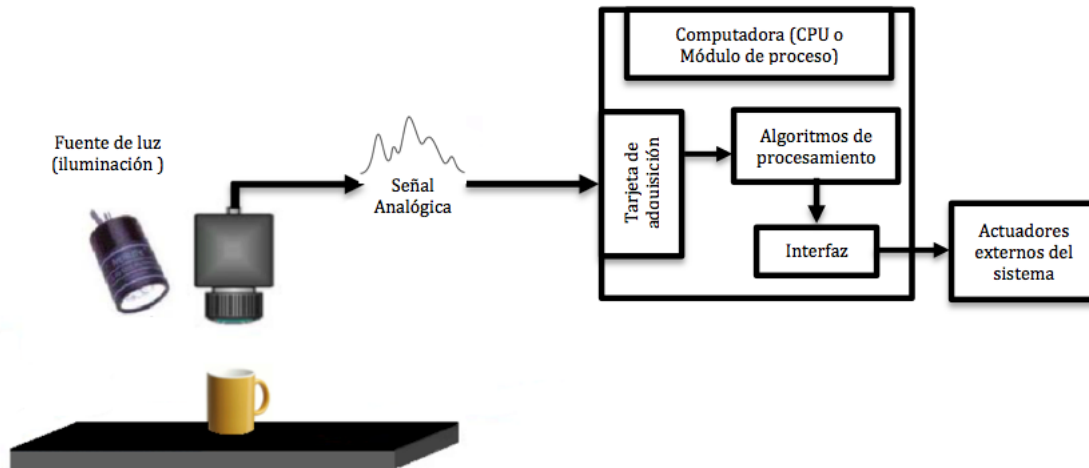


Figura 1.6: Elementos básicos de un sistema de Visión. *Elaboración propia (Julio-2012)

Partiendo de una representación digital de las imágenes, el módulo de procesamiento, procesa la información digital, hasta alcanzar información de más alto nivel (decisión binaria), ayudando a la interacción con las aplicaciones o actuadores externos del proceso.

Es por lo tanto, la parte del sistema de visión, encargada de implementar las funciones de tratamiento de la imagen en función del tipo de análisis a realizar, partiendo la ejecución de la tarea para el que se ha diseñado el sistema automático.

1.4.3. Sistema de adquisición de señales de video

La señal eléctrica procedente de las cámaras de visión o video, forman la señal de video. Hay una tendencia creciente a que su naturaleza sea de tipo digital, pero todavía existen muchas señales de video de carácter analógico.

Para ser tratadas es necesario realizar un muestreo y una cuantificación. Ambas tareas son realizadas por tarjetas de adquisición, también llamadas *frame grabbers*. Recientemente están empleando tecnología de USB o *firewire*, y de conexión a través de redes *Ethernet*.

Las tarjetas de captura o adquisición de imagen son las encargadas de digitalizar y transferir la imagen de la cámara a la memoria de la computadora a fin de realizar el procesamiento adecuado de la escena.

1.4.4. Sistema de iluminación

La iluminación se puede considerar la parte más crítica dentro de un sistema de visión artificial. Las cámaras son mucho menos sensibles y versátiles que la visión humana y las condiciones de iluminación deben optimizarse al máximo para que una cámara pueda capturar una imagen que el ojo humano podría distinguir sin necesidad de una iluminación especializada [37].

Las cámaras capturan la luz reflejada de los objetos. El propósito de la iluminación utilizada en las aplicaciones de visión es controlar la forma en que la cámara va a ver el objeto. La luz se refleja de forma distinta si se ilumina una bola de acero, que si se ilumina una hoja de papel blanco y el sistema de iluminación por tanto debe ajustarse al objeto a iluminar.

Si se utiliza una iluminación adecuada, la aplicación se resolverá más fácilmente, mientras que si la misma aplicación recibe una iluminación incorrecta puede que sea imposible de resolver. Si para resolver una aplicación es necesario utilizar muchos filtros de software, significa que la iluminación que se está aplicando no es lo suficientemente correcta. Una iluminación adecuada

permitirá emplear menos filtros en la imagen y por tanto aumentar la velocidad de proceso en esa aplicación [38].

Los sistemas de clasificación se pueden clasificar tanto por el tipo de fuente de luz como por el tipo de luz, así como por su rango espacial, entre otras clasificaciones. De esta manera, según la primera clasificación, se pueden encontrar sistemas basados en:

- Fluorescentes, mayormente de alta frecuencia
- Led
- Fuentes halógenas
- Láser
- Fibra óptica

Una clasificación posible por tipo de iluminación es [39]:

- Luz radial: la luz, al provenir del perímetro del eje de la cámara, reduce las sombras, suaviza las texturas y minimiza la influencia de rayas, polvo e imperfecciones que pueda tener el objeto inspeccionado en la toma de la imagen. La luz directa, debido a la pequeña apertura del haz de luz suministrado, hace que el sistema pueda ser empleado a mayor distancia del objeto y que proporcione una gran cantidad de luz. Empleado en la detección de marcas de diferentes colores, caracteres y detección de todo lo que suponga un cambio de color tanto en superficies lisas con brillo como en superficies rugosas a distancias elevadas.
- Proyector de luz directa: la luz producida por el proyector de LEDs llega directamente al objeto, produciendo un gran contraste y resaltando texturas, relieves y fisuras del objeto iluminado debido a que cualquier relieve, por pequeño que sea, produce una sombra muy definida. El ángulo de incidencia de la luz respecto al plano de iluminación determinará el grado de resalte de los relieves. Para ángulos muy pequeños respecto a la horizontal, la luz producirá sombras en todos los relieves de la pieza. Para ángulos cercanos a 90° respecto a la horizontal, sólo será detectable la sombra en los grandes relieves. Empleado en la detección de piezas y objetos y su correcta colocación, así como en la detección de bordes, rayas y fisuras en una dirección determinada.
- Proyector de luz difusa: este sistema de iluminación está especialmente indicado para atenuar brillos, suavizando las texturas de los objetos e iluminando éstos uniformemente. Los proyectores de LEDs están diseñados para ser colocados en cualquier posición excepto en el eje de la cámara, permitiendo así variar la sombra proyectada dependiendo de las necesidades de la pieza a inspeccionar. Indicado para poder detectar fácilmente variaciones del color como las que se producen al imprimir texto en un objeto.
- Luz efecto "campo brillante": la fuente de luz es uniforme y difusa e incide con un pequeño ángulo sobre la superficie reflectante del objeto a inspeccionar. La cámara se coloca con el mismo ángulo de forma que obtenga una imagen reflejada de la fuente de luz en la superficie del objeto inspeccionado. Una posible imperfección en el mismo producirá una mancha oscura en la imagen captada por la cámara debida a la variación del ángulo de reflexión que produce la imperfección con respecto al ángulo de la luz incidente, perdiéndose el reflejo de la fuente de iluminación. Indicado para detectar rayas, fisuras y deformaciones en objetos con superficies planas y brillante.
- Luz posterior difusa: la cámara se dirige directamente a la fuente de luz, observando una superficie blanca totalmente uniforme. Cualquier objeto interpuesto entre la cámara y la fuente de luz produce una sombra que es detectada por la cámara como una forma en negro sobre el fondo blanco producido por el iluminador, obteniendo el máximo contraste posible. Empleado en la medición precisa de los perfiles de las piezas y también en la medición de transmitancia y de impurezas en los objetos transparentes o translúcidos.
- Luz efecto "campo oscuro": luz directa de alta intensidad que incide sobre el objeto con un ángulo muy pequeño respecto a la superficie de apoyo. De esta manera se consigue que sobre superficies que presentan hendiduras o elevaciones, éstas interfieran en la trayectoria de la luz

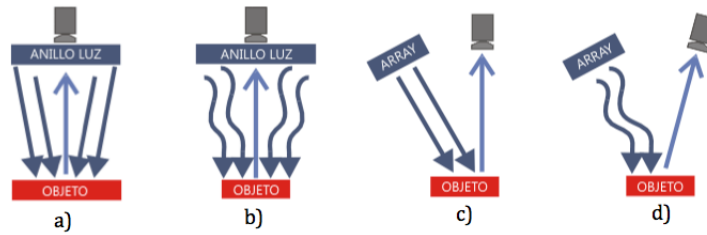


Figura 1.7: De izquierda a derecha: a) luz radial Directa b) luz radial difusa c) proyector de luz directa d) proyector de luz difusa. *Tomado de <http://www.dcmsistemas.com/index.php?sec=doc&sbsec=teo&pag=13>

produciendo zonas brillantes. Modo de iluminación empleado normalmente en la verificación de defectos superficiales o grabados como láser o troquel.

- Luz efecto "día nublado": el objeto es iluminado con luz difusa desde todas direcciones, eliminando sombras y reflejos, suavizando texturas y minimizando la influencia de rayas, polvo, relieves y curvaturas que pueda tener el objeto inspeccionado. Empleados en la detección de marcas de diferentes colores, caracteres y todo lo que supone un cambio de color en superficies lisas, esféricas, rugosas o con brillo.
- Luz axial difusa: al provenir la luz desde el mismo eje de la cámara la sombra se proyecta verticalmente y por ello no es perceptible por el objetivo. En superficies planas reflectantes si no se utilizara este método de iluminación, la cámara vería reflejado su propio objetivo. Sistema de iluminación muy útil cuando es necesario mejorar la homogeneidad evitando brillos y sombras. Empleado para inspeccionar superficies planas reflectantes y cavidades profundas.

1.4.5. Sistema de captación

La cámara es el dispositivo que utilizando un juego de lentes (objetivo) reconstruye una imagen sobre un elemento sensible (sensor), y la transmite al sistema de adquisición de la computadora. Dicha transmisión puede ser digital o analógica. En otras palabras, los sistemas de captación, son los transductores que convierten las radiación reflejada luminosa, en señales eléctricas de video.

La cámara desempeña la función de sensor en un sistema de visión por computadora. Los sensores de imagen son componentes sensible a la luz, que modifican su señal eléctrica en función de la intensidad luminosa que perciben. Existen distintas tecnologías para la captura de imágenes, siendo la naturaleza del proyecto, la que determine la más adecuada en cada caso.

Las cámaras más comunes son las basadas en CCD debido a su bajo costo, bajo consumo, buena resolución, durabilidad y facilidad de manejo. Son consideradas como dispositivos que poseen una baja deformación geométrica de la imagen, buena sensibilidad a la luz, así como resolución. El tiempo de captura de la imagen, se encuentra característicamente en el rango de 1/60 s y 1/10000s. Utilizan un *array* CCD, que se ha convertido en el estándar tecnológico de dichas cámaras.

1.4.6. Actuadores externos

Es el conjunto de elementos receptores de la información de alto nivel. Se trata de sistemas automáticos que responden electromecánicamente. Puede tratarse de monitores de altas prestaciones gráficas, robots, variadores de frecuencia, autómatas programables, dispositivos neumáticos e hidráulicos, etc. Otra de las funciones de estos sistemas, es la generación de estadística e informes del proceso al que están asociados, lo que permite disponer de una información global de los procesos para facilitar la toma de decisiones.

Capítulo 2

Estado de arte

Este capítulo se ofrece una revisión del estado de arte en la detección de defectos en superficies donde se utilizan técnicas de visión por computadora y procesamiento digital de imágenes, particularmente aquellos basados en métodos de análisis de textura, así como algunas técnicas aplicadas específicamente en las pieles curtidas.

Las técnicas utilizadas para la detección de defectos en texturas se discuten en cuatro categorías (enfoques estadísticos, estructurales, métodos basados en modelos y en filtros).

2.1. Introducción

Como menciona Xie[40], existen numerosos trabajos reportados en las últimas dos décadas durante la cual la inspección de visión por computadora se ha convertido en una de las áreas de aplicación más importante. Chin[41] y Newman[42] proporcionaron una visión general de la inspección de superficies de los años ochenta y mediados de los noventa respectivamente. Recientemente, Li y Gu[43] recopilaron los avances recientes en inspección de superficies de forma libre. Sin embargo, han habido avances significativos en los últimos años en todos los aspectos de inspección de superficies donde se utiliza visión por computadora.

2.2. Detección de defectos en texturas.

Una de las características más importantes en la identificación de defectos en superficies es la textura. La Figura (2.1) muestra algunos defectos en diferentes tipos de materiales.

La tarea de detección de defectos ha sido largamente revisada como un problema de análisis en textura. Características como grandes y pequeñas variaciones interclase se buscaron para mejorar la separación de diferentes texturas [40]. El análisis de textura ha estado en el foco de interés en los últimos veinte años. Los psicólogos, biólogos y los investigadores de visión por computadora han tratado de modelar la visión humana de la textura y encontrar reglas matemáticas que puedan determinar si la diferencia entre dos regiones es visible. Los avances obtenidos en este campo justifican el interés creciente que se observa en la clasificación automática de imágenes basadas en descriptores de textura extraídos de las imágenes.

Como se mencionó anteriormente, algunos autores difieren en cuanto a la relación de métodos para estudiar la texturas, pero la mayoría de los autores los agrupan en cuatro grupos principales: métodos estadísticos, estructurales, métodos basados en modelos y en filtros.

2.2.1. Métodos estadísticos

El modelo estadístico es considerado el más importante que se ha propuesto para el análisis de textura. Este modelo utiliza los rasgos estadísticos de la textura de la imagen, en la cual la textura es modelada como una función de los tonos de gris. Un gran número de características de textura se han propuesto que van desde estadística de primer orden a las de orden superior, como

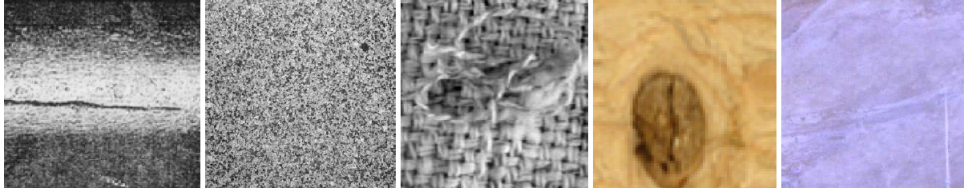


Figura 2.1: Ejemplo de defectos en diferentes tipos de superficies - De la izquierda a derecha: Acero [2], Piedra [3], textil [4], Madera [5] y Azulejo Cerámico [6].

son entre otros, estadística de histogramas, matrices de co-ocurrencia, autocorrelación y patrones locales binarios [40].

2.2.1.1. Propiedades del histograma

Comúnmente utilizados, los estadísticos de primer orden involucran pixeles simples, por ejemplo en un histograma normalizado de los niveles de gris de una imagen, se pueden considerar datos como la media, varianza, suavidad, sesgo, curtosis, correlación, energía o entropía para caracterizar una textura.

Menciona Xie[40], que a pesar de su simplicidad, las técnicas en histogramas han demostrado su eficiencia como un enfoque de bajo costo y método de bajo nivel en diversas aplicaciones tales como [44]. Estos son invariantes a la traslación y rotación. La exactitud de los métodos basados en histogramas pueden ser mejorados mediante el uso de estadística en las regiones locales de una imagen[45, 46]. Momentos simples del histograma, tal como media y desviación estándar, de sub-bloques fueron utilizados para la clasificación de los defectos[47]. Recientemente, Ng[48] propone una técnica para seleccionar automáticamente el valor de umbral óptimo para aplicaciones en detección de defectos basada en el método global del valor umbral de Otsu[49] para segmentar defectos en superficies. Sin embargo, requiere la suposición que la intensidad de las regiones defectuosas sean separables de las regiones normales, que no siempre es cierto para las superficies texturadas.

2.2.1.2. Matriz de co-ocurrencia

La matriz de Co-ocurrencia de niveles de gris (GLCM) se ha convertido en una de las características de la textura más conocida y ampliamente utilizada. Estos estadísticos de segundo orden describe la frecuencia de un nivel de gris que aparece en una relación espacial específica con otro valor de gris, dentro del área de una ventana determinada. En [40] se tienen varios trabajos que reportan el uso de matrices de co-ocurrencia para detectar defectos, tales como en [5, 47, 50]. Por ejemplo en [50], Liverinen et al. aplica las características de co-ocurrencia en la textura para detectar defectos en hojas de papel donde las texturas normales tiene una frecuencia característica.

Para tela de sarga, Tsai et al. [47] aplica la matriz de co-ocurrencia para la extracción de seis características de la imagen como parámetros de entrada para un algoritmo de red neuronal de tipo *back-propagation*. Se consiguió con éxito una tasa de detección de 96 % con 25 imágenes (5 imágenes libres de defecto y 20 defectuosas). Latif-Amet et al. [51] propuso el método de sub-bandas en la matriz de co-ocurrencia, logrando una precisión de detección del 90.78 % en 36 imágenes de tejidos lisos de 256x256. Las imágenes de los ejemplos anteriores fueron adquiridas con una cámara CCD con la calidad más pobre. En concreto, sólo cuatro muestras de tejidos defectuosos de tamaño 100x100 se muestran en [47] y no fue dado un resultado explícito después de la detección. Por lo tanto, la fiabilidad de estos métodos y cómo se aplican a otro tejido no fueron claras.

Las características de la matriz de co-ocurrencia pueden sufrir de una serie de deficiencias. Parece ser que no existe una solución generalmente aceptada para la optimización del vector de desplazamiento. También es importante asegurar que el número de entradas de cada matriz sea adecuada para ser estadísticamente fiable. Para una vector de desplazamiento dado, un gran número de características pueden ser calculados, lo que implica dedicados procedimientos de selección de las características [40]. Pero dos principales debilidades de la Matriz de Co-ocurrencia [25] son

los malos resultados en texturas construidas por grandes primitivas y uso intensivo del ordenador debido al gran número de píxeles adyacentes en el cálculo.

2.2.1.3. Autocorrelación

La característica de autocorrelación se derivan de la observación que algunas texturas son de naturaleza repetitiva, como son los textiles [40]. La función de autocorrelación puede ser usada en una imagen para evaluar su regularidad (periodicidad) así como la rugosidad de la textura presente en la misma. La función de autocorrelación se obtiene superponiendo una imagen a sí misma, moviéndola en ambas direcciones del plano. Esta estadística de segundo orden pueden ser sensibles a interferencia de ruido.

Estadística de orden superior han sido investigados, por ejemplo, Huang y Chan[52]desarrollaron un método de extracción de armónicos de estadística de orden superior para la descomposición de la textura, los resultados de simulación y experimentales demostraron que este método es eficaz para la descomposición de la textura y se comporta mejor que los métodos estadísticos de orden inferior. Sin embargo, la función de autocorrelación es considerada generalmente como inadecuada para texturas aleatorias con arreglos irregulares en sus elementos de textura [40].

2.2.1.4. Patrones locales binarios

El operador de patrones binarios locales (LBP) es uno de los descriptores de textura con mejores resultados y ha sido utilizados en numerosas aplicaciones. Este método asigna una etiqueta a cada pixel de una imagen usando en una vecindad de 3×3 píxeles el valor del pixel central como umbral y considerando el resultado como un número binario [53].

Se ha mostrados que este operador es altamente discriminatorio y tiene varias ventajas como por ejemplo, la invarianza con respecto a los cambios de iluminación y rotación en una imagen, así como la eficiencia computacional, lo cual lo hace apropiado para las tareas demandadas en el análisis de imágenes. Xie [40] menciona que se ha aplicado para la detección de defectos en superficies de baldosas cerámicas[53], y madera[7, 8] e inspección en tiempo real[54]; y que a pesar de que se ha logrado un buen desempeño en la clasificación de texturas, LBP ha reportado un rendimiento considerablemente más bajos que la matriz de co-ocurrencia y los enfoques basados en filtros en detección de defectos en texturas de baldosas cerámicas, en las que las textura generalmente tiene un arreglo aleatorio.

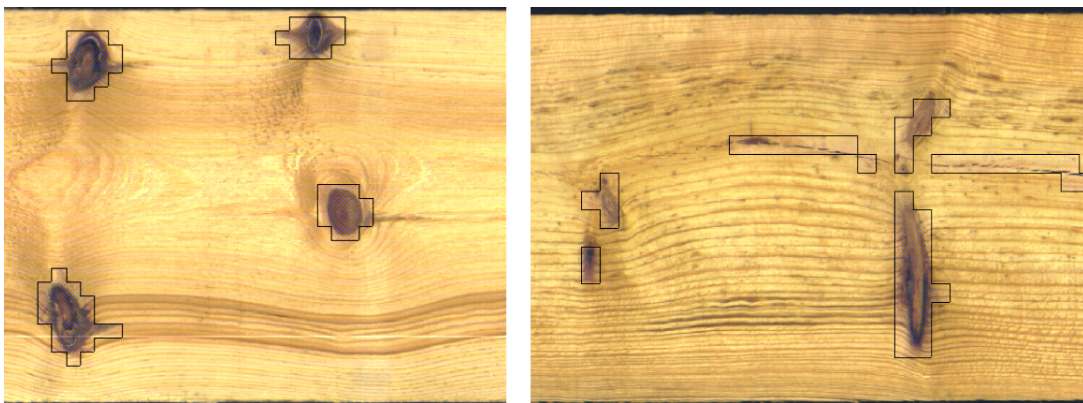


Figura 2.2: Localización de defectos en madera. *Tomado de Niskanen et al.[7, 8]

2.2.2. Enfoques estructurales

En los enfoques estructurales, su desarrollo se basa en obtener patrones primitivos, texel (del inglés "texture element"), de la textura para generar una descripción de cómo se agrupan éstos para formar la textura en sí. Es decir, es una primitiva visual con propiedades invariantes que ocurre a diferentes posiciones, deformaciones y orientaciones en un área. Estas propiedades invariantes pueden ser: forma, tamaño, nivel de gris o color.

En [55], Chen y Jain desarrollaron un método, donde la imagen de una textura primero es umbralizada utilizando análisis de histograma y después mapeada en una estructura de datos que representa la estructura esqueleto de la textura. Estas mediciones fueron comparadas con un rango de muestras de aceptación predefinido para detectar defectos. Kittler et al. [9] presenta diferentes algoritmos para la detección de anomalías tanto en propiedades cromáticas y estructurales de texturas aleatorias. Los resultados presentados son muy prometedores en la detección de grietas, burbujas y defectos cromáticos en azulejos cerámicos y granito. Wen y Xia[56] llevaron a cabo detección de defectos superficiales en piel curtida mediante la extracción de segmentos de borde y evaluación estadística de estos bordes, por ejemplo, en función de su longitud e intensidad [40].

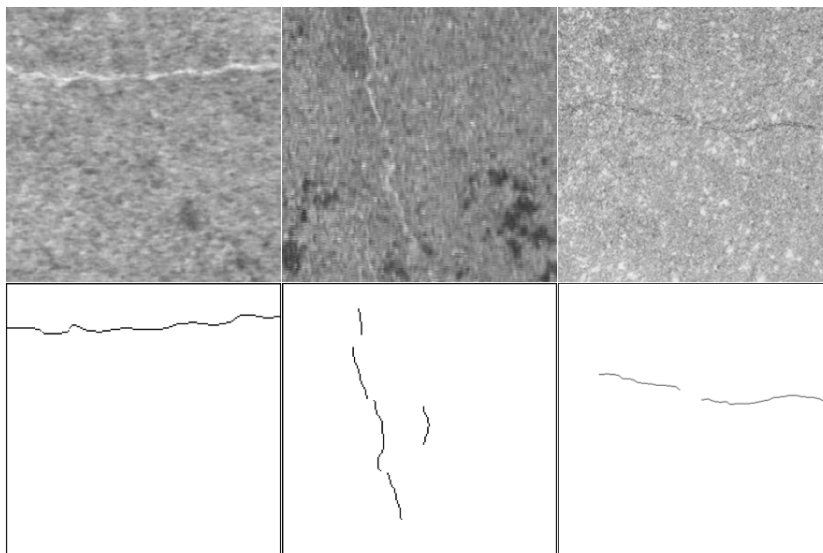


Figura 2.3: Detección de grietas en azulejos de granito. *Tomado de Kittler et al.[9]

2.2.3. Enfoques basado en filtros

Las técnicas que se examinan en esta sección comparten una característica en común: la aplicación de banco de filtros en la imagen y el cálculo de energía de la respuesta de dichos filtros. Los métodos pueden clasificarse en las siguientes categorías: dominio espacial, dominio espacio-frecuencia.

2.2.3.1. Filtrado en el dominio espacio y frecuencia

Las mediciones de intensidad y frecuencia en los bordes constituyen uno de los primeros intentos para discriminar texturas. En el dominio espacial, las imágenes son procesadas generalmente por filtros gradiente para extraer, entre otros, bordes, líneas o puntos aislados. Los filtros Sobel, Robert, Canny, Laplaciano y Deriche han sido utilizados como precursores de medición en la densidad de bordes [40].

El filtrado [57] es utilizado en muchas aplicaciones (por ejemplo, en la mejora de la imagen). Hay dos métodos de filtrado: (1) Filtrado en el dominio de la frecuencia basado en la transformada de Fourier, y (2) filtrado espacial basado en operaciones directas en los pixeles de la imagen, ambos son sensibles a ruido en la imagen. Neubauer [58] recomienda un método de segmentación de defectos basado en múltiples filtros lineales (incluyendo tres filtros de convolución separables como estadística de primer orden). Representando sólo una muestra de tejido en mala calidad, la tasa de verdaderos positivos y de verdaderos negativos de detección fue de 98.3% y 90.6% respectivamente.

En [10], los autores utilizaron la transformación de Fourier para reconstruir imágenes textiles para la detección de defectos. La diferencia entre la imagen restaurada y la imagen original fueron

consideradas como posibles defectos. Una idea similar fue explorada en [59] pero en este caso se utilizó el filtro paso bajas para eliminar información periódica.

Chan y Pang [60] extrajeron picos armónicos de porciones horizontales y verticales del espectro de potencia, basado sobre la observación de que los defectos generalmente ocurren en direcciones horizontales y verticales. Sin embargo, todos estos métodos se basan en la suposición que la tela sin defectos es una textura repetitiva y regular. Estos métodos no son adecuados para la detección de defectos en texturas aleatorias [40].

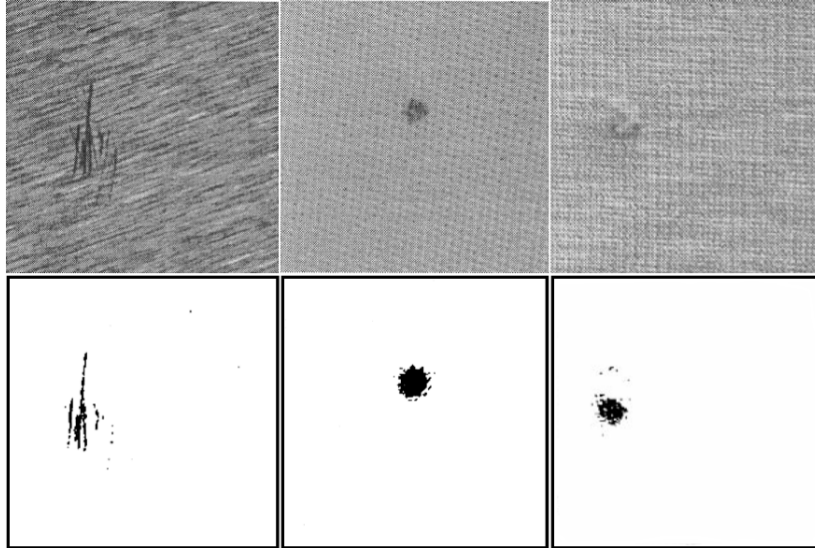


Figura 2.4: Detección de defectos en textiles. *Tomado de Tsai y Hsieh[10]

2.2.3.2. Métodos espacial-frecuencial.

La transformada de Fourier es ampliamente utilizada en el procesamiento y análisis de señales (en nuestro caso imágenes) y con resultados satisfactorios en los casos en que estas señales son periódicas y lo suficientemente regulares, pero no ocurre lo mismo para el análisis de señales cuyo espectro varía con el tiempo. El modo clásico de introducir la dependencia espacial dentro del análisis de Fourier es a través de la transformada de Fourier por ventanas. Si la función ventana es Gausiana, la transformada de Fourier por ventana se convierte en la bien conocida transformada Gabor, lo que posiblemente puede conseguir la localización óptima en el dominio espacial y frecuencial. Turner y Bovik et al. propusieron por primera vez el uso de filtros de Gabor en el análisis de la textura [40].

A mediados de los años 80 se empezó a desarrollar una herramienta matemática, la transformada wavelet. La transformada *wavelet* mapea la imagen en una sub-imagen de baja resolución, o imagen tendencia, y una serie de imágenes de detalles. La imagen tendencia se obtiene enturbiando iterativamente la imagen, mientras que las imágenes de detalles contienen la información perdida durante dicha operación.

Como menciona Xie [40], la resolución espacial de la transformada de *wavelet* se adapta a su contenido de frecuencia, a diferencia de la transformada de Gabor la resolución espacial es constante. En [61], Sari-Sarraf y Goddard realizaron transformaciones discretas de *wavelet* sobre imágenes de tela. Las imágenes de detalles fueron fucionadas para producir un mapa de características en las cual las regiones de textura normal, suponen que son homogéneas y tiene valores pequeños. Los defectos fueron segmentados por umbrales conocidos a partir de plantillas. El proceso clave fue para atenuar las regiones normales, y acentuar las regiones defectuosas, basadas en el supuesto de que la textura normal era regular y homogénea, y los defectos fueron los que rompieron la homogeneidad local. Scharcanski [62] también utilizó la transformada discreta de *wavelet* para clasificar textiles estocásticos. Kim et al [63] utilizó un proceso de aprendizaje para elegir las escalas de *wavelet* para maximizar la detectibilidad de los defectos en telas. Latif-Amet et al. [51] extrae la co-ocurrencia

y características bases de MRF a partir de los coeficientes de la transformada de *wavelet* para la detección de defectos en tejidos. Muy recientemente, Truchetet y Laligant [64] hizo un análisis muy detallado del análisis de *wavelet* en aplicaciones industriales.

2.2.3.3. Enfoque basado en modelos

Estos métodos se basan en construir de un modelo de la imagen, cuyos parámetros estimados sirven para describir y sintetizar una textura, ya que reúnen las características esenciales que caracterizan a la textura. Se consideran como tales las cadenas de Markov, los modelos fractales, los modelos autorregresivos, modelos de campo, modelo de paradigma y el modelo texem.

El análisis de texturas basándose en fractales fue introducido por Mandelbrot [65] y muestra la correlación existente entre la dimensión fractal de una textura y su 'tosquedad'. La descripción fractal de texturas se basa en la determinación de la dimensión fractal. Dimensión fractal y lacunaridad son las medidas más importantes en los modelos fractales. Cita Xie [40] que en un estudio comparativo realizado por Ohannian y Dubes, el método fractal se desempeñó razonablemente bien en contra de las matrices de co-ocurrencia, filtros Gabor y campos aleatorios de Markov en la clasificación de textura. Sin embargo, se ha logrado un éxito limitado en aplicaciones reales. Los fractales pueden tener la misma dimensión fractal, pero un aspecto completamente diferente. Este modelo es especialmente idóneo para texturas naturales donde se puede tener auto-similitud¹.

Los métodos que utilizan los campos aleatorios de Markov (MRF) caracterizan la relación estadística entre un píxel y sus vecinos. Es decir, este método se basa en la hipótesis de que la intensidad de un píxel depende de las intensidades de los píxeles vecinos. En [66] los autores utilizan los campos aleatorios de Markov para modelar la imagen de una textura no defectuosa. El proceso de inspección se presenta como un problema de prueba de hipótesis en las estadísticas derivadas del modelo. La imagen del tejido a inspeccionar es dividida en ventanas no sobre puestas, donde cada ventana se clasifica como defectuosa o no defectuosa. Los métodos basados en MRF fueron competitivos en un estudio de comparación contra otros métodos basados en estadística y espectrales en la detección de defectos [40].

Hace poco tiempo Xie y Mirmehdi [67] propusieron un nuevo modelo estadístico llamado "texture exemplars" o "texems", para representar y analizar texturas aleatorias donde solo se requieren muy pocas imágenes libres de defecto para el entrenamiento no supervisado. Cada imagen producto es dividida en parches superpuestos de diversos tamaños. Entonces, los modelos de densidad mixta son aplicados para reducir las agrupaciones de parches a un número de texturas ejemplo, denominado aquí como texems, caracterizado por su valores medios y covarianzas correspondientes. Cada conjunto de estos texems puede comprender varios tamaños de imágenes en diferentes escalas.

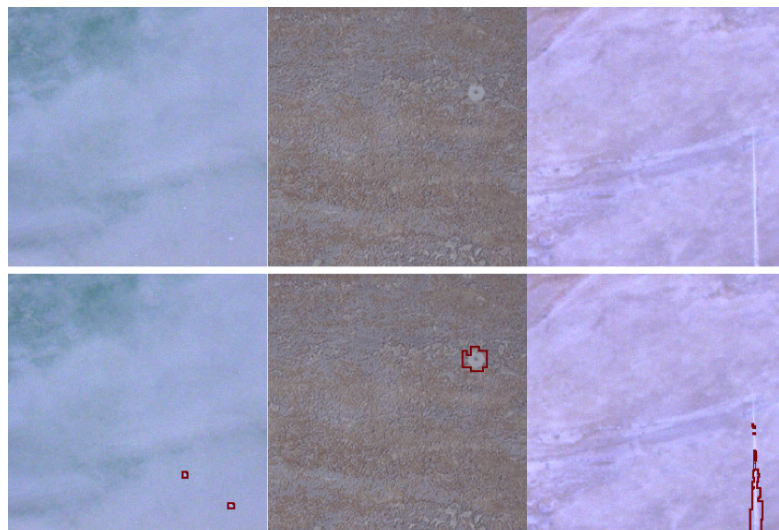


Figura 2.5: Localización de defectos en texturas aleatorias. *Tomado de Xie y Mirmehdi[11]

¹las partes tienen la misma estructura que el conjunto total

2.3. Enfoques aplicados a la piel curtida

Hay muchos tipos de defectos superficiales en pieles curtidas. El sistema LeaVis [68] fue el primer sistema que utiliza un método de visión artificial para detectar defectos en pieles curtidas. El sistema fue destinado a procesar y segmentar las imágenes con defectos y las zonas de diferente calidad, que suelen ser marcados manualmente.

Jian et al.[69] proponen un método para detectar y clasificar automáticamente defectos superficiales en pieles curtidas . En primer lugar, la imagen es preprocesada por métodos como eliminación de ruido, filtrado, separación de defectos del fondo y mejora. Después, de acuerdo con la imagen procesada, se extraen los atributos de propiedad de la región de interés (defectos en la región). Los autores utilizan una red neuronal con propagación hacia delante combinado con un árbol de decisión para seleccionar atributos óptimos y clasificar los defectos. Los resultados mostraron que el uso de la combinación de FNN y el algoritmo del árbol de decisión, tuvieron una exactitud en la inspección de defectos y clasificación de más del 90 %.

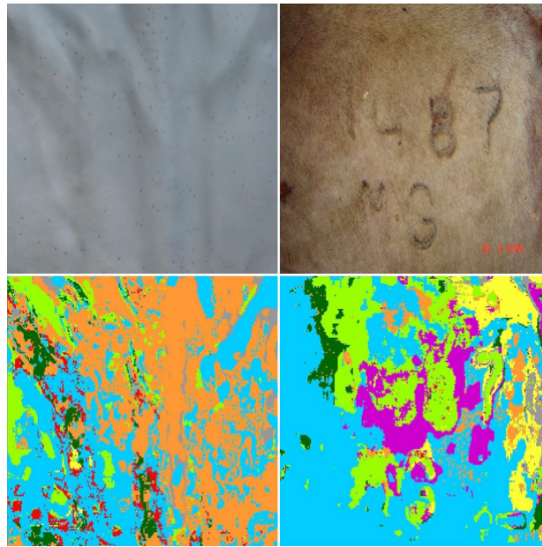


Figura 2.6: Detección de defectos en piel. *Tomado de Amorim et. al.[12]

Amorim et al.[12] comparan cinco técnicas de análisis discriminante: FisherFace, CLDA, DLDA, YLDA y KLDA. La reducción de atributos se aplica al problema de la clasificación de defectos en pieles curtidas utilizando cuatro diferentes clasificadores: C4.5, kNN, Naïve Bayes y SVM. Los resultados mostraron un mejor desempeño en CLDA y DLDA, al mantener una tasa de clasificación correcta. kNN alcanzó el mejor CCR utilizando la máxima y la mínima cantidad de atributos en la prueba. Sin embargo, los tiempos de clasificación para kNN han demostrado ser mucho más alto que C4.5 para este problema.

En [13] se presenta una nueva metodología para detectar defectos en piel curtida basada en la transformada de *wavelet*, que utiliza un banco de filtros optimizados, donde se ajusta cada filtro a un cierto defecto. Estos filtros y los rangos de *wavelets* son elegidos en base a la maximización de los atributos obtenidos a partir de los defectos y regiones de la piel. En este tipo de método se pueden detectar defectos incluso cuando un pequeño cambio en los atributos sucede. Además, esta técnica ha demostrado ser lo suficientemente rápido para la detección en tiempo real.

El rendimiento de las máquinas de soporte vectorial, como muchos otros algoritmos de aprendizaje automático, es muy sensible al parámetro de ajuste, sobre todo en los problemas del mundo real. En [70] se presenta una evaluación de la SVM contra AdaBoost y MLP, para resolver el problema de la clasificación de defectos en pieles curtidas. El resultado obtenido son muy prometedores en el éxito de discriminación de defectos en pieles curtidas, con la precisión de 99.59 %, logrado aplicando una SVM.

Un método de inspección, basado sobre *wavelets* Haar, es presentado por Sobral en [71]. El sistema se realizó para inspección en tiempo real. La metodología propuesta ha sido aplicada a la inspección de piel curtida. Cada imagen muestra fue adquirida con una resolución de 5.5 pixeles por milímetro, consiguiendo imágenes con una resolución de 2048×2048 , que fueron procesadas para tener una intensidad y contraste promedio. Las pruebas fueron realizadas en 80 imágenes, que contiene 129 defectos. Todas la pruebas se realizaron con wavelet del tipo Haar. Los resultados presentados en la inspección de piel curtida mostraron que las mascararas utilizadas pueden detectar la mayoría de los defectos y que estas pueden ser genéricas, ya que alcanzó una tasa de detección aceptable cuando se probó en una nueva serie de muestras.

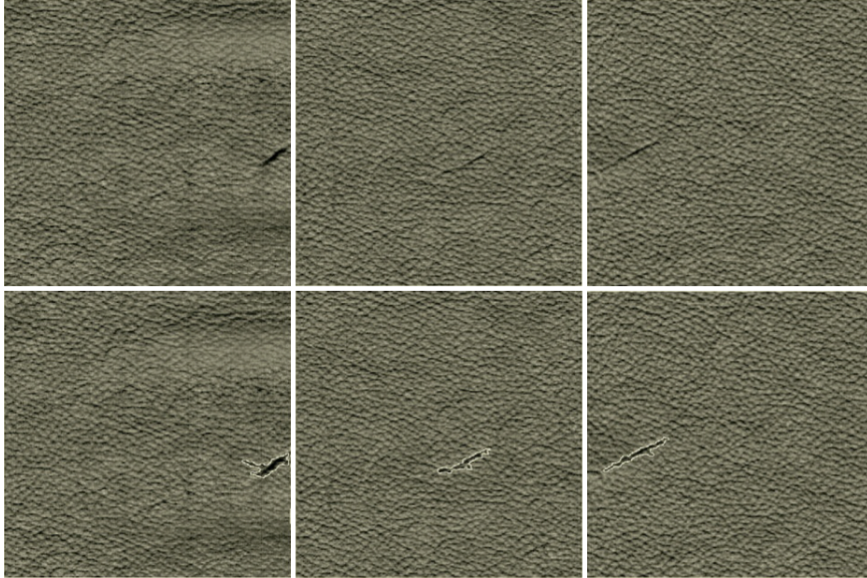


Figura 2.7: Muestras de piel curtida y los resultados de inspección. *Tomado de Sobral[13]

En [17] se describe un sistema de visión automatizada para detectar y clasificar defectos superficiales en piel curtida. En el proceso de inspección de defectos, los defectos visuales se encuentran e informan a través de un procedimiento de segmentación en dos etapas basados en el umbral y procesamiento morfológico. En el proceso de clasificación de defectos, el sistema utiliza características tanto geométricas y estadísticas como un conjunto de características. En un esfuerzo para maximizar la eficiencia en la clasificación, una de tres etapas secuenciales del clasificador de árbol de decisión se adopta para la clasificación de 5 tipos de defectos: líneas, agujeros, manchas, desgastes y rasguños. Los resultados en la prueba de clasificación tuvieron una precisión global del 91.25 %.

En [72] Kumar y Pang se basaron en filtros Gabor y aunque no se indica claramente, el sistema parece haber sido probado sólo en cuero terminado, un problema mucho más simple que cuero crudo o extracción de defectos en piel *wet blue*. En [73] muestra un método de detección basado en histogramas, utilizando los criterios χ^2 para el análisis de imágenes y construcción de histogramas. Este método compara la distancia entre el histograma de la zona analizada y el histograma de las áreas con defectos. Los resultados de la prueba χ^2 y un umbral elegido experimentalmente se utiliza para segmentar regiones defectuosas de la piel curtida.

Se utiliza un enfoque eficiente en [74], utilizando un paquete de *wavelets* para la detección de defectos en imágenes de superficies de piel curtida. Cada imagen de la piel en inspección es descompuesta con una familia de *wavelets* ortonormales. Los coeficientes del paquete de *wavelets* de un conjunto de canales de frecuencia dominantes que contiene información significativa se utilizan para caracterizar las imágenes de la piel curtida. La clasificación de defectos se realiza mediante una red neuronal. Los resultados de la detección de defectos para algunos de los defectos en la piel curtida con variaciones de intensidad muy pequeñas fueron excelentes (100 % de detección). Estos sugieren que el esquema propuesto puede identificar con éxito los defectos, y puede ser usado para

la inspección automatizada de la piel curtida.

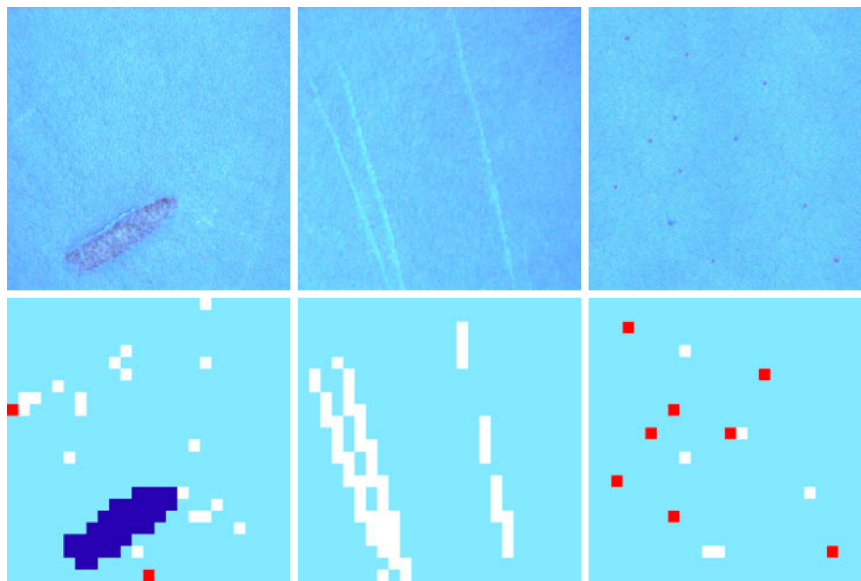


Figura 2.8: Detección de defectos en piel *Wet Blue*. *Tomado de Villar et al.[14]

Capítulo 3

Caracterización de texturas

La inspección automática para productos con aspecto de textura natural es una tarea difícil. Productos tales como azulejos, tejidos, piel curtida y madera proyectan imágenes de texturas que no pueden ser modeladas como periódicas o de una manera regular, por lo tanto se requiere un modelado estocástico de la distribución de intensidad local. Un sistema de inspección para sustituir inspectores humanos debe ser flexible en la detección de defectos como son rasguños, grietas y manchas que ocurren en diferentes formas y tamaños. Por ello en las últimas décadas se han propuesto varios métodos para extraer características de las texturas, pero el problema recae en la dificultad de encontrar un método óptimo [67].

A partir del histograma de una imagen se pueden derivar los estadísticos de primer orden y usarse como características de textura. Pronto se sostuvo que no eran suficientes para una descripción adecuada de la textura, y se vio en la necesidad de introducir estadísticos de segundo orden. Estos se reflejan de forma eficiente en las características calculadas a partir de la matriz de co-ocurrencia. La conjetura de que los estadísticos de segundo orden eran suficientes para el análisis de textura fue rechazada más tarde y se introdujeron otros esquemas de análisis de texturas, como los basados en campos aleatorios de Markov o modelos fractales [75].

El punto débil que tiene en común todos estos análisis de texturas es que la imagen se analiza en una única escala. Estudios del sistema visual humano sostiene que el córtex visual puede ser modelado como un sistema de canales independientes, cada uno con una orientación y sintonización de frecuencia espacial determinadas. Por lo tanto esa limitación puede ser superada empleando representaciones multi-escala¹ [76]. Varios sistemas de análisis de texturas han sido descubiertos. En particular los filtros Gabor fueron empleados para llevar a cabo la segmentación de texturas. Sin embargo la teoría *wavelet*² ha llegado a ser el marco matemático más idóneo para el análisis de imagen multiescala [77]. En conclusión, este trabajo utiliza un algoritmo que combina la estadística y la visión multiescala y se intenta demostrar que la textura de piel curtida de bovino puede ser caracterizada a partir de las propiedades estadísticas de los coeficientes de su representación multiescala.

3.1. Introducción

En este proyecto se ocupa un algoritmo de detección de defectos en texturas que utiliza un enfoque de procesamiento de señales que consta de una descomposición discreta de *wavelet frame* combinada con un enfoque estadístico basado en la estadística de la matriz de co-ocurrencia de niveles de gris (GLCM), para caracterizar visualmente las texturas. La imagen capturada de la superficie de una textura es inicialmente descompuesta en varias sub-bandas (diferentes resoluciones y orientaciones) utilizando *overcomplete*³ y *wavelet frame* invariantes al cambio. Este análisis

¹Una representación multi-escala se compone de múltiples versiones de la misma imagen con resolución diferente.

²Por razones de inclinación personal y de fidelidad a la nomenclatura encontrada en la literatura internacional, en este proyecto se ha preferido la designación *wavelet* a la castellanización ondícula.

³La representación *overcomplete* se trata de una representación redundante en la que no se realiza el sub-muestreo, es decir, se calcula omitiendo el submuestreo de las imágenes de detalle en cada paso. De esta forma se obtiene una representación más estable [78]

multiresolución⁴ facilita la inspección en múltiples escalas y orientaciones. Las imágenes de las sub-bandas son divididas en ventanas no superpuestas (véase Figura 3.4) y la distribución de coeficientes en cada ventana es analizada utilizando una matriz de co-ocurrencia de niveles de grises. Estadística de segundo orden, conocidas comúnmente como características de Haralick[79], son calculados de la representación de la GLCM. Las características de Haralick extraídas de todas la sub-bandas se concatenan para representar particiones en un espacio de características de una dimensión mayor donde la separación lineal sea posible. Eventualmente, una máquina de vectores de soporte (SVM) es entrenada para aprender una descripción general de una textura normal bajo inspección en este espacio de características. Este algoritmo esta implementado por LabVIEW[®] y es probado para inspeccionar visualmente imágenes de piel curtida de bovino en tiempo real.

3.2. Detección de defectos de textura por LabVIEW[®]

La detección de defectos en textura se basa en un entrenador clasificador (*NI Texture Training Interface*) que utiliza texturas muestra que no contiene defectos. El clasificador es entrenado para reconocer texturas muestras que son aceptables en la inspección. El algoritmo acepta como entrada una imagen de la superficie de una textura, identifica los defectos y devuelve una imagen binaria del defecto en esta. véase Figura (3.1).

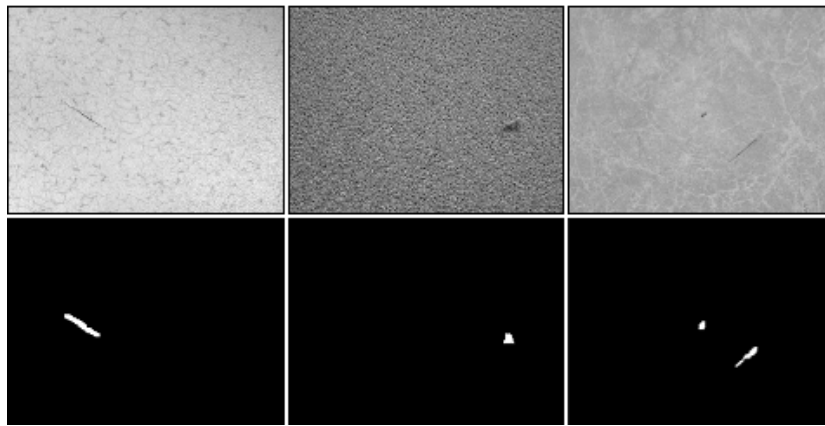


Figura 3.1: Entrada típica y sus imágenes de salida después de ser analizadas. *Tomado de *NI Vision Concepts Help*[15]

3.2.1. Descomposición wavelet frame

El primer paso en el entrenador clasificador para la detección de defectos en texturas es un análisis multiresolución de la imagen bajo inspección. El análisis multiresolución se inspira en descubrimientos psicovisuales que muestran que los humanos y otros mamíferos perciben imágenes de una manera multi-escala [76]. Muchos investigadores en la disciplina de la visión por computadora han intentado simular este comportamiento utilizando herramientas matemáticas de análisis como la transformada de *wavelet*.

La tradicional transformada discreta de wavelet (DWT) es utilizada generalmente para texturas multiresolución y aplicaciones de análisis de imagen. Uno de los inconvenientes asociados con la transformada discreta de *wavelet* es que un simple cambio en la imagen de entrada dará lugar a una serie de cambios en los coeficientes de *wavelet*. Otro problema de la transformada de *wavelet* es que las señales filtradas se muestrean en cada iteración resultando en indeseables dependencias intraescala e interscala. El algoritmo de detección de defectos en texturas que se implementa en este trabajo utiliza la transformada discreta de *wavelet frame* (DWFT) propuesta por Unser⁵ para obtener características invariantes a la translación de texturas con dependencia mínima entre los coeficientes de transformación. La transformada de *wavelet frame* son variaciones

⁴El análisis multiresolución (MRA) se basa en la descomposición wavelet y permite descomponer una señal de entrada (bien unidimensional, bien bidimensional en forma de imagen) en sus componentes de distintas frecuencias y estudiar cada componente a una resolución distinta

⁵Para mayor información a cerca de la transformada discreta de wavelet ver [80]

de la transformada de *wavelet* en la cual la salida del banco de filtros no son sub-muestreadas y como resultado, cada sub-banda de la imagen tiene el mismo tamaño que la imagen de entrada y un mejor rendimiento en la clasificación y segmentación de texturas[80].

Para análisis de textura, la transformada discreta de *wavelet* (DWT) por análisis y descomposición jerárquico de la señal es implementado por medio de un filtrado iterativo y operaciones de reducción en la resolución con un filtro paso bajas *h* y paso altas *g*. La figura (3.2) ilustra una etapa de DWT de dos dimensiones donde $2 \downarrow 1$ denota la reducción de la resolución por un factor de 2.

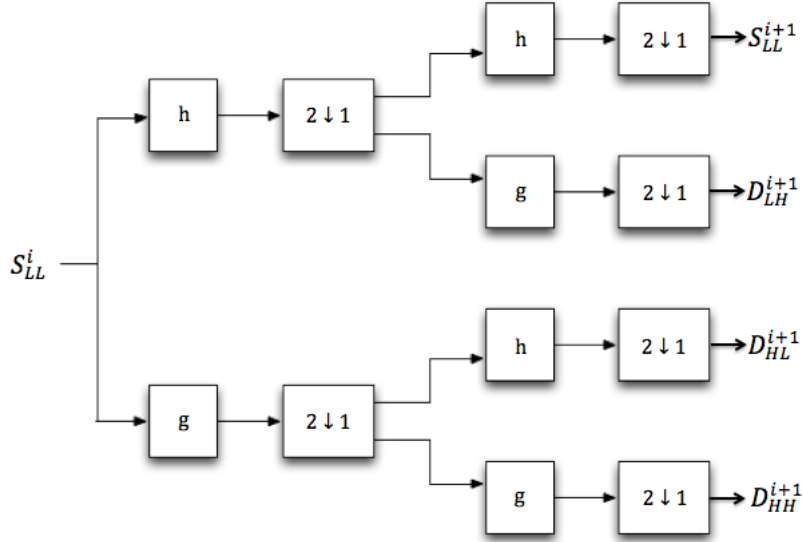


Figura 3.2: Esquema de la transformada de wavelet en dos dimensiones. *Tomado de *NI Vision Concepts Help*[15]

Los filtros *h* y *g* y sus correspondientes contraparte de reconstrucción satisfacen la restricción general de reconstrucción perfecta $h(z) \sim h(z - 1) + g(z) \sim g(z - 1) = 1$ en el dominio de la transformada *z*. En cada iteración, los coeficientes de la aproximación ordinaria, s_{i+1} , y los coeficientes de detalle, d_{i+1} , son calculados de los coeficientes presentes, s_i , por:

$$\begin{cases} s_{i+1}(k) = [h \times s_i(k)] \downarrow_2 \\ d_{i+1}(k) = [g \times d_i(k)] \downarrow_2 \quad (i = 0, \dots, I) \end{cases} \quad (3.1)$$

Donde $s_0(k) = f(k)$ es la señal de entrada del banco de filtros. Este concepto es extendido para señales discretas de 2 dimensiones (imágenes) donde filtros de segundo orden son obtenidos por el producto tensorial de un filtro pasobajas *h* y paso altas *g* de 1 dimensión a lo largo de las filas y columnas. Después de una etapa de descomposición la imagen con resolución *i* se descompone en cuatro imágenes de sub-bandas (una aproximación ordinaria s_{i+1}^{LL} , y tres imágenes detalladas d_{i+1}^{LH} , d_{i+1}^{HL} , y d_{i+1}^{HH}). Las tres imágenes de los detalles de las sub-bandas se conocen como los detalles horizontal (H), vertical (V) y diagonal (D), respectivamente.

Unser[80] propuso una descomposición más completa de la DWFT, mostrando que constituye un marco estrecho de *l2*. Unser implementó el siguiente algoritmo de descomposición iterativo. Esta descomposición DWFT es formulada en (3.2).

$$\begin{cases} s_{i+1}(k) = [h]_{\downarrow 2} \times s_i(k) \\ d_{i+1}(k) = [g]_{\downarrow 2} \times d_i(k) \quad (i = 0, \dots, I) \end{cases} \quad (3.2)$$

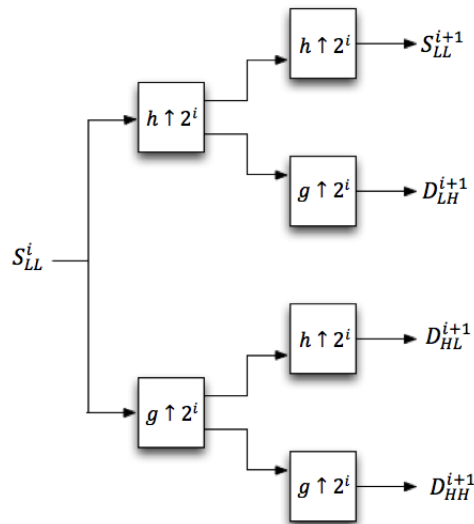


Figura 3.3: Esquema de la transformada discreta de *wavelet frame* de 2 dimensiones. *Tomado de *NI Vision Concepts Help*

Donde $s_0(k) = f(k)$ es la señal de entrada para el banco de filtros. La figura(3.3) ilustra una etapa de un DWTF de 2 dimensiones donde los filtros de una dimensión $[h]_{\uparrow 2^i}$ y $[g]_{\uparrow 2^i}$ son usados para realizar convoluciones sucesivas a lo largo de las filas y columnas de la imagen.

Los filtros de una dimensión $[h]_{\uparrow 2^i}$ y $[g]_{\uparrow 2^i}$ son los filtros h y g expandidos mediante la inserción de un numero apropiado de ceros $(2i - 1)$ entre las tomas del filtro h. Porque no hay sub-muestreo diádico en este DWTF, la descomposición de las imágenes de las sub-bandas son del mismo tamaño que la imagen original.

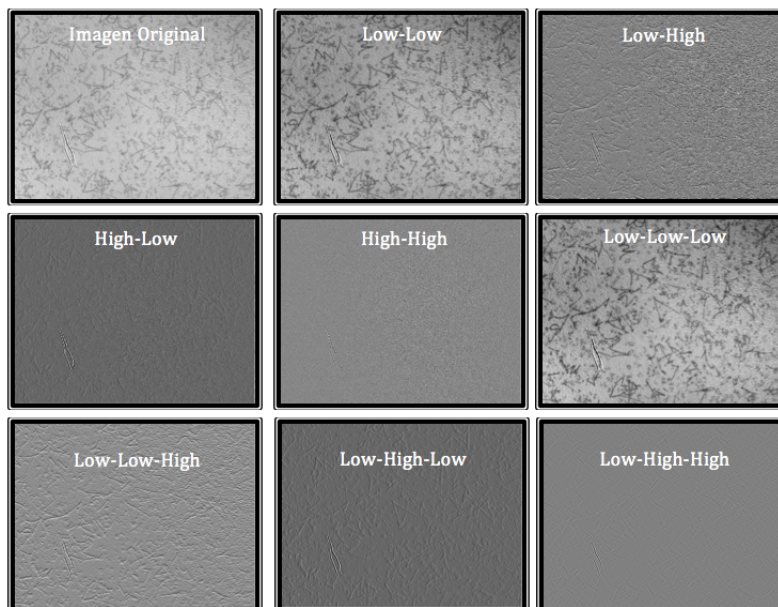


Figura 3.4: La siguiente figura ilustra las imágenes derivadas de las subbandas de dos niveles de descomposición. *Elaboración propia (Julio-2012)

El algoritmo de detección de defectos en texturas realiza dos niveles de descomposición para cada imagen inspeccionada. El primer paso de descomposición produce un aproximación burda A

y detalles horizontal, vertical y diagonales (H, V y D). La aproximación burda A es descompuesta de nuevo para producir subbandas AA, AV, AH y AD.

3.2.1.1. Tipos de *wavelet*

National instruments[®] incluye varios tipos de *wavelet* para la descomposición discreta de *wavelet frame*. La tabla (3.1) describe los tipos de *wavelets* y proporciona notas para seleccionar un tipo de *wavelet*. Por *default*, el entrenador clasificador para la detección de defectos en texturas utiliza *wavelet* biortogonales para la descomposición de las subbandas, lo que significa que los filtros de análisis h y g son diferentes de los filtros de síntesis $\sim h$ y $\sim g$. En comparación con *wavelet* ortogonales, los *wavelet* biortogonales tiene una mayor regularidad, tiene respuesta al pulso infinita, y preserva mejor la fase lineal.

Tipo	Nombre	Abreviatura	Notas
Wavelets ortogonales	Haar	-	Los wavelets ortogonales no son redundantes y son adecuadas para eliminación de ruido y compresión de imágenes.
	Daubechies	dbxx	
	Coiflets	coifxx	
	Symmlets	symx	
Wavelets Biortogonales	-	biorx_x	Los wavelets biortogonales suelen tener una propiedad de fase lineal y son adecuadas para la extracción de características en la imagen.
x indica el orden del wavelet. Un orden alto indica un wavelets más suave y regular.			

tabla 3.1: Tipos de *wavelets*. *Tomado de *NI Vision Concepts Help*[15]

3.2.2. Extracción de características estadísticas

Las características extraídas por el algoritmo son características de Haralick que describen estadística de segundo orden. Descubrimientos Psicológicos por Julesz⁶ indica que el ojo humano no puede hacer una discriminación preatentivo de texturas que tiene estadística idéntica de primer y segundo orden. Por lo tanto, un sistema de inspección automático que compita con los inspectores humanos al menos debe de tener en cuenta las estadísticas de segundo orden. Haralick et al.⁷ define características estadísticas de segundo orden basadas en Matrices de co-ocurrencia de niveles de gris (GLCM). Las características de Haralick son comúnmente usadas para la identificación de texturas.

3.2.2.1. Matrices de co-ocurrencia de niveles de gris (GLCM)

Para extraer las características de Haralick, el algoritmo de detección de defectos construye una GLMC de una imagen de textura $I(x, y)$. La GLMC estima la probabilidad conjunta de que un par de valores de pixel ocurran en un vector de desplazamiento \vec{d} unos de otro. Asumiendo que una imagen de textura $I(x, y)$ es una matriz consistente $N \times M$ de G diferentes tonos de gris, la GLMC por su vector de desplazamiento $\vec{d} = (d_x, d_y)$ es una matriz de $G \times G$.

$$P_{\vec{d}}(i, j) = \sum_{x=1}^N \sum_{y=1}^M \partial \{I(x, y) = i \wedge I(x + d_x, y + d_y) = j\}$$

Donde:

$$\partial \{\text{Verdadero}\} = 1$$

$$\partial \{\text{Falso}\} = 0$$

El numero en el elemento (i, j) de la matriz GLCM $P_{\vec{d}}(i, j)$ cuenta el número de veces que el pixel con valor i ocurrió \vec{d} lejos de un pixel con valor j. Las siguiente figura ilustran una GLCM para una muestra de textura de 4×4 pixeles $I(x, y)$ con 8 valores de grises. La figura (3.5) ilustra la correspondiente GLCM para el vector de desplazamiento $\vec{d} = (0, 1)$. La GLCM es una matriz de 8×8 $P(i, j)$ que representa el numero de veces que un valor de pixel j ocurre a la derecha de un valor de pixel i. Por ejemplo, el valor de pixel 2 esta doblemente situada a la derecha del pixel

⁶Para mas información a cerca de el papel de estadística de segundo orden en percepción humana ver [81]

⁷Para mas información a cerca de las características de Haralick ver [79]

de valor 1 en la textura muestra. Así, $P(0,1)(1,2) = 2$. Similarmente, $P(0,1)(8,1) = 1$ porque solamente una vez el pixel de valor 1 se produce a la derecha del pixel de valor 8.

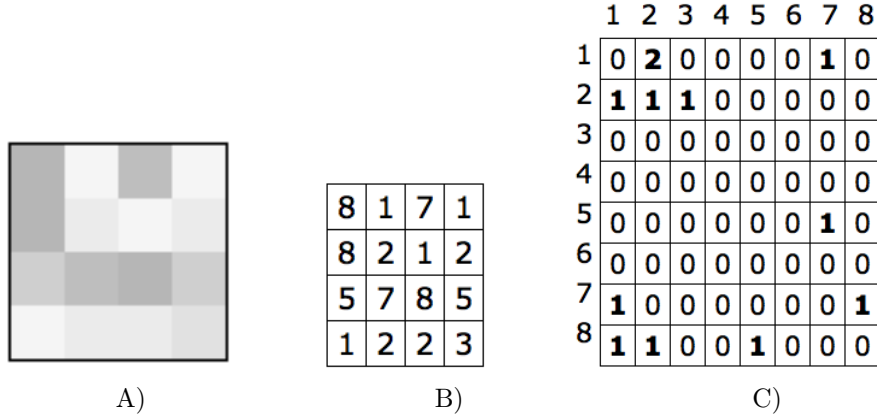


Figura 3.5: A) Textura muestra. B) Valores de los pixeles para la textura muestra. C) GLCM para el vector de desplazamiento. *Tomado de *NI Vision Concepts Help*[15]

En aplicaciones como clasificación de texturas, la matriz de co-ocurrencia se puede extraer de la textura completa. En la detección de defectos en texturas, es mejor extraer la matriz de co-ocurrencia de las características locales. La matriz de co-ocurrencia se puede extraer localmente ya sea mediante la separación de la textura en ventanas adyacentes y calculando la GLCM para cada ventana o moviendo una sola ventana sobre la textura y calculando una GLCM que es asociada con el pixel central en cada caso de la ventana. Por *default*, el algoritmo de detección de defectos en texturas calcula la matrices de co-ocurrencia localmente de la descomposición de las imágenes en sub-bandas de *wavelet* utilizando ventanas adyacentes de 15×15 ; Sin embargo se puede modificar la medida y la superposición de las ventanas.

Un parámetro importante en el calculo de la GLCM es el vector de desplazamiento (\vec{d}), porque una textura visual constante resulta en otra matriz de co-ocurrencia para cada vector de desplazamiento (\vec{d}).

3.2.2.2. Extracción de las características de Haralick

El algoritmo de detección de defectos extrae cinco características de Haralick (entropía, disimilitud, contraste, homogeneidad y correlación) de la GLCM calculada en cada partición de la sub-banda de la textura.

$$Entropía = \sum_{i=1}^G \sum_{j=1}^G P_{i,j} (-\ln P_{i,j})$$

$$Disimilitud = \sum_{i=1}^G \sum_{j=1}^G P_{i,j} |i - j|^2$$

$$Contraste = \sum_{i=1}^G \sum_{j=1}^G P_{i,j} (i - j)^2$$

$$Homogeneidad = \sum_{i=1}^G \sum_{j=1}^G \frac{P_{i,j}}{1 + (i - j)^2}$$

$$Correlación = \sum_{i=1}^G \sum_{j=1}^G P_{i,j} \left[\frac{(i - \mu_i)(j - \mu_j)}{\sqrt{\sigma_i^2 \sigma_j^2}} \right]$$

Donde:

$$\mu_i = \sum_{i,j=1}^G i P_{i,j} \quad y \quad \mu_j = \sum_{i,j=1}^G j P_{i,j} \text{ es la media de GLCM}$$

y

$$\sigma_i = \sum_{i,j=1}^G P_{i,j}(1 - \mu_i)^2 \text{ y } \sigma_j = \sum_{i,j=1}^G P_{i,j}(1 - \mu_j)^2 \text{ son la varianza de la GLCM}$$

Las siguientes figuras ilustran la sub-banda *low-high* y sus cinco correspondientes mapas de características de Haralick. Nótese que las características de Haralick claramente distinguen el defecto en la textura, con la característica de entropía y contraste exhibiendo la más alta diferenciación para este ejemplo.

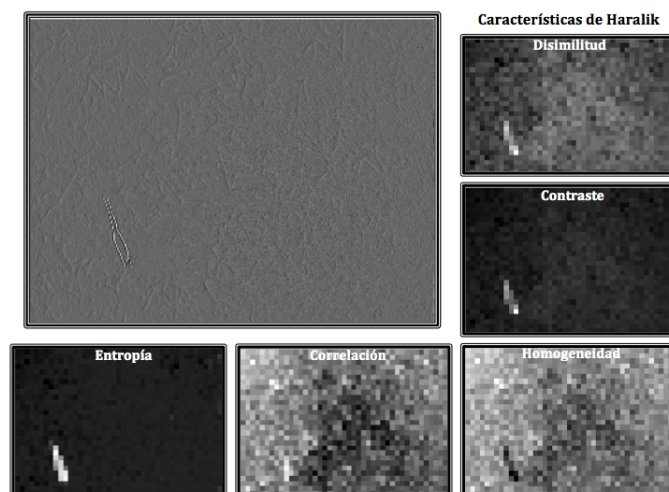


Figura 3.6: Ejemplo de las características de Haralick para la sub-banda *low-high*. *Elaboración propia (Julio-2012)

El numero de elementos del vector característica extraídos para representar una textura muestra es equivalente a el número de bandas de *wavelet* seleccionados multiplicado por cinco. Por ejemplo, si una aplicación utiliza todas las 8 sub-bandas, la tamaño resultante del vector de características es 40.

3.3. Clasificador de máquinas de vectores de soporte

El problema de clasificación en este trabajo es un ejemplo clásico de lo que en la comunidad de reconocimiento de patrones se conoce como "detección atípica". En este sentido, una de las clases (la clase normal) es apoyada con numerosas muestras de entrenamiento mientras que la otra clase (la clase defecto) esta representada con muy pocas o ninguna muestra. Por ejemplo, los defectos que aparecen en productos manufacturados con textura natural tiene una combinación infinita de formas, tamaños y orientaciones. Con el fin de hacer frente a este problema, una clase de clasificadores aprenden la descripción de la normalidad solo con la observación de muestras libre de defectos [82].

En este proyecto, una clase SVM que se argumenta es uno de los mejores detectores de atípicos es utilizado para aprender y generalizar texturas normales a partir de muestras libres de defectos [83].

En los SVM han sido incluidos recientemente dos clases de clasificadores de márgenes máximos que se han hecho muy populares debido a su buen desempeño en "altos espacios de características

dimensionales”, evitando el sobreajuste, y teniendo capacidades de generalización muy buena [84]. Los SVM clasifican los casos buscando la superficie de separación que tiene la máxima distancia a los datos de puntos más próximos en ambos lados. Estos datos de puntos más cercanos son llamados los ”vectores de soporte” y las distancias de la superficie de separación para la instancia más cercana es llamada el ”margen”. La figura (3.8) muestra un ejemplo de un escenario linealmente separable donde la clase perteneciente a los datos de puntos x_1, x_2, \dots, x_N es presentado por variable objetivo $t_n \in \{-1, 1\}$. Dos clasificadores candidatos lineales se muestran en esta figura en líneas sólidas junto con sus correspondientes márgenes y vectores de soporte. La frontera de clasificación en el panel izquierdo tiene escaso margen, mientras que el de la derecha es el clasificador de margen lineal máximo para este ejemplo.

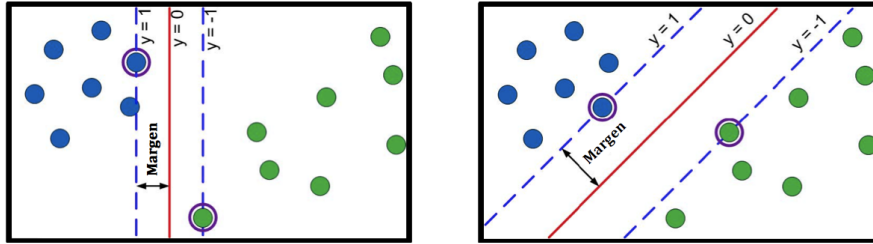


Figura 3.7: Ejemplo de un SVM

En general, un clasificador lineal en un espacio de característica dimensional p es un $p - 1$ hiperplano bidimensional $w^T \cdot x + b = 0$ donde w es el vector normal y $\frac{b}{\|w\|}$ es la compensación desde el origen. El margen de cada clasificador candidato puede ser visualizados moviendo dos hiperplanos paralelos (lineas en trazos en nuestro ejemplo) en direcciones opuestas hasta que lleguen a los datos de puntos más cercanos. Los hiperplanos que pasan a través de los vectores de soporte pueden ser descrito en su forma canónica después de hacer ponderaciones isométricas $w \rightarrow \kappa w$ y $b \rightarrow \kappa b$ por la ecuaciones.

$$w^T \cdot x + b = 1$$

$$w^T \cdot x + b = -1$$

Utilizando geometría simple el margen es simplemente $\frac{2}{\|w\|}$. Por lo tanto en el caso separable, el objetivo es reducir al mínimo $\|w\|$ tal que ningún punto de datos se encuentre entre los hiperplanos o equivalentemente.

$$t_n(w^T \cdot x_n + b) \geq 1 \quad \text{para toda } 1 \leq n \leq N$$

Este problema de optimización es un problema de programación cuadrático y el vector normal óptimo w se muestra que tiene la forma.

$$w = \sum_n \alpha_n t_n x_n$$

Donde los coeficientes α_n son las soluciones del problema de programación cuadrático. Ahora, los nuevos datos de puntos x se clasifican en función de

$$y(x) = (w^T \cdot x + b) \geq 1 - \xi_n \quad \text{para toda } 1 \leq n \leq N$$

$$\xi_n \geq 0, \forall n$$

Donde λ es un parámetro de margen de penalización suave determinado por el usuario. Los conceptos anteriores se extendieron para más decisiones de frontera complejas no lineales por Aizerman et al [85]. Una transformada no lineal es utilizada para mapear datos de puntos en un alto espacio dimensional donde un clasificador lineal es posteriormente utilizado para clasificar

datos de puntos que son separables por un hiperplano en el espacio de características nuevas. Así, el límite de decisión en el espacio original es no lineal en el espacio original.

Una clase SVM busca una superficie de decisión con el volumen mínimo que encierra la mayoría de los ejemplos agrupados en el espacio de características y permite una fracción, v , de muestras entrenadas para la clase normal para quedar fuera del límite de normalidad. Este error de entrenamiento previsto deja espacio para la posible presencia de valores atípicos en el conjunto de entrenamiento. El SVM uniclase es formulado para encontrar el hiperplano óptimo que separa una fracción deseada $1-v$ de los patrones de entrenamiento desde el origen del espacio de características.

$$\min_{w, \xi_n, b} \left(\frac{1}{2} \|w\|^2 - b + \frac{1}{vN} \sum_n \xi_n \right)$$

Donde v es la fracción de los patrones de entrenamiento que se les permite ser rechazada y N es el número total de muestras de entrenamiento.

3.3.1. Máquinas de vectores de soporte

Una máquina de vector de soporte (SVM) es un método de aprendizaje supervisado que generaliza un gran conjunto de muestras entrenadas en un menor número de vectores de soporte para predecir la clase de muestras desconocidas.

Un clasificador SVM es matemáticamente más complejo que un clasificador basado en la distancia. Sin embargo un clasificador SVM tiene mejores capacidades de generalización que un clasificador basado en la distancia, y es más rápido cuando el conjunto de muestra es grande porque el clasificador opera solo en los vectores de soporte.

Generalmente un clasificador SVM se utiliza en los siguientes tipos de aplicaciones:

- Cuando la aplicación tiene una clase de muestras buenas pero un número indeterminado de clases para las muestras malas. Un ejemplo de este tipo de aplicación es la detección de defectos. Para este tipo de aplicación, se utiliza un clasificador SVM *one-class* para entrenar las muestras de la clase en buen estado. Muestras, tales como defectos, que no pueden ser clasificados como la clase conocida son clasificados como desconocido.
- La aplicación requiere un número grande de muestras para el entrenamiento. Durante el entrenamiento, el clasificador SVM identifica los vectores de soporte para las muestras de entrenamiento. En la clasificación el clasificador operará solo en los vectores de soporte, lo cual reducirá el tiempo requerido para la clasificación.

El algoritmo SVM construye un modelo para clasificar las muestras. Este modelo representa las muestras en un espacio multidimensional donde las muestras son separadas por la distancia máxima posible. Por ejemplo, la Figura(3.8).

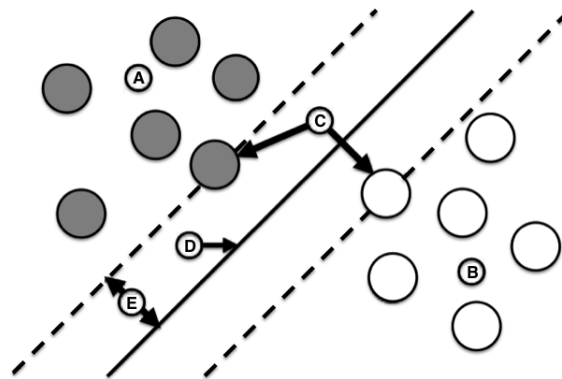


Figura 3.8: Se muestra una aplicación que involucra dos clases linealmente separables representadas en un espacio bidimensional. A) Muestras de clase 1 B) Muestras de clase 2 C) Vectores de soporte D) Hiperplano E) Margen. *Elaboración propia (Julio-2012)

El algoritmo utiliza una función cuadrática para establecer los vectores de soporte para cada clase. Un vector de soporte es una muestra en una clase que esta más cercana a otra clase. El algoritmo SVM identifica un hiperplano que separa los vectores de soporte de cada clase. La distancia entre el vector de soporte y el hiperplano es llamado margen. El algoritmo SVM selecciona una superficie de soporte que produce el mayor margen posible para cada vector de soporte.

3.3.1.1. Entrenamiento

Cuando se entrena al clasificador SVM, el algoritmo utiliza un proceso iterativo para optimizar la función de vector de soporte. El entrenamiento se termina cuando el gradiente de la función de optimización es menor o igual a la tolerancia. Un valor de tolerancia que sea demasiado alto puede causar que el algoritmo SVM termine el entrenamiento antes de que la función de vector de soporte este suficientemente optimizado. Un valor de tolerancia que sea demasiado bajo hará que el algoritmo trate de lograr un nivel muy alto de optimización, consumiendo demasiado tiempo y resultando computacionalmente costoso.

3.3.1.2. Clasificación

Cuando se utiliza el clasificador SVM, el algoritmo determina la clase de una muestra desconocida comparando esta con los vectores de soporte de las muestras entrenadas. El algoritmo utiliza la siguiente formula para clasificar una muestra desconocida x :

$$\text{sgn}\left(\sum y_i a_i k_i(x_i, x) + b\right)$$

Donde y_i es la clase de asociación (-1 o 1)

a_i es el coeficiente de peso

k es la función *kernel*

x_i es el numero de vectores de soporte

b es la distancia de los hiperplanos al origen

La velocidad de clasificación dependerá del número de vectores de soporte y la selección de la función de *kernel*. El coeficiente de peso a_i (que es una salida de la función de vectores de soporte optimizado) determina el número de vectores de soporte. Si el coeficiente de peso de una muestra no es igual a 0, la muestra es una vector de apoyo.

3.3.1.3. SVM multiclase

La clasificación SVM generalmente consta de dos clases. Para aplicaciones que requieren más de dos clases, el algoritmo SVM utiliza un enfoque de "uno-contra-uno". En un enfoque de "uno-contra-uno", el algoritmo crea un modelo de clasificación binaria para cada posible combinación de clases, de modo que n numero de clases produce $n \times (n - 1)/2$ modelos de clasificación. Durante la clasificación, el algoritmo utiliza un mecanismo de votación para identificar la mejor clase. Si el mecanismo de votación identifica varias clases, el algoritmo selecciona la clase que esta mas cerca de la muestra.

3.3.1.4. Modelos

La siguiente sección describe los modelos que el algoritmo SVM pueden utilizar para clasificar las muestras. Para aplicaciones que requieran una sola clase, tal como en la detección de defectos en texturas, se selecciona el modelo de una sola clase. Para aplicaciones que requieran múltiples clases, se seleccionan los modelos C-SVC o nu-SVC.

C-SVC

El modelo C-SVC permite al algoritmo SVM separar claramente las muestras por un margen muy estrecho. El entrenamiento consistente en minimizar la función de error:

$$\min_{w, b, \xi} \frac{1}{2} W^T W + C \sum_{i=1}^l \xi_i$$

Sujeto a $Y_i(W^T K(X_i) + b) \geq 1 - \xi_i; \xi_i \geq 0, i = 1 \dots l; \rho \geq 0$
 Donde W es el vector normal del hiperplano al origen
 C es el parámetro de costo
 y ξ es la variable *slack*

Si el algoritmo SVM no puede definir un margen claro, entonces se utiliza el parámetro de costo para permitir algunos errores de entrenamiento y producir un margen suave. Si el valor de costo es demasiado alto se prohíbe errores de entrenamiento, generando un margen estrecho y una clasificación rígida.

Nu-SVC

En el modelo Nu-SVC, el parámetro *nu* controla los errores de entrenamiento y el número de vectores de soporte. El entrenamiento implica minimizar la función de error.

$$\min_{w, b, \xi} \frac{1}{2} W^T W - \nu \rho + \frac{1}{l} \sum_{i=1}^l \xi_i$$

Sujeto a $Y_i(W^T K(X_i) + b) \geq \rho - \xi_i; \xi_i \geq 0, i = 1 \dots l; \rho \geq 0$
 Donde W es el vector normal del hiperplano al origen
 ν es el parámetro *nu*
 y ξ es la variable *slack*

El valor *nu* especifica tanto la relación máxima de errores de entrenamiento y el mínimo número de vectores de soporte en relación con el número de muestras. *Nu* debe ser mayor que 0 y no puede exceder 1. Un Alto valor de *nu* aumenta la tolerancia para la variación en la textura, pero también puede aumentar la tolerancia para de los defectos en la textura. Si *nu* es demasiado alta, el entrenamiento produce demasiados errores de entrenamiento para ser útiles.

One-Class SVM

En el modelo de una clase, el algoritmo SVM considera que la distribución de información espacial de cada muestra para determinar si la muestra pertenece a la clase conocida.

$$\min_{w, b, \xi} \frac{1}{2} W^T W - \rho + \frac{1}{vl} \sum_{i=1}^l \xi_i$$

Sujeto a $W^T K(X_i) \geq \rho - \xi_i; \xi_i \geq 0, i = 1 \dots l; \rho \geq 0$
 Donde W es el vector normal del hiperplano al origen
 ν es el parámetro *nu*
 y ξ es la variable *slack*

3.3.1.5. Kernels

Un clasificador SVM es un clasificador lineal. Típicamente, un clasificador SVM utiliza una *kernel* lineal, que es el producto del vector de características muestras multiplicador por el vector de soporte de la muestra. Un clasificador SVM puede utilizar los siguientes *kernel* no lineales.

Polinomial	$(Gama \times Kernel(x_i, x) + coeficiente)^{grado}$
Función base radial (RBF)	$e^{-gama \ x_i - x\ ^2}$
Gauss	$e^{-\frac{\ x_i - x\ ^2}{2 \times sigma^2}}$

Se utiliza un *kernel* no lineal para transformar muestras con información de características no lineales a una dimensión donde la información característica es linealmente separable, como se ilustra en la Figura (3.9).

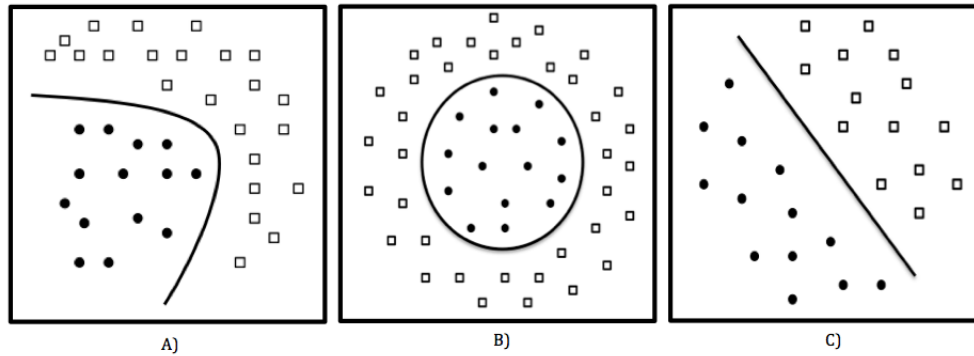


Figura 3.9: La Figura A) ilustra como un kernel polinomial separa información de características no lineales. La Figura B) ilustra como un kernel RBF separa la información de características no lineales. La Figura C) muestra la información de características no lineales claramente divisible obtenido después de usar un kernel no lineal para transformar la muestra a una una dimensión donde la información de características son linealmente separables. *Elaboración propia

3.3.1.6. Elección de los parámetros correctos dentro del entrenador clasificador (*NI Texture Training Interface*)

La siguiente lista proporciona información para seleccionar adecuadamente los parámetros de SVM para la aplicación en este proyecto.

- Modelo: si la aplicación utiliza una solo clase, se utiliza el modelo *one-class*. Si la aplicación involucra más de una clase, siempre se comienza con el modelo nu-SVC.
- Tolerancia: especifica el gradiente máximo de la función cuadrada utilizada para calcular los vectores de soporte. El entrenamiento se finaliza cuando el gradiente de la función de optimización es menor o igual al valor de la tolerancia. El valor por defecto es 0.001. Por lo general no es necesario cambiar este valor.
- Nu: especifica tanto la relación máxima de errores de entrenamiento y el número mínimo de vectores de soporte en relación con el numero de muestras. Los valores deben de ser mayores que 0 y no pueden exceder 1. El valor predeterminado es 0.1. Un valor mayor de *nu* incrementa la tolerancia para la variación en la textura, pero también puede aumentar la tolerancia de los defectos en la textura. Si el clasificador de texturas no funciona como se espera debido a que las muestras de texturas entrenadas no representan todas las posibles variaciones de la textura, se prueba aumentando el valor de *nu*.
- Costo: especifica la pena por errores de entrenamiento. Si el valor de costo es demasiado alto se prohíbe errores de entrenamiento, produciendo un margen estrecho y una clasificación rígida. Disminuir el valor de costo para permitirá más errores de entrenamiento y obtener un margen más suave entre las clases.
- *Kernel*: especifica el núcleo que utiliza el clasificador. RBF es el valor predeterminado. En general, no es necesario modificar esta configuración. Si el número de características de la muestra es alto, se prueba el núcleo lineal.
- Grado: especifica el grado del núcleo polinomial. En general, seleccione un valor inferior a 10.
- Gama: especifica el valor de gama para los núcleos polinomiales y RBF. Un valor alto requiere más vectores de soporte para clasificar las muestras. Utiliza un alto valor para muestras con información de características distribuidas regularmente, y un bajo valor para las muestras con distribución irregular en la información de características. Es posible que se necesite cambiar este valor para apoyar los valores seleccionados del coste o *nu*. Por ejemplo, si se especifica un alto valor de *nu* (que eleva el numero mínimo de vectores de soporte) también puede ser necesario aumenta el valor de gama.

Si se utiliza un clasificador personalizado, se especifica un valor del vector de características para el clasificador personalizado que sea mayor que 0 pero menor que 1. La escala del vector de características reduce los problemas de desbordamiento y mejora la tasa de clasificación.

Capítulo 4

Diseño Mecatrónico

4.1. Especificaciones del sistema

En esta apartado, se comentará en detalle en que consiste el sistema de visión artificial que se ha implementado. A grandes bloques el sistema de visión se puede desglosar en las siguientes partes:

- Piel curtida de bovino a examinar.
- Cámara de vídeo y óptica.
- Módulo de adquisición de imagen de un solo canal (monocromático).
- Sistema de iluminación.
- Manipulador cartesiano.
- Hardware de control.
- Software incorporado.

Una vez enumerados los puntos de los que consta el proyecto, se empezará por describir el proceso en el cual se realizará la inspección de la piel curtida de bovino.

4.1.1. Descripción del proceso

Se trata de desarrollar un sistema de visión artificial para la detección de defectos en pieles curtidas de bovino, donde el proceso de análisis y detección de defectos se desarrollará de forma automática. Este tendrá un sistema de iluminación, así como un gabinete para aislar al sistema de cualquier posible ruido que pueda afectar las pruebas experimentales.

Se utiliza un manipulador cartesiano con tres grados de libertad para escanear la piel curtida. El área de trabajo en el plano xy fue dividido en una cuadrícula de 5×7 cuadros con el objetivo de obtener imágenes más detalladas, cada uno de estos tiene una dimensión de $42[mm]$ ancho por $56[mm]$ de largo para cumplir con el área necesaria para detectar los defectos mínimos que se pueden encontrar en la piel curtida de bovino (Figura 4.1).

Primeramente, se pondrá la piel curtida manualmente en el espacio de trabajo del manipulador cartesiano. Al inicial el análisis, se verificará que el manipulador se encuentre en su posición de *home*, si no es así este tendrá que llegar al mismo. Cuando este en su posición de inicio se tomará una foto de la correspondiente área, esta será analizada y se señalará por medio del monitor si la porción tiene o no tiene defecto, la imagen se guardará en una carpeta y se borrará su espacio de la memoria RAM, se moverá el manipulador para llegar a su segunda posición, al llegar tomará y analizará la siguiente muestra, y así sucesivamente siguiendo un recorrido establecido (Figura 4.2). Al terminar el recorrido, la fotos guardadas de cada una de las posiciones se unirán en una sola imagen para desplegar en pantalla una imagen general de la piel curtida de bovino puesta en el área del manipulador, resaltando con rojo los defectos encontrados en ella.

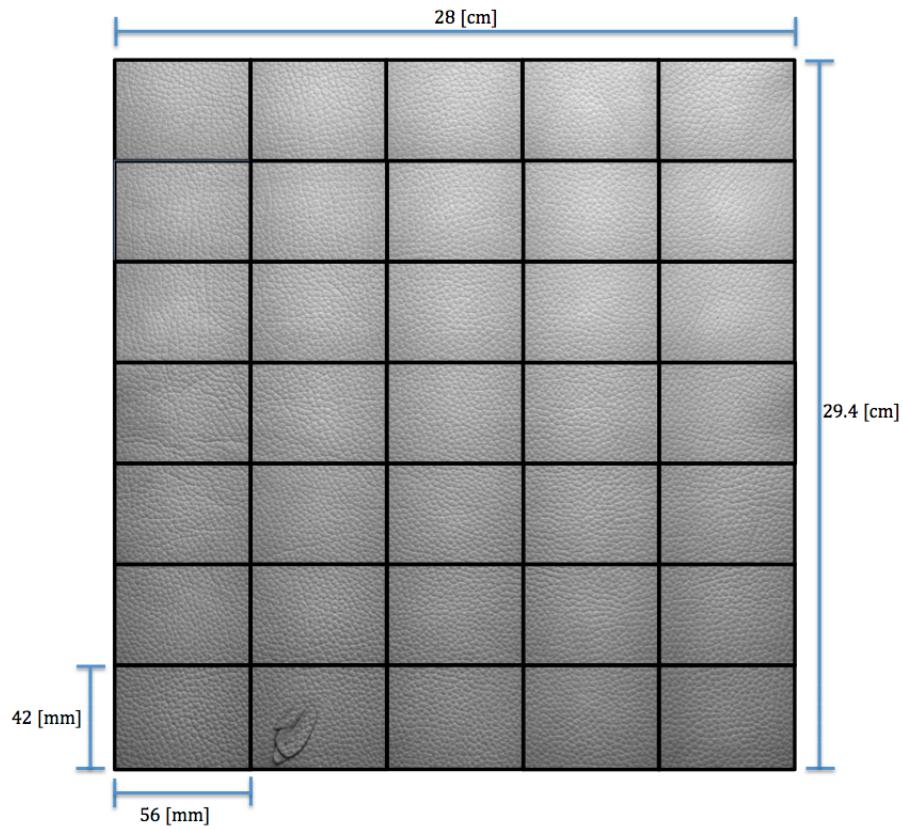


Figura 4.1: División del espacio de trabajo del manipulador. *Elaboración propia (Julio-2012)

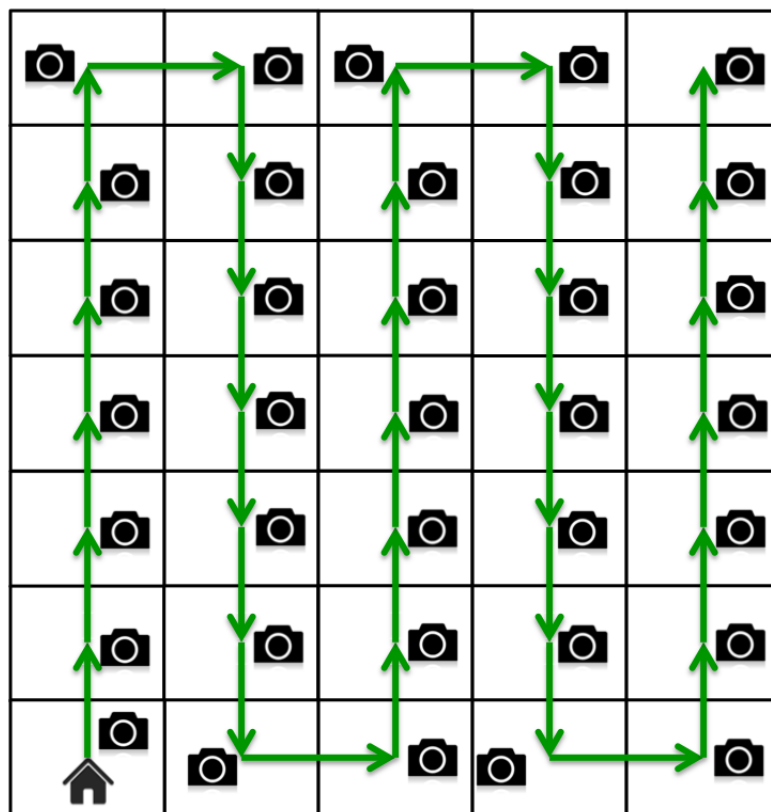


Figura 4.2: Recorrido de la cámara en el espacio de trabajo del manipulador. *Elaboración propia (Julio-2012)

La cámara estará posicionada en el eje z del manipulador, el grado de libertad en este eje solo se utilizará para ajustar la distancia de trabajo a la cual tiene que encontrarse la cámara de su objetivo. La iluminación en la parte experimental es controlada por medio de un gabinete que aísla la luz externa del interior mediante paredes opacas que impiden la entrada de luz del exterior. La luz es implementada en forma directa, esta fue fijada en el eje z junto con la cámara para tener una iluminación homogénea en todas las tomas de cada espacio, manteniendo la misma iluminación no importando la posición en la que se encuentre el manipulador.

La iluminación y la distancia entre la cámara y la piel curtida son dos factores importantes en el experimento. Evidentemente, variaciones significativas en la iluminación conducirá a resultados falsos. La distancia entre la cámara y la piel curtida se definió a través de los requerimientos de dominio. Se ha observado que una distancia demasiado lejos falla en extraer satisfactoriamente características estadísticas de los defectos para resultados de clasificación buenos. Por otro lado, una distancia demasiado cercana aumenta la carga de procesamiento debido a grandes cantidades de datos a procesar en la misma área de la superficie de cuero. En este proyecto, una distancia fija se mantuvo a partir de la lente de la cámara a el espacio del trabajo del manipulador durante todo el experimento.

Cabe destacar que en general, la aplicación de visión se realiza con respecto a la piel curtida de bovino tipo Camelia Oyster. Por lo tanto, la flexibilidad de la aplicación es bastante limitada. En el caso de que se llegara a cambiar el tipo de piel, el sistema no será capaz de adaptarse a los cambios realizados. Si esto sucede, solo basta con entrenar y calibrar el sistema de iluminación para adaptarlo a esta nueva piel para así detectar los defectos en dicha piel.

4.1.2. Objeto a examinar

El objeto a inspeccionar es una conjunto de muestras de piel curtida de bovino del tipo Camelia Oyster de $28 \times 29.4 [cm^2]$. Como se mencionó anteriormente cada porción analizada de esta será de $42 [mm]$ ancho por $56 [mm]$ de largo (Figura 4.3). Este como un material natural tiene una variedad visual no homogénea en color, grosor, brillo, etc.

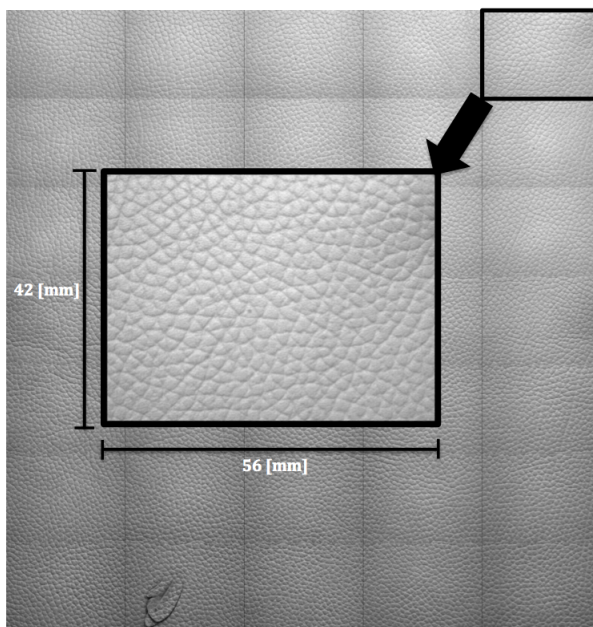


Figura 4.3: Porción de piel curtida de bovino a analizar. *Elaboración propia (Julio-2012)

La piel que se analiza, en particular, se caracteriza por la aparición de defectos que en su mayoría son provocados por:

1. Heridas durante su fase productiva (por ejemplo, cortadas, lucha con otros machos, marcas de hierro caliente, infecciones, entre otros).
2. Exposición del ganado a los ectoparásitos y a un manejo inadecuado.
3. Desarrollo de problemas durante el transporte y fase de conservación.

En la Tabla (4.1), se da una definición de la aparición de cada tipo de defecto. También se enlistan ejemplos de la marca o cicatriz de diferentes tipos de defectos en piel curtida. Los defectos que se encuentran debajo de la piel e invisibles sobre la superficie de la piel curtida no están incluidos en este trabajo de investigación.

Tipo de defecto	Apariencia y definición del defecto	Marca o cicatriz del defecto en la piel curtida
1. Puntos delgados	Puntos como cicatrices con un diámetro de ≤ 1 [mm]	Raíz del pelo, agujero de alfiler, punto putrefacto, dermatitis.
2. Puntos	Puntos como cicatrices con un diámetro de > 1 [mm]	Arañazos de aguijón, marca de uñas, efecto ojo de pez, mancha de cromo, mancha de sal.
3. Líneas	Una cicatriz tipo línea con una anchura de ≤ 1 [mm]	Venas, pliegues de grasa, grietas.
4. Tiras	Una cicatriz tipo línea con una anchura de >1 mm	Arrugas de cuello, marca de hierro.
5. Agujeros	Orificio	Marcas de bala, agujeros de larva.
6. Patrones	Marca de cria animal	Marca de hierro.
7. Irregulares	Apariciones no regulares como en cualquiera de los tipos anteriores o varios tipos de defectos mezclados	Verrugas, variación de tonos, rasguños, signo de rozadura, manchas de parásitos, mancha de estiércol.

tabla 4.1: Siete tipos de defectos en pieles curtidas de bovino. Tomado de Yeh y Perng [16].

Los defectos que se encuentran en la piel curtida aparecen en diferentes formas. Varían desde puntos y defectos de líneas hasta fallas mayores las cuales recorren la longitud de una mayor área en la imagen. En la Figura (4.4) se ejemplifican los diferentes tipos de defectos encontrados en la piel curtida de bovino a examinar.

4.1.3. Adquisición de imágenes

El conjunto cámara-óptica, hardware NI PXI-1411 de captura de imágenes y el NI-IMAQ forman el sistema de adquisición de imágenes en la aplicación de visión artificial basada en computadora (Figura 4.5).

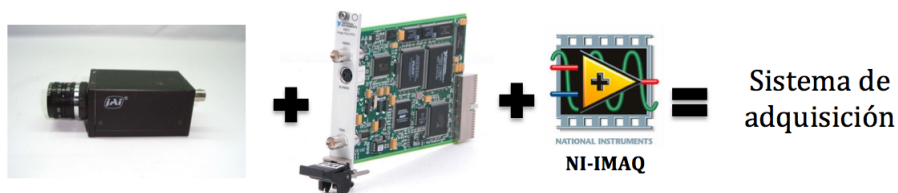


Figura 4.5: Sistema de adquisición de imágenes. *Elaboración propia (Julio-2012)

En este apartado, se explicará que tipo de cámara se ha utilizado para la aplicación en estudio. El tipo de cámara es transcendental para captar una imagen que refleje características de la superficie, por lo cual el éxito del proyecto dependerá de esta.

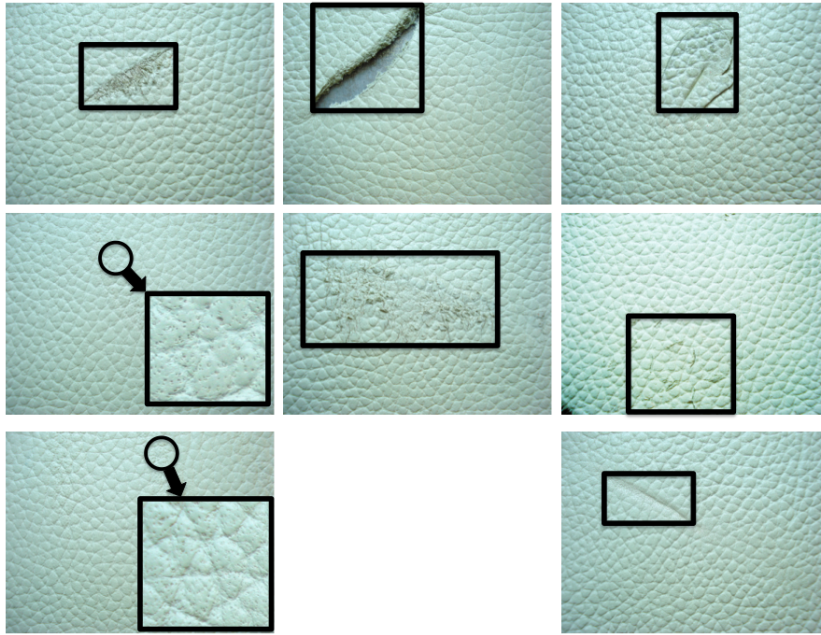


Figura 4.4: Diferentes tipos de defectos en la piel curtida de bovino. *Elaboración propia (Julio-2012)

4.1.3.1. Cámara y óptica

Cuando se habla de una cámara se hace referencia a un sofisticado transductor que convierte una escena en una señal eléctrica de vídeo, que posteriormente con los restantes elementos del sistema de adquisición y un PC se procesa y analiza. En el mercado existen dos tipos básicos (de acuerdo con el tipo de señal que proporcionan): analógicas y digitales. En nuestro caso la cámara que se ha utilizado es del tipo analógico, estas son cámaras que proporcionan una señal de vídeo analógica que se digitaliza posteriormente mediante la tarjeta de adquisición de imágenes.

Típicamente, las cámaras utilizan un *array CCD* (*Charge Coupled Device*) que se ha convertido en el estándar tecnológico en las cámaras. Este tipo de cámaras cumple con una serie de características ventajosas respecto a otro tipo de cámaras. En primer lugar, es fácil configurarlas y son simples a la hora de manejarlas. Además, aportan una alta resolución, son resistentes y pequeñas y de bajo costo. Su consumo de potencia es también bajo, por lo que son aconsejables para aplicaciones con mucha carga de trabajo.

Aunque la relación señal-ruido es mayor que en una digital (donde la digitalización se hace a nivel de la propia cámara antes de enviarla a la placa), las cámaras analógicas son más baratas, con estándares fáciles de configurar y además pueden resolver numerosas aplicaciones a un precio atractivo.

La cámara utilizada en este proyecto es una JAI[®] modelo CV-M50 E de alto rendimiento con un encapsulado compacto y que utiliza un lente marca *Lens* de 12 [mm] adecuada para aplicaciones industriales tales como inspección y medida. Se trata de una cámara con una matriz CCD de 640X480 *pixels* monocromático que posee un ancho de banda de 3dB la cual ocupa la mayor parte del espectro visible, que va desde los 400 a los 780 nm. (Figura 4.6).

Las características principales de este dispositivo son:

- *Shutter*: Controla el tiempo de exposición del sensor CCD. Los valores que se pueden adoptar son 1/50, 1/100, 1/250, 1/500, 1/1000, 1/2000, 1/4500 y 1/10000 seg. La forma de configurar este parámetro es mediante unos microrruptores instalados en la propia cámara.
- *CCD Accumulation*: Es el modo de acumulación del dispositivo, existen dos modos diferentes:

- *Field Accumulation*: Es útil para filmar objetos en movimiento.
- *Frame Accumulation*: Es más efectivo para imágenes estáticas.

Las mediciones a realizar serán totalmente estáticas, por eso se aplicará el *Frame Accumulation*.

- *Scanning System*: Se emplea para elegir el sistema de escáner entre:
 - No entrelazado, para imágenes en movimiento.
 - 2:1 Entrelazado, para imágenes estáticas.

En el caso que se estudia se tomará el modo 2:1 entrelazado pues permite operar de acuerdo con los sistemas estándar de TV, EIA o CCIR.

- *Gamma Correction*: Es el factor de compensación de *gamma*, hay dos valores:
 - 1.0 *liniar*
 - 0.45 *correction*

Debe emplearse la posición 1.0 *liniar* ya que es la recomendada para las aplicaciones de procesado de imagen o de visión directa.

- *Auto Gain Control*: Corresponde a la forma de actuar de la ganancia, la cual puede ser manual o automática. En posición de Auto, este parámetro se ajusta de una forma autónoma según sean las características de la imagen. Por otro lado, el control manual hace que la ganancia se constante independientemente del tipo de imagen. La ganancia se seleccionó de forma manual y ajustada por ensayo y error.



Figura 4.6: Cámara CCD JAI CV-M50 E. *Tomado de http://www.pixequip.com/product_images/18550_jai_cv-m50_01.png

Las características técnicas son las siguientes:

Specifications	CV-M50C	CV-M50E
Scanning system	625 lines 25 frames/sec.	525 lines 30 frames/sec.
CCD sensor	Monochrome 1/2" Hyper HAD IT CCD	
Sensing area	6.6 mm (h) x 4.8 mm (v)	
Picture elements effective	752 (h) x 582 (v)	768 (h) x 494 (v)
Elements in video out	737 (h) x 575 (v)	758 (h) x 486 (v)
Cell size	8.6 (h) x 8.3 (v) μm	8.4 (h) x 9.8 (v) μm
Resolution (horizontal)	560 TV lines	570 TV lines
Resolution (vertical)	575 TV lines	485 TV lines
Sensitivity	0.5 Lux, F1.4	
S/N ratio	>56 dB (AGC off, Gamma 1)	
Video output	Composite VBS signal 1.0 Vpp, 75 Ohm	
Gamma	0.45 - 1.0	
Gain	Manual - Automatic Potmeter - AGC	
Scanning	2:1 interlace - non-interlace	
Accumulation	Field - frame	

tabla 4.2: Especificaciones de la cámara CCD. Véase Apéndice B

Configuración del sistema de imagen

Para la implementación del sistema de captación, es necesario establecer los requerimientos denominados: requerimientos de dominio; mediante los cuales son establecidos, parámetros que permiten determinar el entorno de la imagen a obtener y procesar.

Los requerimientos de dominio a considerar son:

1. Campo de visión(FOV): Es el área del objeto que se desea capturar y que la cámara puede adquirir con la mayor resolución. En el caso de este proyecto al dividir el espacio de trabajo del manipulador en 5 cuadros de ancho por 7 de largo se tiene que cada cuadro tiene una dimensión de 56×42 [mm], por lo tanto este será el área que se tomará para cada muestra.
2. Distancia de Trabajo: Que es la distancia desde la óptica de la cámara hasta el objeto a inspeccionar.
3. Tamaño del sensor: El tamaño de un área activa del sensor, típicamente definida por la dimensión horizontal del sensor (mismas que se encuentran en las especificaciones de la cámara a utilizar, véase Apéndice B).
4. Resolución: Es una medida de la capacidad del sistema de visión de reproducir los detalles de un objeto. Esta viene definida por las características de la cámara y el lente a utilizar
5. Longitud focal: medida en milímetros, es la distancia desde el centro de un lente al sensor. Esta definida por el lente y como se mencionó anteriormente es de 12 [mm].

Distancia de trabajo

Los cálculos de la distancia de trabajo se basan en la siguiente ecuación.

$$\text{Longitud focal} \times \text{FOV} = \text{tamaño del sensor} \times \text{distancia trabajo} \quad (4.1)$$

despejando de la ecuación (4.1) la distancia de trabajo.

$$\text{distancia trabajo} = \frac{\text{longitud focal} \times \text{FOV}}{\text{tamaño del sensor}} \quad (4.2)$$

Para nuestro caso se tiene que:

- FOV= 56×42 [mm]
- Tamaño del sensor= 6.4×4.8 [mm]
- Longitud focal= 12 [mm]

Sustituyendo estos valores en la ecuación (4.2)

$$\text{distancia trabajo} = \frac{12[\text{mm}] \times 56[\text{mm}]}{6.4[\text{mm}]} = 105 [\text{mm}] = 10.5 [\text{cm}]$$

Una vez determinado la distancia de trabajo, se procede a implementarla en el sistema. La Tabla(4.3) muestra los requerimientos que se han de considerar para la implementación del sistema óptico.

Requerimientos de dominio	
Requerimiento	Parámetro
Resolución	640×480 pixel <i>resolution</i>
Campo de visión (<i>Field of view</i> -FOV)	Ancho 56 mm Alto 42 mm
Distancia de trabajo	$10.5 [\text{cm}]$
Medida del sensor	$6.6 \times 4.8 \text{ mm}$ ($1/2''$ <i>format</i>)

tabla 4.3: Requerimientos de dominio para la inspección de defectos en piles curtidas de bovino.

Nivel de detalle

Es necesario calcular cuál será la resolución (también denominada nivel de detalle) con la que el sistema será capaz de obtener imágenes. Así, se podrá determinar que límite se tiene para inspeccionar detalles pequeños como los defectos puntos de alfiler en la piel curtida.

Como se ha dicho anteriormente la porción de piel curtida es de 56×42 [mm]. Una vez que la imagen se ha encuadrado y se han eliminado los márgenes, se obtiene una imagen de 640×480 *pixels*.

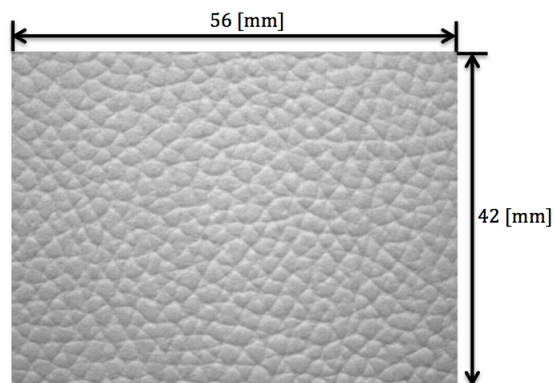


Figura 4.7: Porción de piel curtida de bovino a examinar. *Elaboración propia (Julio-2012)

Si dividimos $640/56 = 11.42$ *pixels*/mm y $480/42 = 11.42$ *pixels*/mm se puede decir que cada milímetro de la textura equivale aproximadamente a 11 *pixels* de la imagen. Con esta resolución se pueden inspeccionar la mayoría de los defectos en las pieles curtidas.

4.1.3.2. Hardware IMAQ de adquisición de imágenes

El último elemento que forma parte del sistema de adquisición es la placa de captura de imágenes junto con el software *driver* NI-IMAQ asociado. A grandes rasgos, la tarjeta se encarga de digitalizar la señal de video analógica procedente de la cámara, a continuación se aplica un preprocesado si es necesario, se envía al ordenador para que el software de programación analice los datos y decida. Todo este proceso entre el ordenador y el hardware lo controla el *driver* NI-IMAQ.

IMAQ PXI-1411

La tarjeta de adquisición IMAQ analógica de National Instruments[®] utilizada es la IMAQ PXI-1411, que es un módulo que acepta una entrada analógica de video desde cámaras estándares a color o monocromáticas, esta trabaja con 8 bits (256 tonos de gris) y permite una alta velocidad de conversión (hasta 25 imágenes por segundo)

La tarjeta trabaja con un *driver* de National Instruments[®], y dentro de las aplicaciones NI IMAQ (LabVIEW[®] y IMAQ *Vision builder*) se pueden acceder a la tarjeta de video sin necesidad de programar los registros internos de la misma. El software se encarga de realizar los accesos al hardware de manera amigable. Con esto, se facilita mucho el trabajo del programador y se optimiza el funcionamiento del hardware. El inconveniente más grande es que si se requiere utilizar un lenguaje de programación que no sea LabVIEW[®] (como *visual C* o otros), es muy probable que haya que programar un *driver* específico para controlar la placa de adquisición.

A continuación, se presenta las características básicas de la tarjeta de adquisición.

- Conversión interna HSL a color en tiempo real para igualar color rápidamente.
- Tarjeta de 1 canal para adquisición de imagen a color o monocromática.

- Filtrado interno; ganancia y desfases programables.
- Adquisición de imagen parcial con región interna programable.
- Entrada tipo NTSC, PAL, S-video, RS170 o CCIR

NI-IMAQ *driver software*

Esta herramienta proporciona la unión entre la aplicación software y hardware. Es la interfaz de software que controla los detalles de hardware, se comunica con diferentes tarjetas, y permite que la funcionalidad del hardware sea fácilmente accesible al programador sin necesidad de escribir a nivel de registro u otros programas complicados. Esta herramienta está incluida en el dispositivo IMAQ de National Instruments®. NI-IMAQ es una extensa librería de funciones que se llaman desde el entorno de programación LabVIEW®. IMAQ hardware y NI-IMAQ están diseñados para trabajar con *Vision Deployment* (software de procesamiento de imagen), ofreciendo capturas en tiempo real, así como la extracción de información crítica en procesos de tiempo real.

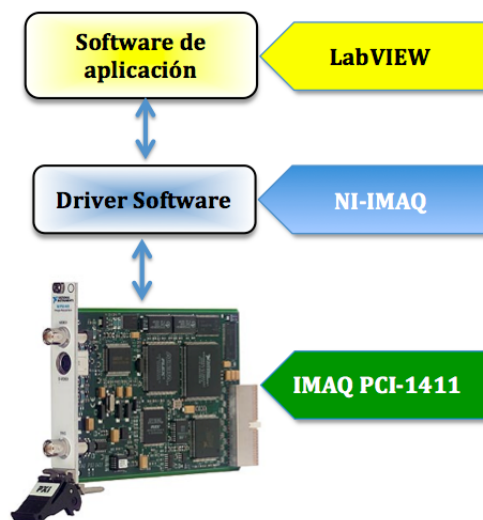


Figura 4.8: Unión entre hardware y software. *Elaboración propia (Julio-2012)

4.1.4. Iluminación y acondicionamiento de la imagen

El tener una buena iluminación es una función vital para realizar un apropiado acondicionamiento de la imagen. De esta manera se optimizará y mejorará el rendimiento en su procesamiento. Iluminar la escena de forma adecuada es un paso crítico muy importante para el éxito del desarrollo de una aplicación de inspección basada en computadora. Sin embargo, muy a menudo se ignora su relevancia. El objetivo de la iluminación es acondicionar el campo de imagen para separar el elemento que se requiere inspeccionar de su entorno o *background*.

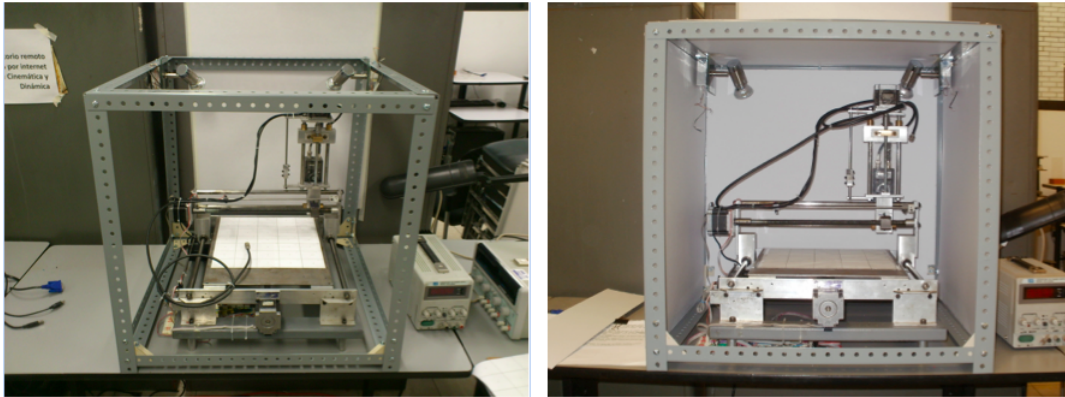
La elección de una buena técnica de iluminación debe permitir contrastar bien cada uno de los diferentes defectos en la piel curtida de bovino para poder realizar un apropiado procesamiento. Con una buena imagen, los posteriores pasos serán más fáciles de llevar a cabo. Por lo tanto, los aspectos básicos que se buscan con una buena iluminación en este proyecto son:

- Luz homogénea sobre el campo de visión.
- Máximo contraste para las características de interés
- Mínima sensibilidad a variaciones ambientales.

Además, no se debe de olvidar que para conseguir imágenes de calidad con el máximo contraste, no sólo influye la iluminación. Intervienen otros factores tales como características mecánicas, cámaras, óptica, así como la técnica empleada para procesar la imagen.

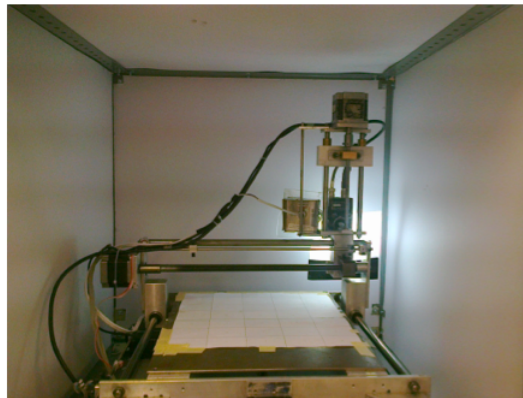
4.1.4.1. Sistema de iluminación implementado

En este proyecto se ha aislado al sistema de las condiciones ambientales construyendo una cámara opaca, la cual encierra al manipulador evitando así perturbaciones por la luz exterior, obteniendo una imagen homogénea en todo el campo de visión, la estructura está construida a partir de perfil de estantería y cubierto de papel cascarón en cada una de sus caras (Figura 4.9).



A)

B)



C)

Figura 4.9: A) Estructura de la cámara de aislamiento. B) Vista de la estructura. C) Manipulador completamente aislado de luz ambiental.

Las medidas de la cámara de aislamiento son:

- Altura: 62.8 [cm]
- Anchura: 62.8 [cm]
- Profundidad: 62.8 [cm]

Para la iluminación se ha utilizado un foco de luz de leds sostenidos en el eje z del manipulador, este fue envuelto en una pantalla para distribuir la iluminación en la escena. De este modo, se consiguieron niveles de luz contrastante que favorecen el proceso de inspección. En cuanto a las especificaciones propias de la iluminación se ha utilizado un lampara Led GU10 con 60 Leds de 4.5[W] con luz blanco cálido (Color de temperatura: 3000K – 3200K) que ofrece aproximadamente 300 – 350 lúmenes, esta fue seleccionada por medio de experimentación después de experimentar con diferentes tipos de lamparas. Con esa cantidad de luz y su colocación se ha disminuido los efectos de las sombras, de manera que se facilita enormemente la tarea de análisis y reconocimiento de defectos. Cabe destacar que la iluminación fue posicionada por ensayo y error hasta obtener una imagen adecuada para su procesamiento. En la siguiente figura, se muestra la iluminación implementada:

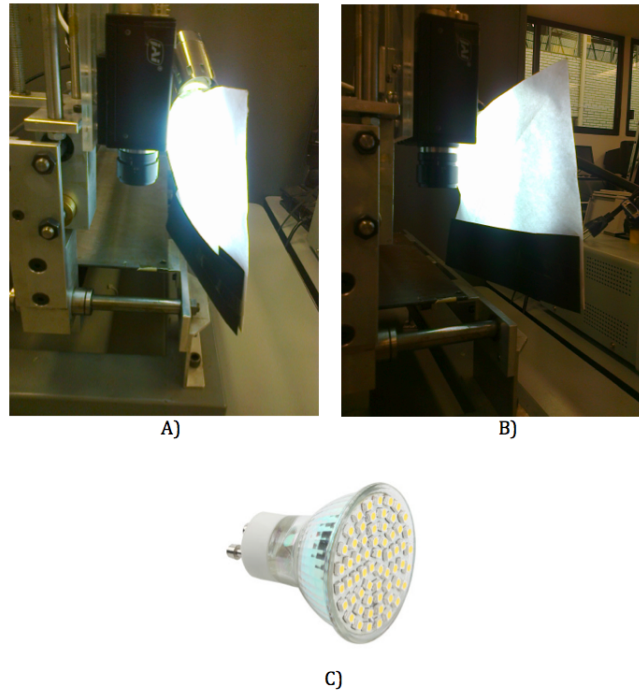


Figura 4.10: A) y B) iluminación implementada. C) Lámpara Led GU10 con 60 Leds

4.1.5. Manipulador cartesiano

Para llevar a acabo el recorrido o escaneado de la piel de forma automática se utiliza un manipulador cartesiano. La estructura mecánica de este manipulador esta formada por tres eslabones ortogonales. Cada uno constituido por un par helicoidal, que permite obtener un movimiento de translación linealmente relacionado con el movimiento rotatorio del motor, además de dos ejes cilíndricos de baja fricción que dan estabilidad a la estructura (Figura 4.11). La combinación de dos o tres grados lineales en dirección "y", "x" y/o "z" nos permite un movimiento libre en el plano vertical y horizontal o en el espacio. Como se mencionó anteriormente el grado de libertad en "z" no será utilizado durante el análisis de la piel, este se utilizará solo para posicionar a la cámara la distancia de trabajo especificada.

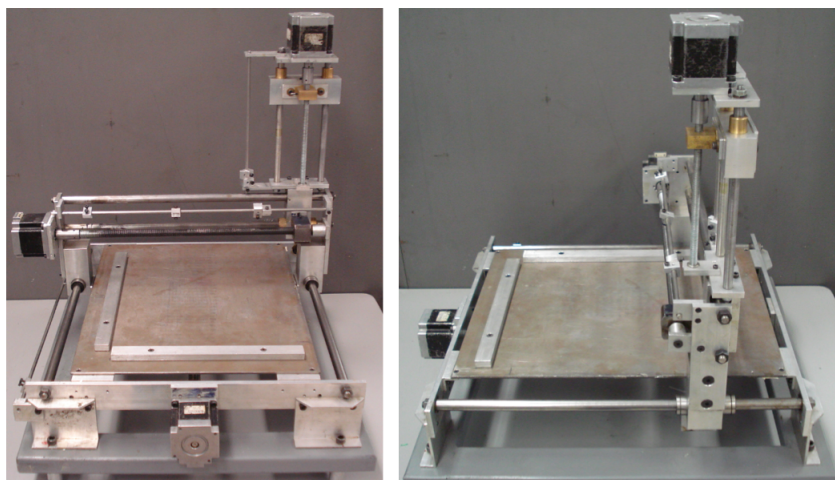


Figura 4.11: Manipulador cartesiano con 3 grados de libertad.

A continuación se enlistan las principales características de este manipulador:

- Estructura de un robot cartesiano. Se desplaza en las direcciones (x, y, z)

- 3 Motores Minebea 23KY-K unipolares
- 3 husillos de pares helicoidales.
- 3 Soportes para colocar sensores de herradura.

Los módulos lineales de husillo garantizan una gran fuerza de avance manteniendo una gran rigidez, pero en contraposición a una velocidad más baja, la precisión dependerá del paso del husillo.

El manipulador cuenta con 3 sensores para *home* que funcionan sin entrar en contacto. Los sensores se encuentran junto a los rieles de desplazamiento, exactamente al mismo nivel, estos son sensores de herradura. También se pueden encontrar dos sensores colocados en la zona de seguridad, si el manipulador pasa por este para directamente el programa cortando el movimiento de los motores.

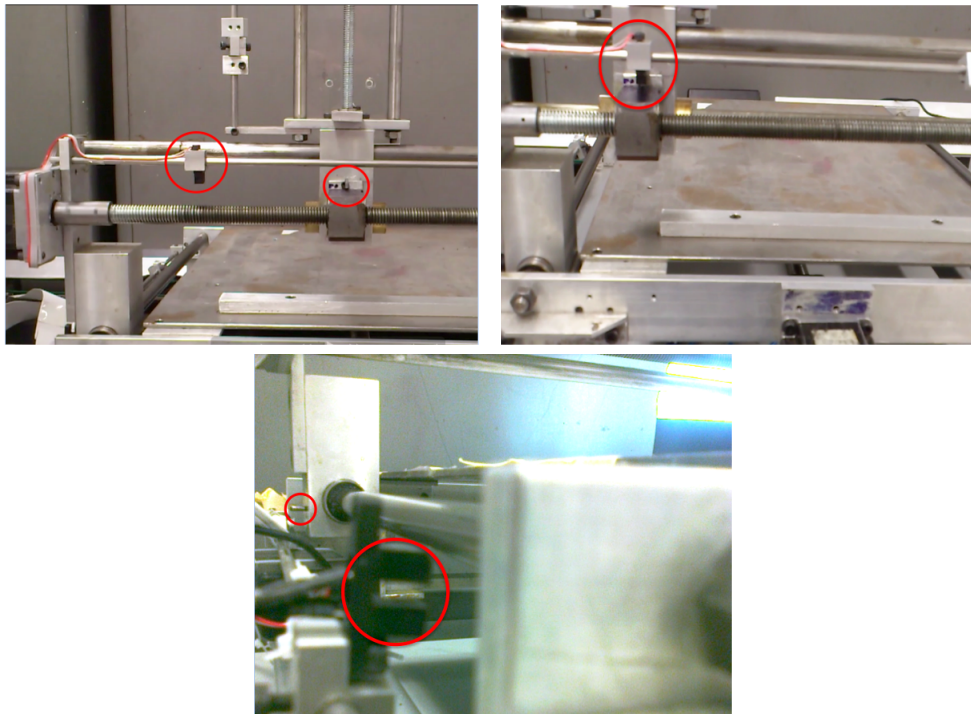


Figura 4.12: Sistema de sensado para establecer el *home* del sistema.

El sistema entrará en funcionamiento una vez que se le envié la señal de inicio mediante software, siempre y cuando este se encuentre en su posición de inicio. Este sistema interactuará con su entorno y se desplazara dentro del área de trabajo con respecto a una trayectoria establecida. Se podrá parar al sistema mediante un pulso de *stop*. Es de mucha importancia los elementos que permiten activar y desactivar el sistema, estos serán mediante software, así como elementos de señalización que indiquen los estados de operación y también el de procesamiento que es muy importante para poder saber si el sistema está operando o si se encuentra en una posición estática.

4.1.5.1. Motores

El manipulador cuenta con tres motores eléctricos paso a paso híbridos marca Minebea[®] 23KM-K unipolar que mueven los husillos de precisión para el desplazamiento del manipulador. Como se puede apreciar en la Figura (4.13), del motor paso a paso salen dos grupos de tres cables, uno de los cuales es común a dos bobinados. Las seis terminales que parten del motor, deben ser conectados al circuito de control, el cual, se comporta como cuatro conmutadores electrónicos que al ser activados o desactivados producen la alimentación de los cuatro grupos de bobinas.



Figura 4.13: Motor Minebea 23KM-K.

Se utilizó una secuencia de paso doble completo (normal) para la implementación de los motores, la secuencia se detalla a continuación.

Paso	Bobina A	Bobina B	Bobina C	Bobina D
1	+V	+V	-V	-V
2	-V	+V	+V	-V
3	-V	-V	+V	+V
4	+V	-V	-V	+V

tabla 4.4: Secuencia normal de pasos para motores unipolares.

Esta es la más utilizada y la que generalmente recomienda el fabricante. Con esta secuencia el motor avanza un paso por vez y debido a que siempre hay al menos dos bobinas activas, se obtiene un alto torque de paso y retención.

Características principales del motor (Véase Apéndice B).

- 1.8° por paso.
- Secuencia Uni-polar
- Corriente 3 [A]
- Frecuencia máxima: 10 MHz

A continuación se observa el *driver* recomendado por el fabricante, así como el diagrama de conexión en las terminales del motor paso a paso.

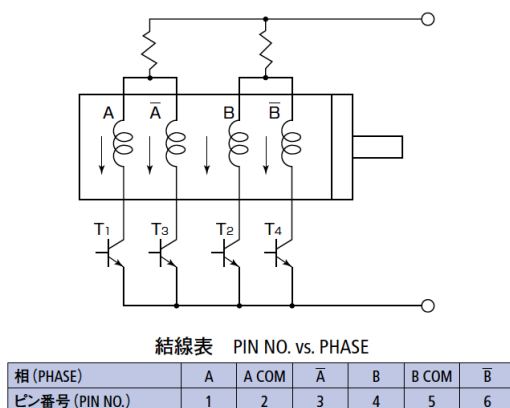


Figura 4.14: *Driver* para el motor y terminales de conexión. Véase Hoja de datos

4.2. Hardware de control

Para realizar el hardware de control se debe tener en cuenta el tipo de sistema de control a implementarse, es por esto que a continuación se describe los tipos de sistemas de control que existen y por consiguiente se podrá establecer el sistema que será útil para el proyecto

4.2.1. Tipos de sistema de control

Sistema de lazo abierto

Los sistemas de control en los que la salida no tiene efecto sobre la acción de control, se denominan sistemas de control de lazo abierto. En otras palabras, en un sistema de control de lazo abierto la salida ni se mide ni se retroalimenta para compararla con la entrada¹.

En cualquier sistema de control de lazo abierto, no se compara la salida con la entrada de referencia. Por lo tanto, para cada entrada de referencia corresponde una condición de operación fija, cabe señalar que la precisión del sistema depende de la calibración.

Sistema de lazo cerrado

En un sistema de lazo de control cerrado, la variable a ser controlada es continuamente medida y así comparada con un valor predeterminado. Si existe una diferencia entre estas dos variables (error o desviación del sistema), los ajustes son realizados hasta que la diferencia cuantificada es eliminada y la variable controlada iguala la variable de referencia. Con frecuencia se llama así a los sistemas de control retroalimentado. La señal de error actuante, que es la diferencia entre la señal de entrada y la de retroalimentación (que puede ser la señal de salida o una función de la señal de salida y sus derivadas), entra al controlador para reducir el error y llevar la salida del sistema a un valor deseado. El término lazo cerrado implica siempre el uso de la acción de control retroalimentado para reducir el error del sistema².

Al conocer las características de los tipos de sistema de control se elige el sistema de control de lazo abierto, puesto que el manipulador tiene motores a pasos y estos pueden ser controlados sin retroalimentación. En el manipulador, se envía la orden desde el PXI al controlador (entrada) y los motores del manipulador cartesiano la ejecutan (salida), entonces, al no comparar la entrada con la salida para efectuar la acción el sistema de control a utilizarse es el sistema de lazo abierto.

La visión general del control a realizar se presenta en la Figura (4.15).

4.2.1.1. Computadora

Se utiliza un Chasis NI PXI-1031 con el módulo PXI-8195 de National Instruments[®] que es un controlador embebido de alto rendimiento basado en Celeron M 370. Este es ideal para aplicaciones que requieren de análisis intenso o desarrollo de sistemas PXI. Un controlador embebido NI PXI-8195 en un chasis PXI ofrece una plataforma de PC compacta y de alto rendimiento para aplicaciones de instrumentación modular y adquisición de datos. El módulo se encarga de la comunicación entre el usuario y la máquina, así como del procesamiento de imágenes para la detección de defectos en pieles curtidas de bovino.

Dispositivo de adquisición y envío de señales

La tarjeta PXI-6123 está diseñada para aplicaciones de laboratorio y pruebas, sin embargo proporciona gran confiabilidad al alcanzar grandes tasas de velocidad de adquisición de datos. El módulo de adquisición de datos (DAQ) multifunción de National Instruments[®] tiene un convertidor analógico-digital dedicado (ADC) por canal para máximo rendimiento del dispositivo y más alta precisión de múltiples canales. Con velocidades de muestreo de 500 kS/s por canal, cuatro rangos de entrada desde ± 0.2 a $\pm 10V$, dos contadores/temporizadores de 24 bits y ocho líneas de E/S digital temporizadas por hardware. La plataforma PXI junto con la tarjeta NI 6123 nos permite

¹http://www.steamcontrol.com/index_docum_valvcontrol.html

²http://www.steamcontrol.com/index_docum_valvcontrol.html

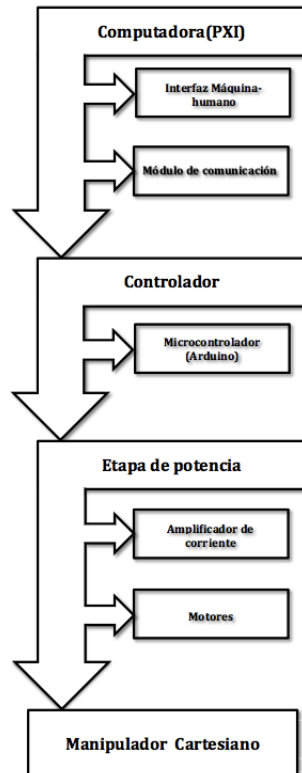


Figura 4.15: Diagrama de bloques del hardware de control. *Elaboración propia (Julio-2012)

hacer la comunicación entre el PXI y el movimiento del Manipulador. Se enlistan las principales características de este módulo.

Analógica

Características de entrada

- 8 entradas analógicas
- Resolución del ADC 16 bits
- Frecuencia de muestreo 500 kS/ por canal
- Rangos de señal de entrada $\pm 10, \pm 5, \pm 2.5, \pm 1.25$ [V]

Digital I/O

- 8 canales de entrada/salida
- Compatibilidad TTL/CMOS
- tasa de transferencia 10 *Mwords/s*
- 8 Entradas y salidas digitales

En la Figura (4.16) se observa la tarjeta de adquisición PXI-6123, cable R6868 y el bloque conector CB-68LP utilizados en este proyecto.

4.2.2. Microcontrolador

Para el control de los motores se utiliza el hardware Arduino. Este es un prototipo electrónico de plataforma abierta de fácil uso que consta básicamente de una placa de circuito impreso que contiene un microcontrolador de la marca "ATMEL" que cuenta con suficientes entradas y salidas

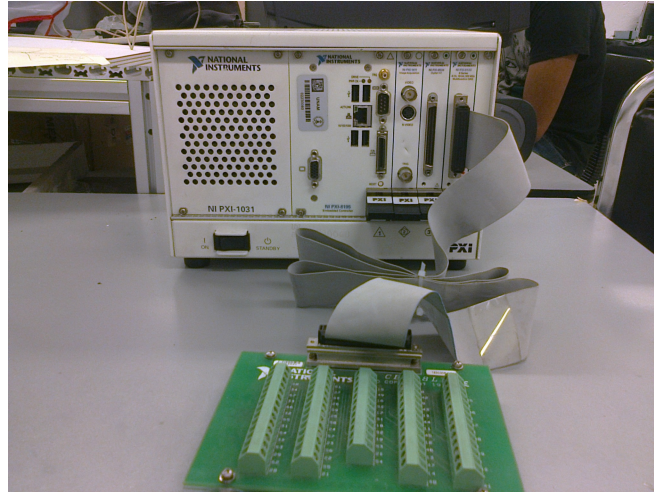


Figura 4.16: PXI y modulo de adquisición de datos con su correspondiente bloque conector.

digitales para cubrir con las necesidades del proyecto. Arduino posee una gran cantidad de ventajas a comparación de otros microcontroladores:

- Viene en una placa donde todo está listo para empezar a ser utilizado.
- Se programa con facilidad.
- El costo del microcontrolador no es tan elevado a comparación de otros microcontroladores.

4.2.2.1. Características básicas

En este proyecto se utiliza la versión de Arduino: *Mega* (Figura 4.17).

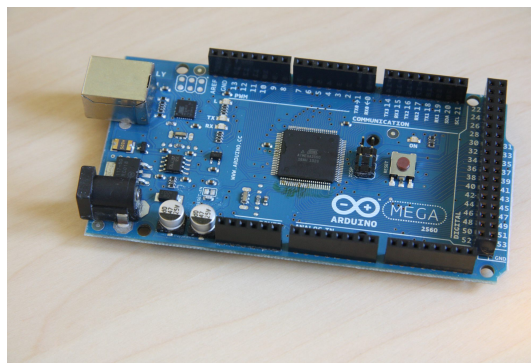


Figura 4.17: Microcontrolador utilizado.

El arduino Mega tiene 54 entradas/salidas digitales (de las cuales 14 proporcionan salida PWM), 16 entradas analógicas, 4 UARTS (puertos serie por hardware), un cristal oscilador de $16MHz$, conexión USB, entrada de corriente, conector ICSP y botón de *reset*. El arduino Mega puede ser alimentado vía conexión USB o con una fuente de alimentación externa. El origen de la alimentación se selecciona automáticamente. En nuestro caso el arduino será alimentado vía USB.

En la Tabla(4.5) se en listan las Características básicas del microcontrolador.

Microcontrolador	ATmega 1280
Voltaje de funcionamiento	5 [V]
Voltaje de entrada (recomendado)	7-12 [V]
Voltaje de entrada (límite)	6-20 [V]
Pines E/S digitales	54 (14 proporcionan salida de PWM)
Pines de entrada analógica	16
Intensidad por pin	40 [mA]
Intensidad en pin 3.3 [V]	50 [mA]
Memoria Flash	128 KB de las cuales 4kb las usa el gestor de arranque (<i>bootloader</i>)
SRAM	8KB
EEPROM	4KB
Clock Speed	16 MHz

tabla 4.5: Especificaciones básicas del Arduino mega.

Es necesario recalcar que se utiliza solo un microcontrolador para controlar tres motores a pasos (Figura 4.18).

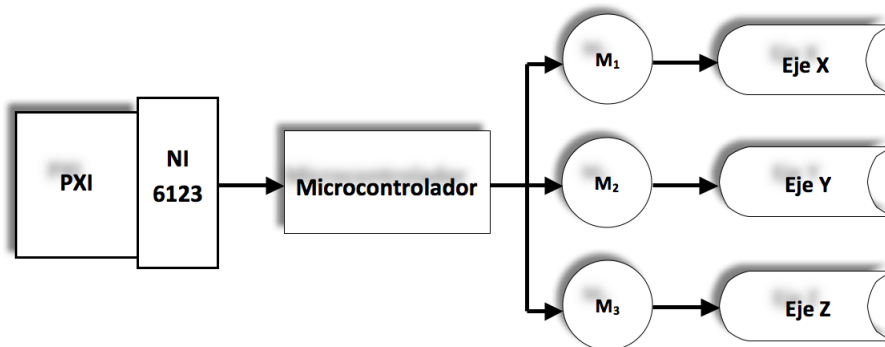


Figura 4.18: Diagrama de flujo del controlador. *Elaboración propia (Julio-2012)

4.2.3. Etapa de potencia

Como se describió anteriormente, los motores implementados requieren una secuencia de pulsos en sus bobinas para generar un movimiento. En este proyecto, el controlador capaz de enviar esta secuencia, queda a cargo de un microcontrolador Arduino mega. Para el movimiento del manipulador se utilizan tres motores a pasos, por lo cual se debería de realizar un controlador que tenga la capacidad de manejar tres secuencias de manera independiente, pero cabe mencionar que las tres secuencias no son manejadas simultáneamente, ya que para la trayectoria que se definió en el escaneado de la piel curtida no es necesario mover los motores al mismo tiempo, es decir, se moverá un motor a la vez, simplificando el manejo y estructura de la programación.

Del microcontrolador Arduino se obtendrán las señales necesarias para genera los pulsos para polarizar a las bobinas y así mover los motores. El alcance de un paso es variable, depende de la construcción física del motor, puede ser sólo de 1.8° ó hasta de 90° . En nuestro caso el paso es de 1.8° por lo que se requiere de 200 pasos. Esta precisión en el movimiento por paso hace a los motores PAP ideales para sistemas de lazo abierto.

En el motor paso a paso unipolar, el común en las bobinas es conectado a Vcc. El otro extremo es llevado a tierra a través de un transistor, como se muestra en la Figura(4.21), donde el microcontrolador controla el disparo de los transistores y con ello energiza a alguna de las bobinas, imponiendo un polo magnético norte.

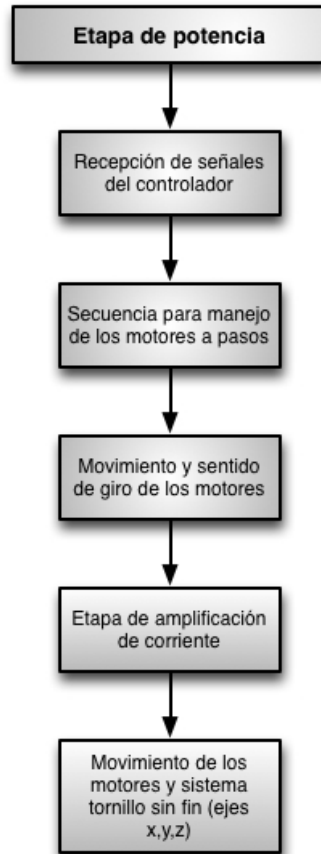


Figura 4.19: Diagrama de flujo de la etapa de potencia. *Elaboración propia (Julio-2012)

Para la elaboración de la etapa de potencia se implementa un *driver* utilizando transistores y diodos para una mayor seguridad en el manejo de corriente. Con base a las especificaciones de los motores se utilizaron transistores tip41 y diodos 1N4002 que tiene las siguientes características y que soportan los voltajes y corrientes máximas de los motores:

- Tip41
 - Voltaje colector-emisor $60 [V_{dc}]$
 - Voltaje colector-base $60 [V_{dc}]$
 - Voltaje emisor-base $5 [V_{dc}]$
 - Corriente Colector (picos continuos) $6 - 10 [A_{dc}]$

- Diodos
 - $100 [V]$ de bloqueo máximo de Voltaje DC
 - Corriente directa en valor medio $1 [A]$
 - Valor de pico no repetitivo $30 [A]$

A las bobinas se les colocó diodos inversamente polarizado en paralelo para proteger al elemento de conmutación de la sobretensión que causa la bobina cuando es conmutada. Se utilizaron 3 diodos 1N4002 en paralelo en cada bobina para evitar daños en el microcontrolador arduino. En la Figura (4.21) podemos observar la etapa de potencia implementada y el diagrama a bloques de la etapa de potencia en la Figura(4.22).

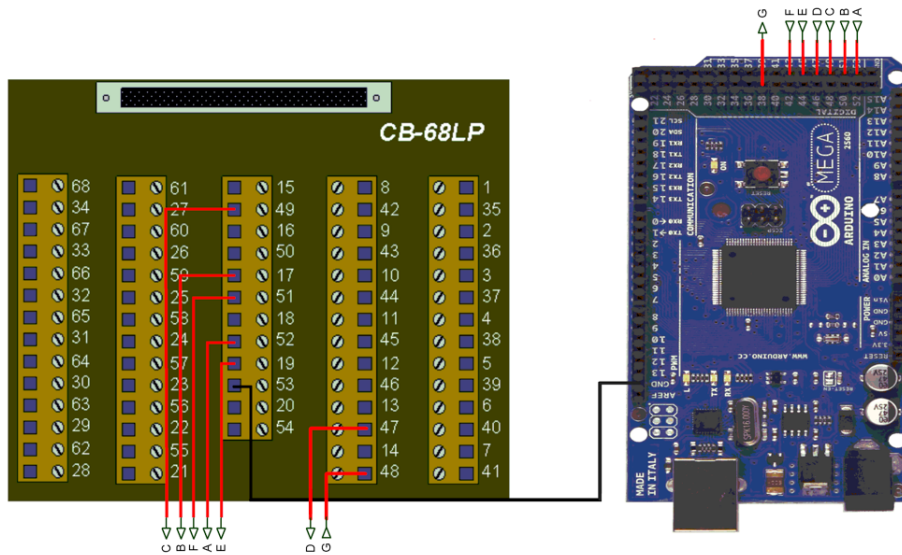


Figura 4.20: Conexión entre la tarjeta de adquisición y el microcontrolador arduino.

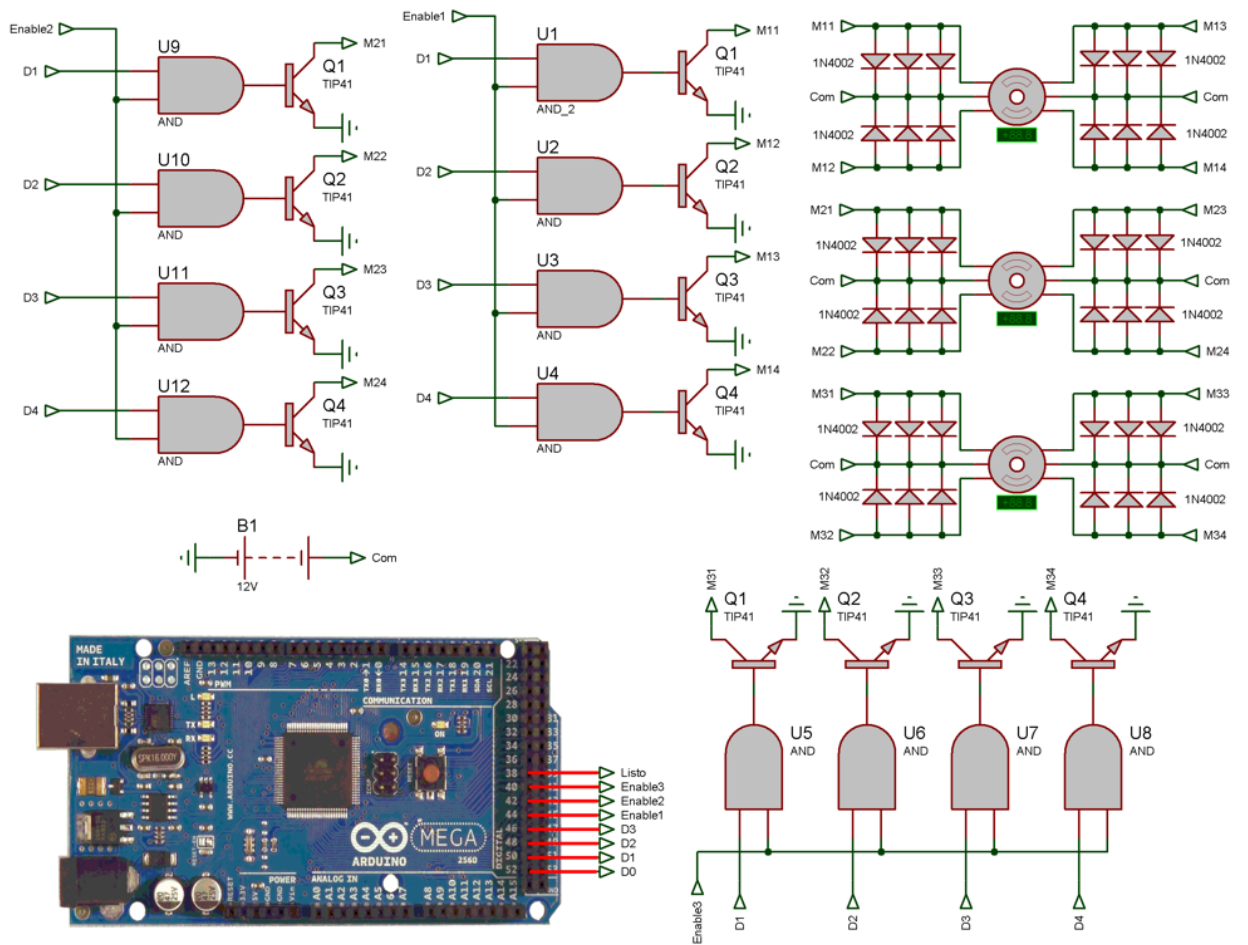


Figura 4.21: Esquemático del hardware para el controlador de motor a pasos.

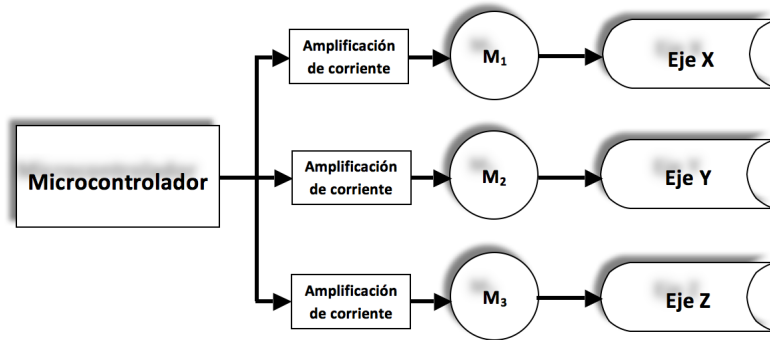


Figura 4.22: Diagrama de la etapa de potencia. *Elaboración propia (Julio-2012)

El diagrama a bloques en la figura(4.23) muestra el hardware final del controlador para los motores a paso.

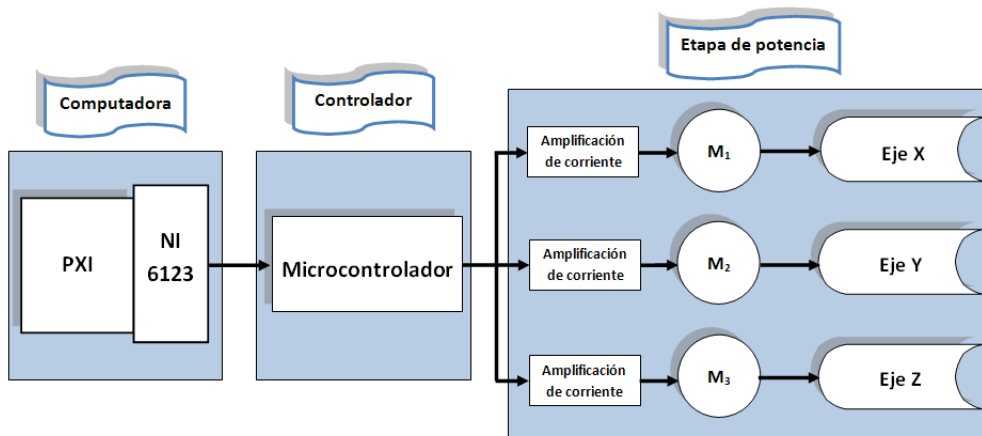


Figura 4.23: Diagrama del Hardware total. *Elaboración propia (Julio-2012)

4.2.4. Sensores

Se ubicaron sensores finales de carrera para evitar que las piezas del manipulador sufran algún tipo de choque entre ellas, además de que se incluyen también tres sensores para el punto cero de inicio, estos son del tipo *switch* óptico H21A1 (tipo herradura) como se pueden ver en la Figura (4.24). Estos sensores tienen como emisor un diodo infrarrojos y como receptor un fototransistor. En este caso el emisor y el receptor están separados a una distancia de 3 mm y entre ellos existe un espacio para que un objeto pueda introducirse y romper la barrera infrarroja, obteniendo un pulso en el momento que se bloquea el paso de la luz en medio de la herradura.

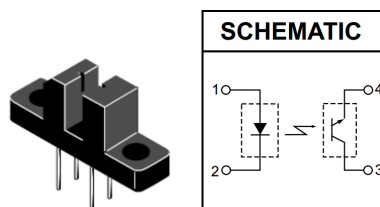


Figura 4.24: Esquemático del sensor de herradura. *Tomado de Apéndice C .

A continuación se muestra el circuito implementado para los sensores de herradura.

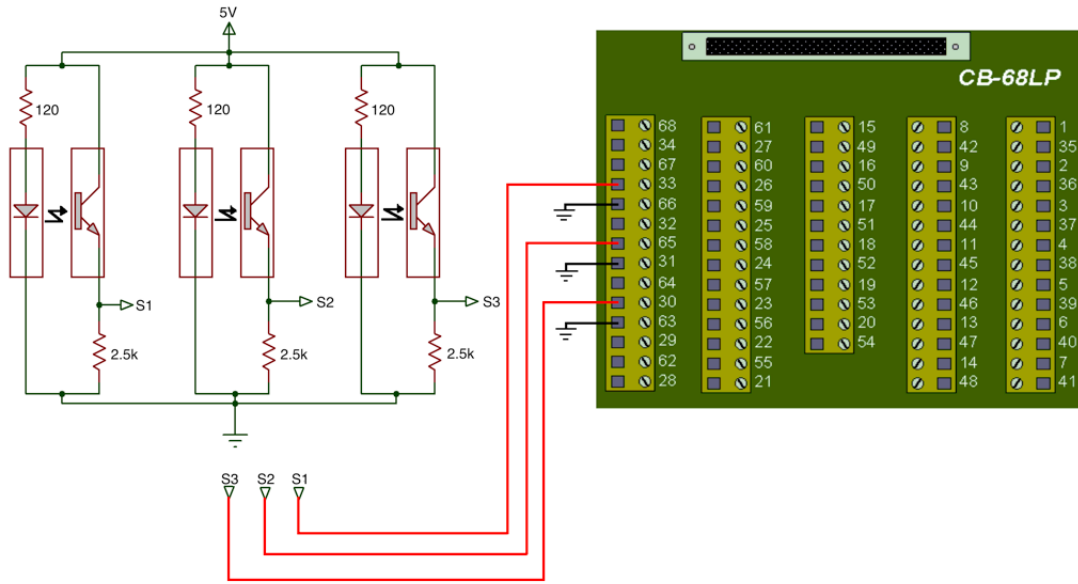


Figura 4.25: Diagrama de conexión del sistema de sensado para el *home*.

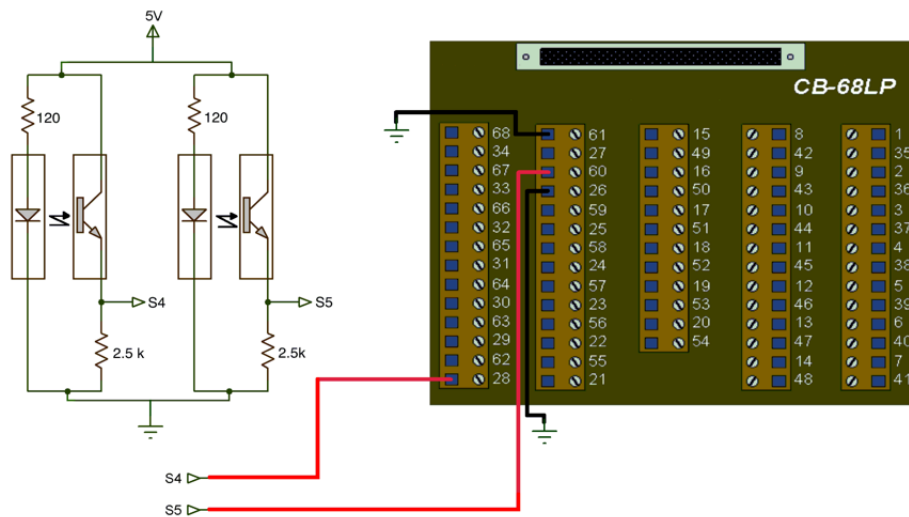


Figura 4.26: Diagrama de conexión del sistema de sensado de emergencia.

4.3. Software incorporado

El paquete de programación LabVIEW[®] permitirá crear una interfaz gráfica amigable entre la computadora y el usuario. Con este programa se estableció la comunicación entre el PXI y el microcontrolador, así como el envío y recepción de la información de control.

Para la programación del proyecto se utiliza una arquitectura maestro-esclavo donde se divide la programación en dos partes:

- Control del manipulador y procesamiento de imágenes.
- Control de motores.

El control del manipulador así como el procesamiento se realizan en el PXI y el microcontrolador arduino se encargará de mandar los pulso a la etapa de potencia para controlar los motores. Los

ejes del manipulador y los motores se establecieron de la siguiente forma: M1(Izquierda-Derecha), M2(Adelante-Atrás), M3(Arriba-Abajo). véase Figura(4.27)

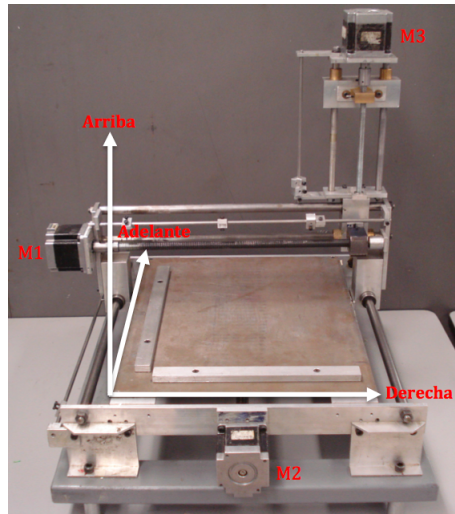


Figura 4.27: Ejes y motores del manipulador.

El software diseñado tiene dos modos de análisis (manual y automático), en el modo manual el usuario puede mover el manipulador mediante los botones dedicados para ello sobre toda la área de trabajo y tomar una porción de ella para analizarla, y en el modo automático el programa realizará todo el recorrido, escaneando y analizando la piel curtida sin la intervención del operador y obtener al finalizar una imagen general de la piel en el espacio de trabajo del manipulador, así como indicadores en pantalla de las porciones donde se encontraron los defectos.

El sistema entrará en funcionamiento una vez que se le envié la señal de inicio mediante software, siempre y cuando este se encuentre en su posición de *home*, si no es así y este se encuentra en el modo automático, el programa mandará al manipulador a la posición de inicio. Este sistema interactuará con su entorno y se desplazará dentro del área de trabajo con respecto a una trayectoria establecida. Se podrá parar el sistema mediante un pulso de *stop*. Es de mucha importancia los elementos que permiten activar y desactivar el sistema, estos serán mediante software, así como elementos de señalización que indiquen los estados de operación y de procesamiento que son muy importante para poder saber si el sistema está operando o si se encuentra en una posición estática.

Como se mencionó anteriormente, LabVIEW[®] es un ambiente completamente gráfico y permite realizar toda la programación mediante bloques, es decir de forma gráfica. No es necesario implementar cientos de líneas de código como en lenguajes de programación tradicionales.

Existen dos pantallas principales en LabVIEW[®], una que es directamente la interface con la cual el usuario interactúa llamada "Panel frontal". En esta se muestran los resultados obtenidos de los experimentos, mediante *displays* e indicadores. La segunda es conocida como "Diagrama de bloques", aquí es donde se realiza toda la programación, es decir lectura y envío de señales, procesamiento y manipulación de señales para obtener los resultados necesarios, además de configurar y seleccionar los parámetros adecuados para la comunicación con la tarjeta de adquisición de datos e imágenes.

Cada uno de los pasos seguidos se explicará con más detalle en las siguientes secciones, aunque ambos se realizaron en paralelo, se decidió mostrarlo por separado para fines de comprensión.

4.3.1. Diagrama de bloques

En esta parte es donde se encuentran todas las conexiones realizadas entre los diferentes bloques con los que cuenta LabVIEW[®]. Cabe mencionar que la programación del proyecto se realizó por

medio de máquinas de estado.

4.3.1.1. Lógica de programación

La lógica de programación se presenta en el diagrama de estado de la Figura (4.28), pero a continuación se explicará de forma detallada su contenido:

- Estado 0: el programa esperará la señal del usuario para comenzar el programa.
- Estado 1: Se leerá si el programa a ejecutar debe ser en modo manual o automático.
- Modo manual
 - Estado 2: Los valores de los botones en el Panel frontal (Izquierda, Derecha, Adelante, Atrás, Arriba, Abajo y Cámara) serán leídos, dependiendo de su combinación serán enviadas 6 señales de control al microcontrolador. A continuación se muestra las posibles combinaciones y los estados a los cuales conmutan:

<i>Entradas</i>							
<i>Cam</i>	<i>Abaj</i>	<i>Arrib</i>	<i>Atrs</i>	<i>Adel</i>	<i>Der</i>	<i>Iz</i>	<i>Estado</i>
0	0	0	0	0	0	0	Estado3 (No hacer nada)
0	0	0	0	0	0	1	Estado4(Mover M1-)
0	0	0	0	0	1	0	Estado5(Mover M1+)
0	0	0	0	0	1	1	Estado3(No hacer nada)
0	0	0	0	1	0	0	Estado6(Mover M2+)
⋮	⋮	⋮	⋮	⋮	⋮	⋮	Estado3(No hacer nada)
0	0	0	1	0	0	0	Estado7(Mover M2-)
⋮	⋮	⋮	⋮	⋮	⋮	⋮	Estado3(No hacer nada)
0	0	1	0	0	0	0	Estado9(Mover M3+)
⋮	⋮	⋮	⋮	⋮	⋮	⋮	Estado3(No hacer nada)
0	1	0	0	0	0	0	Estado8(Mover M3-)
⋮	⋮	⋮	⋮	⋮	⋮	⋮	Estado3(No hacer nada)
1	0	0	0	0	0	0	Estado10(Procesar Imagen)

Las salidas enviadas al microcontrolador para cada estado son las siguientes:

<i>Salidas</i>						
<i>Estado</i>	M_1	M_2	M_3	<i>Sentido</i>	<i>Modo</i>	<i>Maestro</i>
Estado3	0	0	0	0	0	0
Estado4	1	0	0	0	0	1
Estado5	1	0	0	1	0	1
Estado6	0	1	0	1	0	1
Estado7	0	1	0	0	0	1
Estado8	0	0	1	0	0	1
Estado9	0	0	1	1	0	1
Estado10	0	0	0	0	0	0

- Modo automático
 - Estado10: Antes de iniciar el análisis, el programa comprobará si el manipulador se encuentra en *home*, para esto se leerán 3 sensores posicionados en los 3 planos: arriba-abajo (S_3), atrás-adelante (S_2) e izquierda-derecha (S_1). Las siguientes tablas muestran los casos en los cuales se pueden encontrar las señales de los sensores y a que estado cambian, así como las correspondientes salidas para que el manipulador pueda llegar a su estado de *home*.

<i>Sensores de home</i>			
<i>Sensor1</i>	<i>Sensor2</i>	<i>Sensor3</i>	<i>Estado</i>
0	0	0	Estado11 (L1)
0	0	1	Estado12 (L5)
0	1	0	Estado11 (L2)
0	1	1	Estado13 (L7)
1	0	0	Estado11 (L3)
1	0	1	Estado12(L6)
1	1	0	Estado11(L4)
1	1	1	Estado 14(el manipulador esta en Home)

<i>Salidas</i>						
<i>Estado</i>	M_1	M_2	M_3	<i>Sentido</i>	<i>Modo</i>	<i>Maestro</i>
Estado11(Arriba)	0	0	1	1	0	1
Estado12(Atrás)	0	1	0	0	0	1
Estado13(Izquierda)	1	0	0	0	0	1
Estado14	0	0	0	0	0	1

- Estado19: En este estado se crearán 2 contadores para formar una matriz de 7×5 que son los cuadros en los cuales se dividió el espacio de trabajo y con los cuales el programa sabrá en que posición se encuentra (Matriz 4.3). Se partirá con ($i = 0$ y $j = 0$), se analizará la imagen tomada y se señalará el defecto (si esta porción la tiene), se mandará un señal al arduino para que inicie con el movimiento de los motores dando los pasos establecidos, cuando el arduino termine de dar los pasos establecidos para llegar a la siguiente posición se enviará una señal al PXI para que analice la siguiente porción, y así sucesivamente. Cuando el contador (j) sea "par" el Motor 2 se moverá positivamente y cuando el contador (j) sea "impar" el Motor 2 se moverá en sentido negativo (Matriz 4.4). Si se cumple la condición ($i = 0$ ó $i = 6$) se mandará una señal para mover el motor 1 negativamente. Con la condición ($i = 6$ y $j = 4$) el programa habrá terminado el recorrido en el espacio de trabajo (Matriz 4.4).

$$\begin{bmatrix}
 (6,0) & (6,1) & (6,2) & (6,3) & (6,4) \\
 (5,0) & (5,1) & (5,2) & (5,3) & (5,4) \\
 (4,0) & (4,1) & (4,2) & (4,3) & (4,4) \\
 (3,0) & (3,1) & (3,2) & (3,3) & (3,4) \\
 (2,0) & (2,1) & (2,2) & (2,3) & (2,4) \\
 (1,0) & (1,1) & (1,2) & (1,3) & (1,4) \\
 (0,0) & (0,1) & (0,2) & (0,3) & (0,4)
 \end{bmatrix} \quad (4.3)$$

Par Impar Par Impar Par

$$\begin{bmatrix}
 \Rightarrow & \Downarrow & \Rightarrow & \Downarrow & Fin \\
 \Uparrow & \Downarrow & \Uparrow & \Downarrow & \Uparrow \\
 \Uparrow & \Downarrow & \Uparrow & \Downarrow & \Uparrow \\
 \Uparrow & \Downarrow & \Uparrow & \Downarrow & \Uparrow \\
 \Uparrow & \Downarrow & \Uparrow & \Downarrow & \Uparrow \\
 Ini & \Rightarrow & \Uparrow & \Rightarrow & \Uparrow
 \end{bmatrix} \quad (4.4)$$

[+ - + - +]

- Estado24: Si el contador j es par se toma una imagen de la porción y se procesa para la identificación de defectos, el resultado obtenido se mostrará en pantalla señalando si se encontraron defectos o no, una imagen de la correspondiente identificación del defecto es guardada. Al terminar se mandará una señal al arduino para que mueva el motor 1 positivamente y así alcanzar la siguiente posición, al llegar a esta posición el arduino regresará una señal para indicar que termino de dar los pasos y esta en una nueva posición, el PXI aumentará el contador una unidad en i y se regresa al estado 19.
- Estado23: Si el contador j es impar a diferencia del estado 24 se moverá el motor 1 negativamente para alcanzar su posición y se decrementará el contador i .

Envío de señales

En el caso de este proyecto se generan seis señales de control para el arduino, estas son del tipo digital y serán enviadas a través de la tarjeta de adquisición. En la Figura (4.30) señalado en rojo se puede observa la parte en el programa donde se hace el envío de las señales digitales, cabe mencionar que las señales que se recalcan en amarillo cambiarán dependiendo del estado.

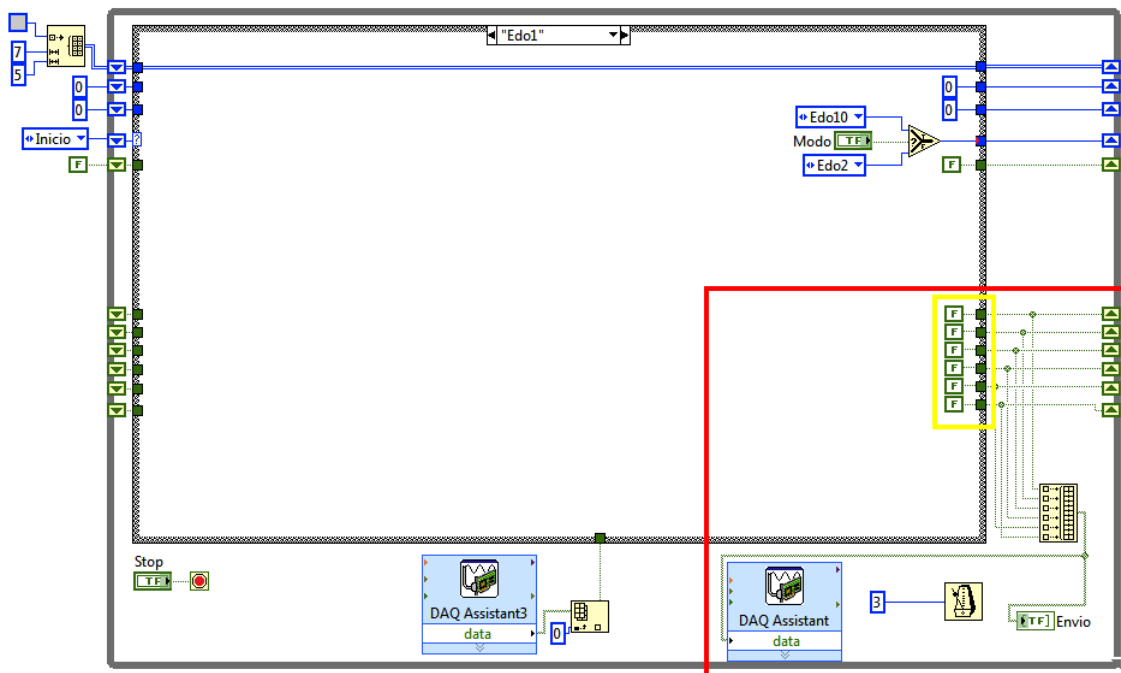


Figura 4.30: Envío de señales digitales al microcontrolador.

Las señales se especifican a continuación:

- Motor izquierda-derecha: Señal de control para el motor número 1 del manipulador.
- Motor adelante-atrás: Señal de control para el motor número 2 del manipulador.
- Motor arriba-abajo: Señal de control para el motor número 3 del manipulador.
- Sentido: Indica que dirección debe de girar el motor (positivo o negativo).
- Modo: Forma de ejecución manual o automático.
- Maestro: Señal para llamar al arduino para que ejecute una operación.

<i>Señal enviada</i>	<i>Pin en el módulo PXI-6123</i>	<i>Canal físico de la tarjeta CB-68LP</i>
Motor Izq-Derecha	Dev2/port0/line0	52
Motor Adel-Atrás	Dev2/port0/line1	17
Motor Arrib-Abajo	Dev2/port0/line2	49
Sentido	Dev2/port0/line3	47
Modo	Dev2/port0/line4	19
Maestro	Dev2/port0/line5	51

tabla 4.6: Señales digitales enviadas al arduino.

Adquisición de señales

Con respecto a la adquisición se necesitan leer 6 señales digitales, pero conociendo que ya se han ocupado 6 de las 8 salidas/entradas digitales de la tarjeta de adquisición, se utilizará una de estas señales digitales para leer la señal que envía el arduino indicando que termino de dar los pasos que se le ordenaron y 5 señales analógicas de voltaje para los sensores de herradura.

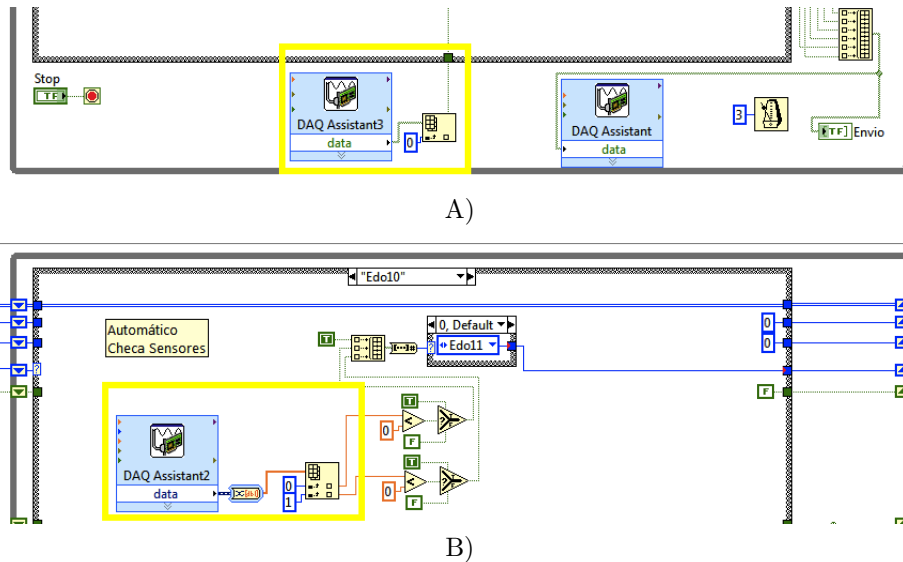


Figura 4.31: A) Lectura proveniente del Arduino B) Lectura de sensores de home.

Las señales adquiridas se especifican a continuación:

- Listo: Esta señal de tipo digital proviene del arduino e indica que el motor ha terminado de mover los pasos establecidos.
- Sensor1: Señal analógica del sensor posicionado en el eje izquierda-derecha del manipulador.
- Sensor2: Señal analógica del sensor posicionado en el eje atrás-adelante del manipulador.
- Sensor3: Señal analógica del sensor posicionado en el eje arriba-abajo del manipulador.
- Sensor4: Señal analógica del sensor de seguridad en el eje izquierda-derecha del manipulador
- Sensor5: Señal analógica del sensor de seguridad en el eje atrás-adelante del manipulador

<i>Señal recibida</i>	<i>Pin en el módulo PXI-6123</i>	<i>Canal físico de la tarjeta CB-68LP</i>
Listo	Dev2/port0/line7	48
Sensor1	Dev2/ai1	33
Sensor2	Dev2/ai2	65
Sensor3	Dev2/ai3	30
Sensor4	Dev2/ai4	28
Sensor5	Dev2/ai5	60

tabla 4.7: Señales digitales recibidas.

4.3.1.3. Configuración de adquisición de imagen

Para la captura de las imágenes se ejecutan VIs de inicialización y cierre de la cámara, se requiere además realizar una asignación de memoria para almacenar imágenes así como para descargarla. A continuación podemos observar los VIs para la adquisición de una imagen y la tarea que ejecutan.



IMAQ *Create*.vi crea un espacio en la memoria temporal, para que se almacene una imagen mientras se realizan cálculos sobre ella.



IMAQ *Dispose*.vi libera espacio de la memoria temporal (RAM). Este VI puede destruir y liberar su espacio de memoria o el de todas las imágenes creadas.



IMAQ *Snap* adquiere una única imagen cada vez que se ejecuta.



IMAQ *init* inicializa la tarjeta de adquisición de video.



IMAQ *Close* cierra todos los recursos utilizados a nivel de hardware (tarjeta de video)

El programa realiza los siguientes pasos para la adquisición de una imagen digital :

1. Crea una ubicación temporal de memoria para la imagen
2. Inicializa la tarjeta de adquisición
3. Adquiere una imagen de la cámara
4. Se realiza los demás pasos para la detección de defectos
5. Libera los recursos asociados con la adquisición.

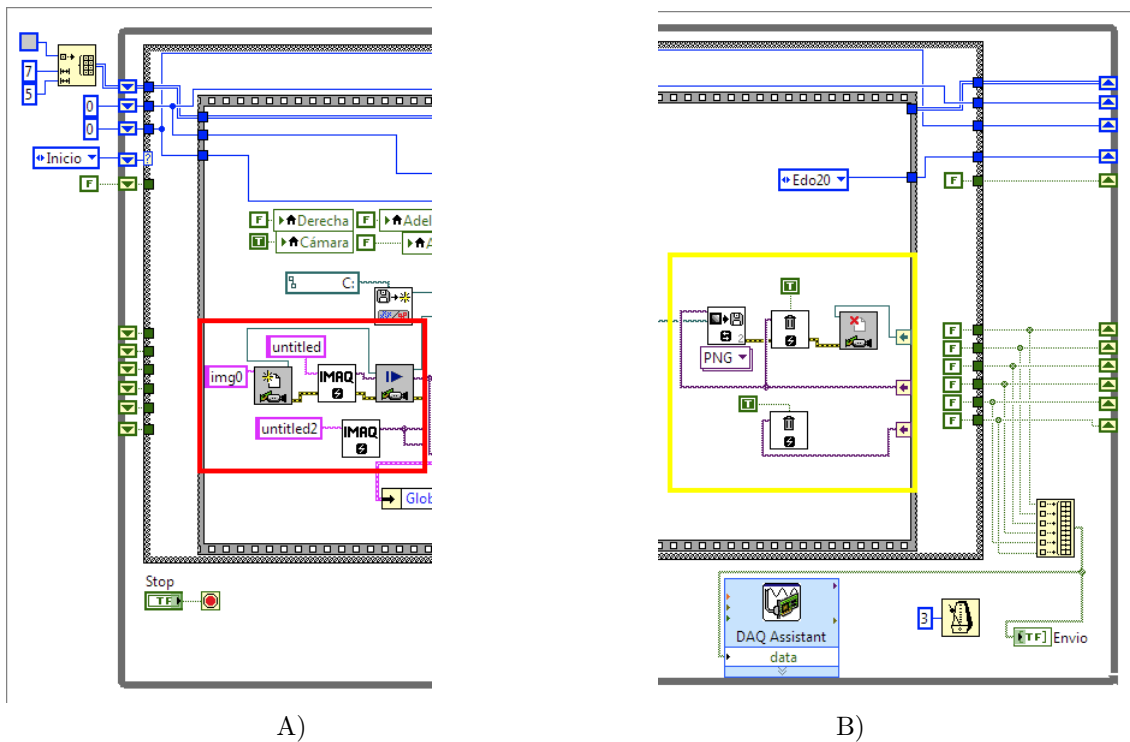


Figura 4.32: A) En el cuadro rojo se puede observar la parte donde se lleva a cabo la inicialización, asignación de memoria y adquisición de la imagen. B) En el cuadro amarillo se encuentra la parte del programa donde se cierra los recursos utilizados de hardware y libera espacio en memoria.

4.3.1.4. Configuración de la etapa de procesamiento

Después de haber adquirido la imagen, se utilizara el módulo "NI Vision Development" que nos facilita el procesamiento para la detección de defectos en texturas. La librería de visión contiene herramientas como detectores de borde y análisis de texturas.

Para realizar nuestra aplicación se utilizan los siguientes Vi's que nos ayudan a detectar y mostrar los defectos en la piel curtida.



IMAQ *Read Classifier* lee el clasificador que fue creado utilizando el *NI texture training interface*.



IMAQ *detect texture defect* detecta defectos en imágenes de texturas, teniendo como entradas el clasificador que fue creado utilizando "Texture Training Interface" y la imagen adquirida previamente. Como salida obtendremos una imagen binaria con todos los pixeles defectuosos.



IMAQ *MasktoROI* transforma una mascara de imagen en una región de interés (ROI).



IMAQ *Overlay ROI* superpone una región de interés (ROI) en una imagen



IMAQ *Merge Overlay* este proceso crea una superposición destructiva, eliminando la superposición no destructiva de una imagen.



IMAQ *Particle Analysis Report* devuelve el número de partículas detectadas en una imagen binaria y una matriz de informes que contienen las mediciones de partículas comúnmente utilizados.



IMAQ *Overlay Rectangle* superpone un rectángulo en una imagen.

El programa para la detección de defectos y señalamiento de defectos realiza los siguiente pasos:

1. Lee el clasificador antes ya creado y entrenado en *NI texture training interface*
2. El IMAQ *detect texture defect* analiza la imagen tomada y devuelve una imagen binaria.
3. Esta imagen es convertida a una región de interés.
4. De la región de interés se toma el parámetro *global rectangle* creando un cuadro que cubre el área donde se detectaron los defectos.
5. Se crea una superposición destructiva del rectángulo para esta se quede en la imagen.
6. La imagen es guarda con un nombre con respecto a los contadores (i,j).
7. Se libera espacio de la memoria.

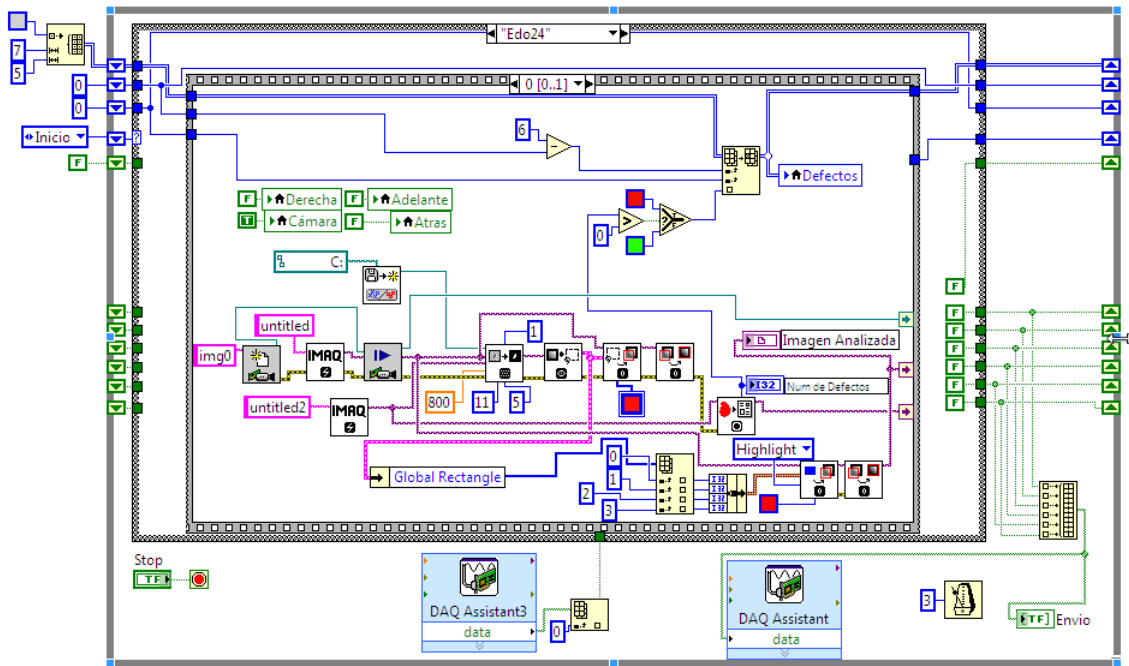


Figura 4.33: Etapa de procesamiento y detección de defectos en pieles curtidas.

4.3.2. Panel frontal

La pantalla accesible al usuario es conocida como panel frontal del instrumento virtual, es aquí donde se localizan todos los indicadores, botones y display que muestran los resultados de cada experimento.

Dependiendo de el modo (automático o manual) algunos elementos pueden comportarse como controles o indicadores. El panel frontal esta constituido de los siguientes elementos:

- Manual
 - Display de imagen analizada.
 - Indicadores de envío de señal al arduino.
 - Botones de control para el manipulador (Arriba-Abajo, Izquierda-Derecha y Arriba-Abajo).
 - Botón para seleccionar el modo a analizar la piel (Manual/Automático).
 - Botón de inicio de operación.
 - Botón de paro de análisis.
 - Botón para toma de imagen y procesamiento de la porción deseada.
- Automático
 - Display de imagen analizada.
 - Indicadores de envío de señal al microcontrolador.
 - Indicadores de movimiento del manipulador (Arriba-Abajo, Izquierda-Derecha y Arriba-Abajo).
 - Botón para seleccionar el modo a analizar la piel (Manual/Automático).
 - Botón de inicio de operación.
 - Botón de paro de análisis.
 - Indicador de toma de imagen y procesamiento de imagen.
 - Indicadores de posición de la cámara en el espacio de trabajo.

- Matriz de led's que indica los defectos encontrados en la piel curtida analizada.

En la Imagen (4.34) se pueden observar el panel frontal implementado.

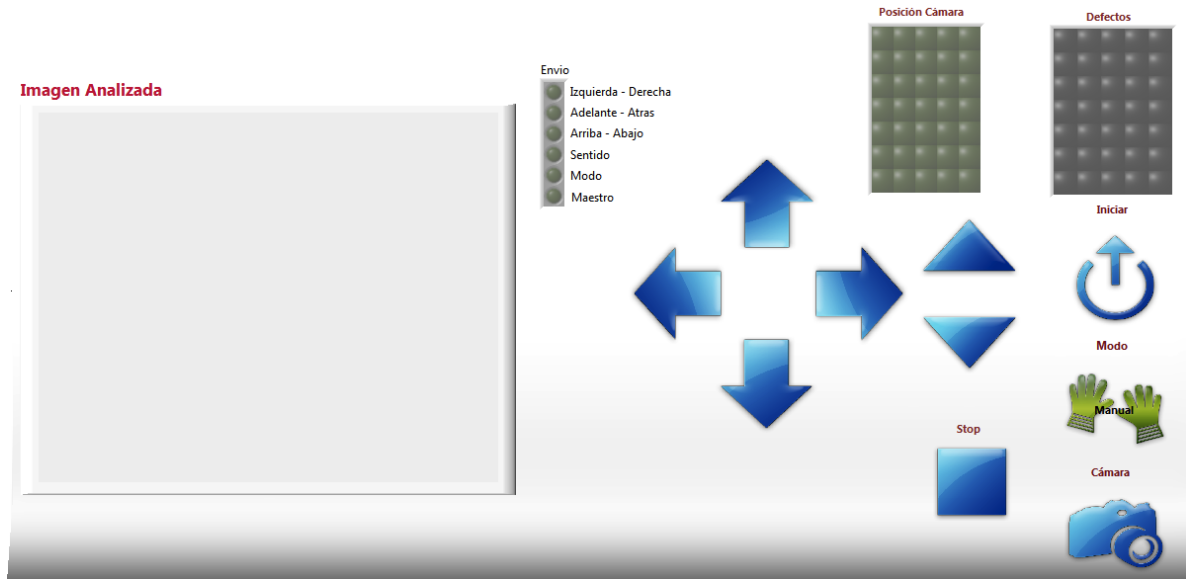


Figura 4.34: Panel frontal del proyecto.

4.3.3. Programación del controlador

El arduino fue programado mediante el software "Arduino 0019", este se utilizó para la creación y compilación del programa. Como se mencionó anteriormente del PXI se obtienen 6 señales de control, esta señales servirán para ordenar al arduino cuando comenzar y cuando dar los pasos establecidos para que los motores lleguen al punto deseado. Como resultado el arduino generará ocho señales, de las cuales siete señales serán enviadas a la etapa de potencia y una señal al PXI, esta última señal servirá para indicarle al PXI que el arduino terminó de ejecutar la orden enviada.

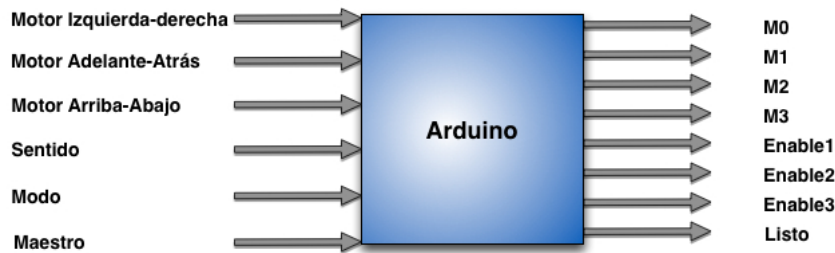


Figura 4.35: Diagrama de señales de entrada y salida del arduino.

Lógica de programación

La lógica de programación se presenta en los diagramas de estado de las Figuras(4.36 y 4.37), pero a continuación se explicara de forma detallada su contenido:

Manual

- El arduino esperará a que el PXI le envíe una señal para que este comience el programa (Maestro). Cuando el PXI llame al arduino este primero tendrá que saber si el programa que tiene que ejecutar es el modo manual o automático.

- En el modo manual el arduino esperara las señales para saber que motores mover y en que sentido. Para esto se leerán cuatro señales digitales (Izquierda_Derecha, Adelante_Atrás, Arriba_Abajo, Sentido), recordando que solo se moverá un motor a la vez. En la siguiente tabla se muestra las combinaciones posibles y la acción a realizar:

<i>Izq_derecha</i>	<i>Adel_atrás</i>	<i>Arrib_abj</i>	<i>Sentido</i>	<i>Acción</i>
0	0	0	0	Nada
0	0	0	1	Nada
0	0	1	0	Abajo
0	0	1	1	Arriba
0	1	0	0	Atrás
0	1	0	1	Adelante
0	1	1	0	Nada
0	1	1	1	Nada
1	0	0	0	Derecha
1	0	0	1	Izquierda
⋮	⋮	⋮	⋮	⋮
1	1	1	1	Nada

tabla 4.8: Tabla de verdad del modo manual.

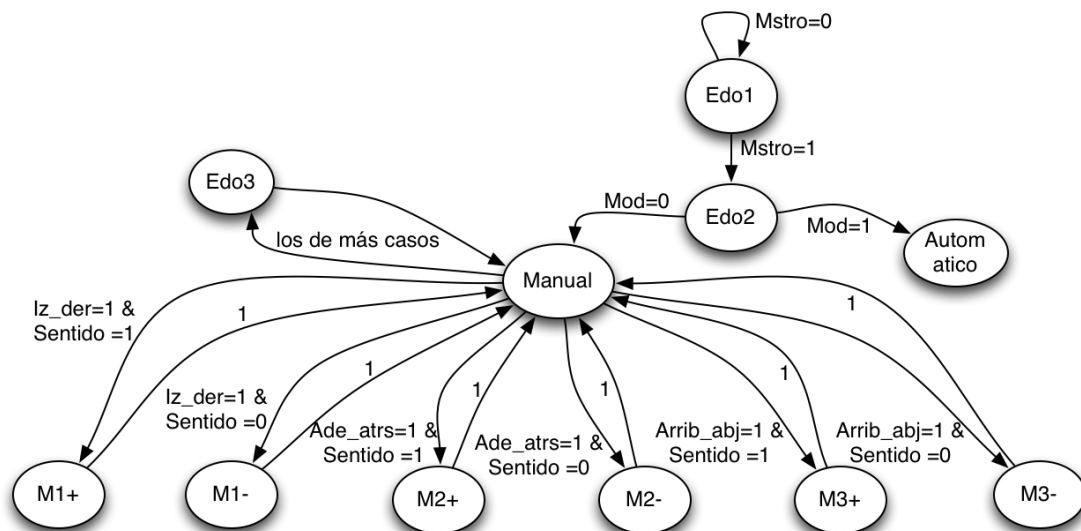


Figura 4.36: Diagrama de estados para la parte manual del programa.

<i>Modo Manual</i>	
<i>Edo1</i>	M0=M1=M2=M3=Enable1=Enable2=Enable3=Listo = LOW
<i>Edo2</i>	Leer modo (automático o manual)
<i>Manual</i>	Leer señales (Izq-der, Ade-atra, Arri-abajo, Sentido)
<i>M1+</i>	Mover motor 1 positivo, Enable1=High, Enable2=Enable3=LOW
<i>M1-</i>	Mover motor 1 negativo, Enable1=High, Enable2=Enable3=LOW
<i>M2+</i>	Mover motor 2 positivo, Enable2=High, Enable1=Enable3=LOW
<i>M2-</i>	Mover motor 2 negativo, Enable2=High, Enable1=Enable3=LOW
<i>M3+</i>	Mover motor 3 positivo, Enable3=High, Enable1=Enable2=LOW
<i>M3-</i>	Mover motor 3 negativo, Enable3=High, Enable1=Enable2=LOW
<i>Edo3</i>	Mo, M1, M2, M3, Enable1, Enable2, Enable 3, Listo = LOW

tabla 4.9: Tabla de los estados con sus correspondientes salidas y lecturas.

Automático

- Si el modo seleccionado es automático, el arduino leerá las tres señales: Izquierda_Derecha, Adelante_Atrás y Sentido, pero a diferencia del modo manual se generarán dos contadores uno para Derecha y otro para Arriba-Abajo, estos contadores establecen los números de pasos necesarios para que el manipulador avance una posición (Para ver el cálculo de los número de pasos para cada motor véase Apéndice C), cuando esta condición se cumpla el arduino mandará una señal para avisarle al PXI que se terminaron de dar los pasos que se le ordenaron. En la Figura(4.37) se muestra el diagrama de estados para el modo automático y en la Tabla(4.10) observamos las posibles combinaciones y la acción a realizar.

Izq_der	Adel_atrs	Sentido	Contador1 (Cont=925)	Contador2 (Cont2=1450)	Acción
0	0	0	0	0	Nada
⋮	⋮	⋮	⋮	⋮	⋮
0	1	0	0	0	Atrás
0	1	0	0	1	Nada
0	1	0	1	0	Se avanzó un cuadro hacia atrás
0	1	0	1	1	Nada
0	1	1	0	0	Adelante
0	1	1	0	1	Nada
0	1	1	1	0	Se caminó un cuadro hacia adelante
0	1	1	1	1	Nada
1	0	0	0	0	Derecha
1	0	0	0	1	Se caminó un cuadro hacia la derecha
⋮	⋮	⋮	⋮	⋮	Nada

tabla 4.10: Tabla de verdad del modo automático.

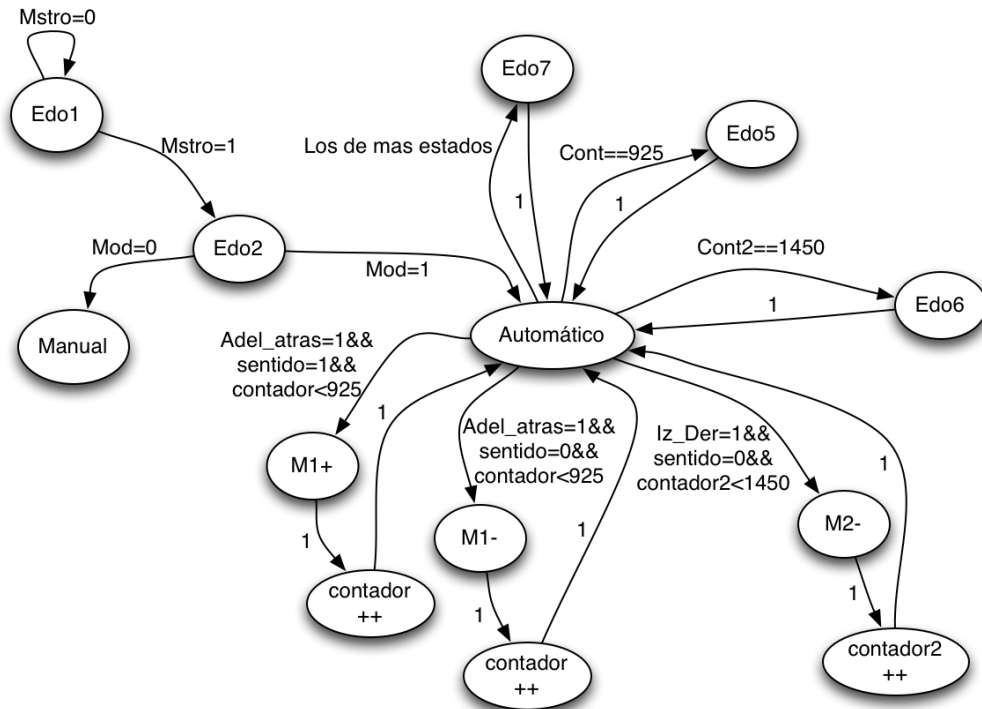


Figura 4.37: Diagrama de estados para la parte automática del programa.

<i>Modo Automático</i>	
<i>Edo1</i>	Mo=M1=M2=M3=Enable1=Enable2=Enable3=Listo = LOW
<i>Edo2</i>	Leer modo (automático o manual)
<i>Automático</i>	Leer señales (Izq-Der, Adel-atras, sentido)
<i>M1+</i>	Mover motor 1 positivo, Enable1=High, Enable2=Enable3=LOW
<i>M1-</i>	Mover motor 1 negativo, Enable1=High, Enable2=Enable3=LOW
<i>M2+</i>	Mover motor 2 positivo, Enable2=High, Enable1=Enable3=LOW
<i>M2-</i>	Mover motor 2 negativo, Enable2=High, Enable1=Enable3=LOW
<i>Edo5</i>	Listo=High / Listo=Low, Contador=0
<i>Edo6</i>	Listo=High / Listo=Low, Contador2=0
<i>Edo7</i>	Mo, M1, M2, M3, Enable1, Enable2, Enable 3, Listo = LOW

tabla 4.11: Tabla de los estados con sus correspondientes salidas y lecturas del modo automático.

Capítulo 5

Resultados experimentales

A continuación, se presentan los resultados experimentales que se han considerado de más valor académico para ilustrar el desempeño del modelo de textura e iluminación implementada. Los experimentos fueron realizados sobre imágenes reales capturadas en tiempo real. El objetivo de la experimentación es tomar una serie de muestras de pieles curtida de bovino y determinar si tienen defecto. Se utilizan 7 pieles curtidas de bovino, en las cuales se toman 35 muestras por cada piel. Se examinarán un total de 245 muestras con una resolución de $640 \times 480 \text{ pixels}$ en escala de grises, estas imágenes capturadas son analizadas y posteriormente guardadas en formato jpg.

Se hace uso del módulo de visión *NI training texture*, el cual proporciona un ambiente interactivo y de viable navegación para configurar los parámetros en la detección de defectos. Para entrenar al clasificador se utilizan 15 muestras de una piel libre de defectos, así como tres tipos de defectos encontrados en las otras pieles con la finalidad de verificar que estos se distinguen en las características de Haralick, este clasificador es guardado en formato .clf e introducido en el programa realizado en LabVIEW[®].

Como se expuso en el capítulo 4, existe la posibilidad de encontrar distintos tipos de defectos en pieles curtidas de bovino, por lo tanto se experimentará qué tipos de defectos el sistema de inspección es capaz de reconocer. El trabajo aquí presentado esta diseñado para el tipo de piel seleccionado, es decir, que si en el momento de realizar la inspección automática, se desea que el sistema inspeccione otro tipo de piel, se tendrá que detener completamente el proceso para configurar al clasificador y ajustar la iluminación y así poder inspeccionar la piel curtida deseada.

La parte final del proyecto de tesis involucró pruebas, donde se tuvieron en cuenta diferentes aspectos del sistema como:

- Entrenamiento del archivo clasificador.
- Repetibilidad del manipulador cartesiano.
- Resultados obtenidos.

5.1. Entrenamiento del archivo clasificador de texturas

La interfaz de entrenamiento para la detección de defectos en texturas (*NI training texture*) consta de dos partes importantes :

- Caracterización de defectos.
- Clasificador de textura.

NI training texture utiliza los ajustes de la caracterización de defectos para extraer las características, con el fin de distinguir los defectos en la piel curtida. La caracterización de defectos consiste en tres pasos:

1. Basado en la selección de un tipo de *wavelet*, el clasificador lleva a cabo una descomposición *wavelet frame* de la imagen para calcular las imágenes de las sub-bandas.

2. El clasificador mueve una ventana a través de las imágenes de las sub-bandas del *wavelet* seleccionado y genera una matriz de co-ocurrencia en escala de grises para cada posición de la ventana.
3. Con base en la matriz de co-ocurrencia en escala de gris, el clasificador calcula estadística de segundo orden (características de haralick) para cada sub-banda del *wavelet* seleccionado. Las características de Haralick para todas la sub-bandas de los *wavelets* seleccionados se concatenan para representar la imagen actual. Es importante recalcar que el clasificador de textura no detecta defectos si estos no son visibles en el espacio de características de Haralick.

Para esto se carga una muestra de textura que tenga algún tipo de defecto y se utiliza la pestaña de caracterización de defectos para configurar los parámetros de este. El *wavelet* seleccionado es un tipo Haar con las siguientes opciones para obtener las características de Haralick:

<i>Co-ocurrence level</i>	64
Vector de desplazamiento	$x\Delta = 2 \quad y\Delta = 0$
Tamaño de ventana en x	11
Tamaño de ventana en y	11
Tamaño de paso	0

Se utiliza un nivel de co-ocurrencia de 64 para aumentar el tamaño de la matriz de co-ocurrencia generada y el número de intensidades de pixel utilizados para caracterizar la textura y algunos defectos en ella. Aunque un nivel alto en el nivel de co-ocurrencia requiere más tiempo de procesamiento este nos permite que el clasificador distinga los defectos en la textura más fácilmente. El vector de desplazamiento se ajusto mediante ensayo y error, encontrando la mejor distinción de los defectos para el vector de desplazamiento propuesto. En general, se especifica un tamaño de ventana que coincida con el defecto más pequeño en la textura que se desee detectar, para nuestro caso el defecto más pequeño que se encuentra en la piel curtida de bovino es de $.1[mm]$.

Como se mencionó anteriormente las pieles curtidas de bovino que se han utilizado para llevar a cabo las pruebas son siete. De estas, se ha tomado una para extraer texturas muestra para el entrenamiento del clasificador (es decir, las muestras correctas que se utilizan como comparación en el software de inspección). Las demás pieles curtidas sufren algún problema, como se indicará posteriormente. En la Figura (5.1) se muestra el defecto escogido para seleccionar la sub-banda, donde a través de las características de haralick el defecto puede visualizarse. Este es un defecto con profundidad.

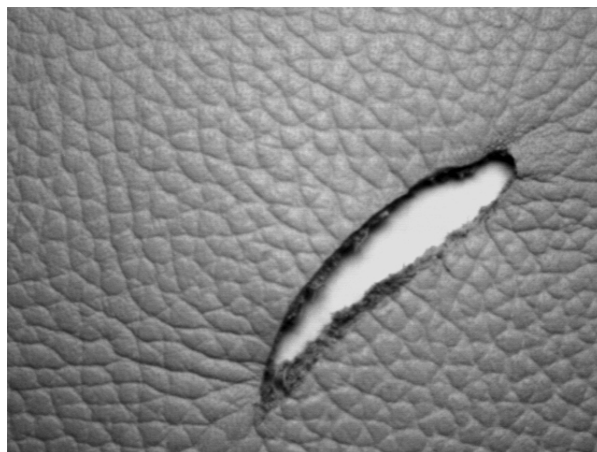


Figura 5.1: Muestra 1 con defecto para el entrenamiento del *NI training texture*.

A partir de esto se pueden obtener las sub-bandas para el *wavelet* seleccionado (Figura 5.2). Al observar las características de haralick para las diferentes sub-bandas la mayor diferenciación del defecto se encontró en la sub-banda *low – low – low* (Figura 5.3), este defecto es claramente visible en las características de haralick, por lo tanto se selecciona esta sub-banda.

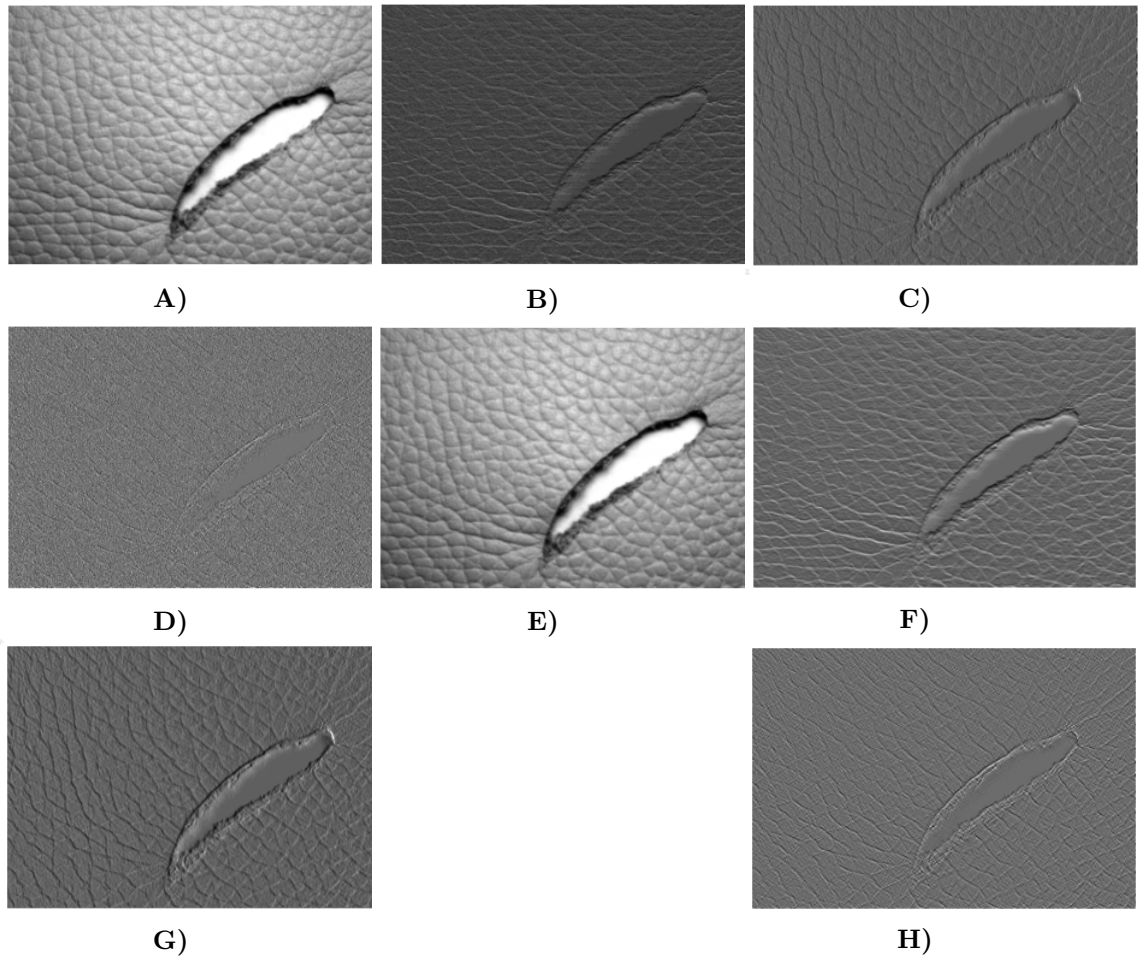


Figura 5.2: Sub-bandas del *wavelet* tipo Haar para la muestra 1. A) *Low-Low*. B) *Low-High*. C) *High-Low*. D) *High-High*. E) *Low-Low-Low* F) *Low-Low-High* G) *Low-High-Low*. H) *Low-High-High*.

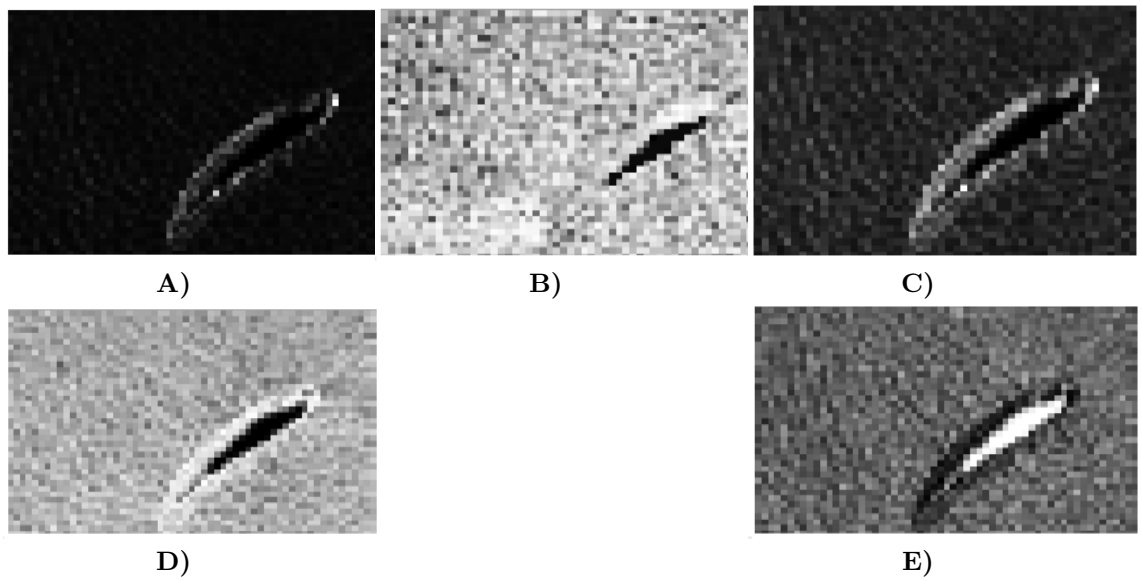


Figura 5.3: Características de Haralick para la muestra 1. A) Contraste. B) Correlación. C) Disimilitud. D) Entropía. E) Homogeneidad.

El siguiente defecto seleccionado es un defecto sin profundidad (ver Figura 5.4).



Figura 5.4: Muestra 2 con defecto para el entrenamiento del *NI training texture*.

Al introducir la imagen al entrenador se pueden observar visualmente las sub-bandas para el *wavelet* seleccionado (Figura 5.5). La mayor diferenciación del defecto también se encontró en la sub-banda *low – low – low*, este defecto es claramente visible en las características de haralick (ver Figura 5.6).

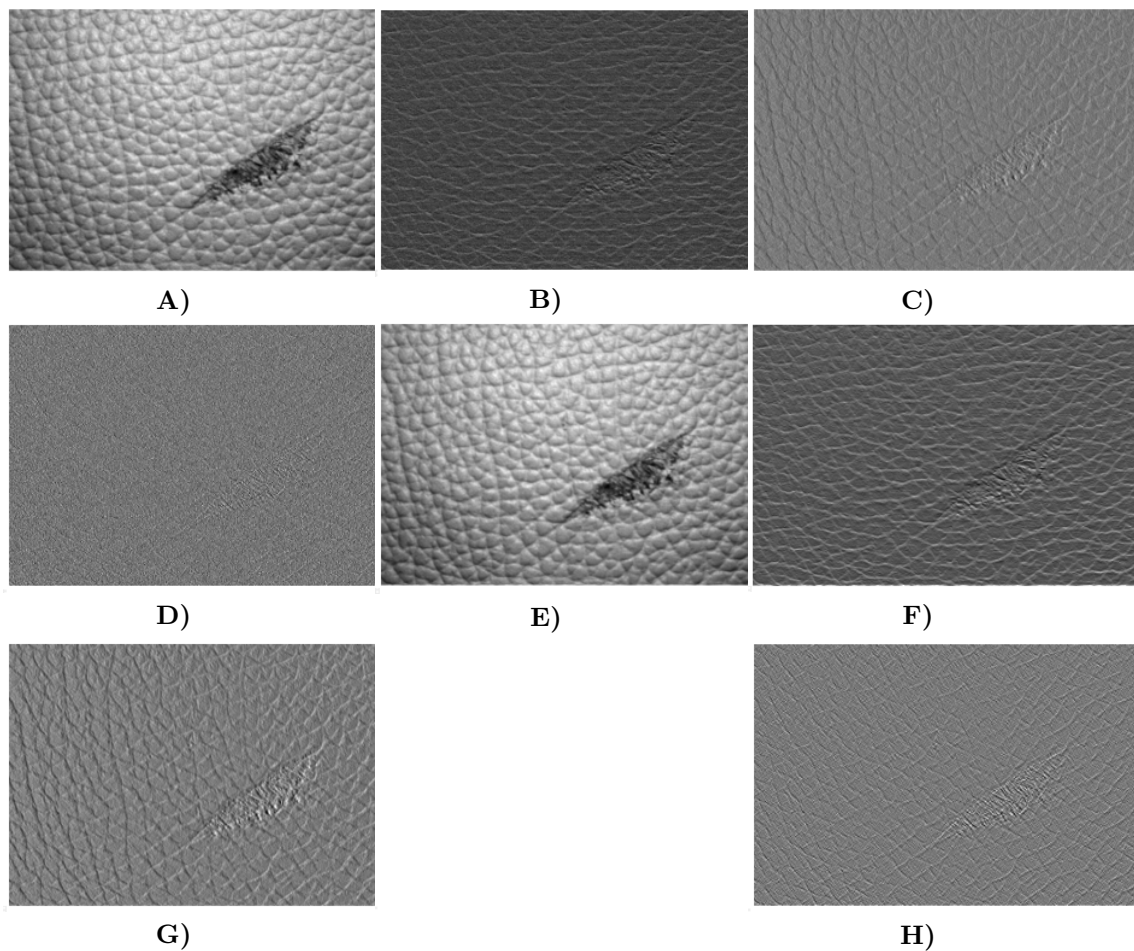


Figura 5.5: Sub-bandas del wavelet tipo Haar para la muestra 2. A) *Low-Low*. B) *Low-High*. C) *High-Low*. D) *High-High*. E) *Low-Low-Low* F) *Low-Low-High* G) *Low-High-Low*. H) *Low-High-High*.

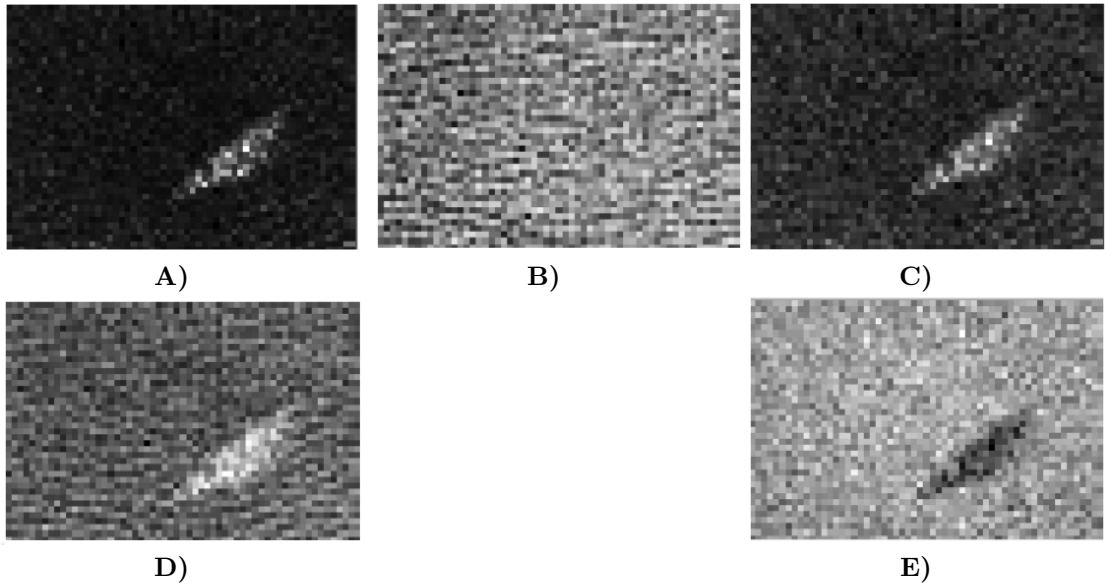


Figura 5.6: Características de Haralick para la muestra 2. A) Contraste. B) Correlación. C) Disimilitud. D) Entropía. E) Homogeneidad.

Por último se probará con otro defecto para verificar si este también puede ser visualizado en la misma sub-banda. Esta muestra se forma por defectos pequeños sin profundidad alguna distribuidos sobre el área de la imagen (ver Figura 5.8).

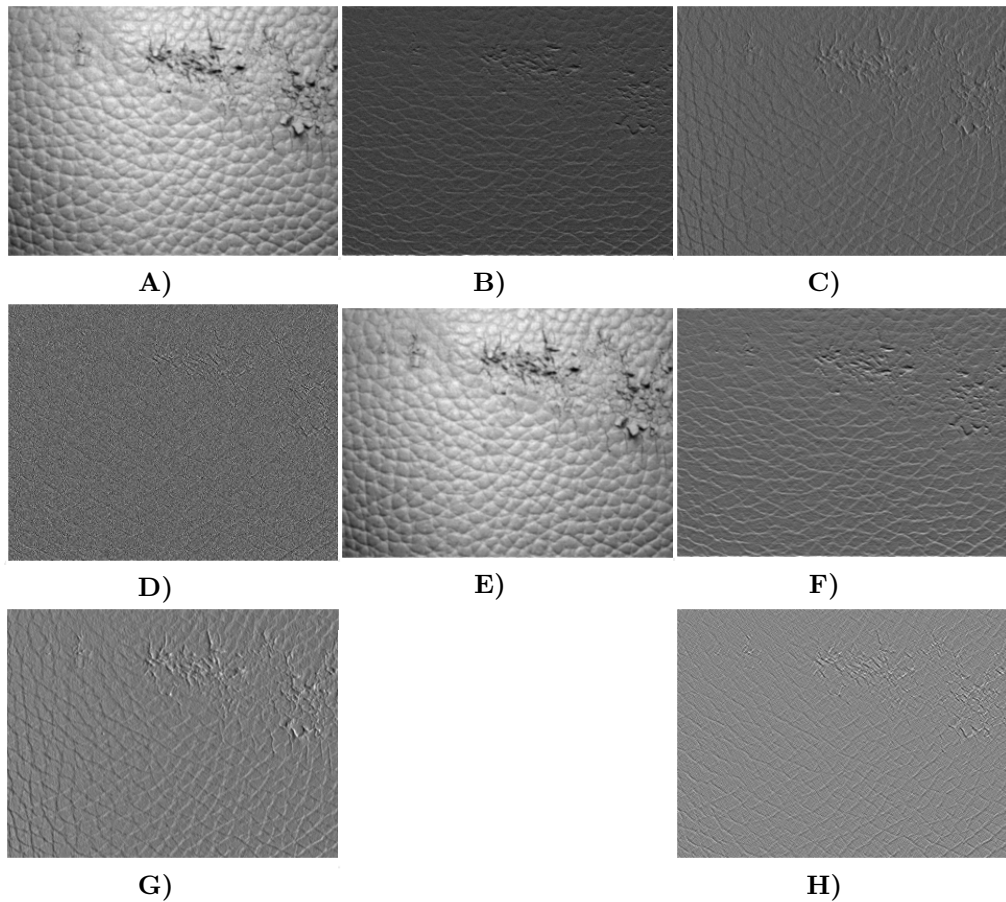


Figura 5.7: Sub-bandas. A) *Low-Low*. B) *Low-High*. C) *High-Low*. D) *High-High*. E) *Low-Low-Low*. F) *Low-Low-High*. G) *Low-High-Low*. H) *Low-High-High*.



Figura 5.8: Tercera muestra con defecto para el entrenamiento del *NI training texture*.

Al igual que los defectos anteriores la mayor diferenciación del defecto se encontró también en la misma sub-banda, por lo tanto se selecciona esta sub-banda. Al seleccionar una sub-banda el vector de características para representar una textura muestra será de 5.

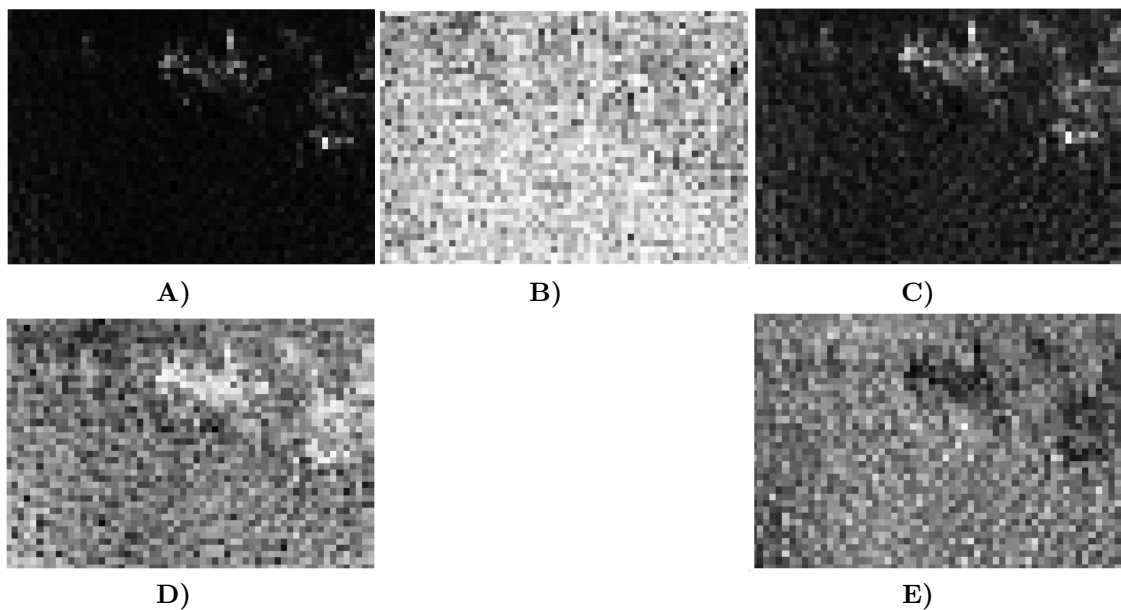


Figura 5.9: Características de Haralick. A) Contraste. B) Correlación. C) Disimilitud. D) Entropía. E) Homogeneidad

Después de haber seleccionado la sub-banda, se utiliza la parte del clasificador de textura para entrenar al SVM. Nuestra aplicación para la detección de defectos requiere de una sola clase, por lo tanto se utiliza un clasificador SVM *one-class* y los parámetros del clasificador se dejan por *default*.

Parámetros SVM	
Tolerancia	0.001
nu	0.1
Kernel	RBF
Gama	1

Un total de 15 muestras sin defecto son utilizadas para el entrenamiento del clasificador (ver Figura 5.10). Estas se pueden ver detalladamente en el Apéndice A.

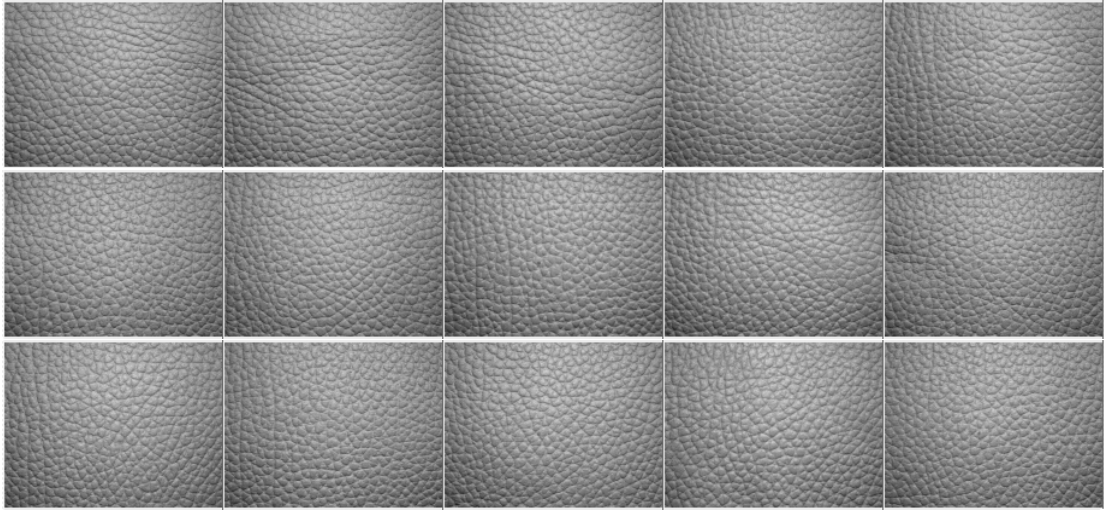


Figura 5.10: Texturas muestra sin defecto.

Al probar el clasificador con los parámetros seleccionados se puede comprobar que este puede detectar los defectos. A continuación se muestran las imágenes de salida tras aplicar el clasificador a las muestras con defecto utilizadas para el entrenamiento.



Figura 5.11: Imágenes binarias de salida del clasificador *NI training interface*

5.2. Prueba de repetibilidad

Para verificar la repetibilidad¹ del manipulador cartesiano, se diseñó una plantilla del espacio de trabajo con sus correspondientes divisiones, donde cada división tiene una pequeña circunferencia en el centro, esta plantilla fue impresa y fijada en el plano xy del manipulador. También se diseñó un programa en LabVIEW[®] con el cual se manda al manipulador a una posición seleccionada, al llegar a esta se toma una imagen generando una circunferencia de color rojo que marca el centro de la misma. Esta circunferencia servirá como referencia para saber que tanto se desfasa con respecto a su posición original. Las pruebas partieron desde el *home* hasta la posición seleccionada.

Se escogieron cuatro posiciones en el espacio de trabajo (Figura 5.12), y para cada posición se realizaron 4 pruebas. Las imágenes tomadas fueron comparadas y los resultados se muestran en las Figuras 5.13, 5.14, 5.15 y 5.16. En los cálculos realizados (Véase Apéndice D) podemos observar que el sistema llega satisfactoriamente a las posiciones indicadas, se tiene un pequeño desfase significativo, pero debemos recalcar que el radio del círculo que encierra todos los puntos en las pruebas es de .5 [mm] y todos los puntos caen adentro de esta circunferencia, teniendo un aceptable repetibilidad para las necesidades de este proyecto. A continuación se establece las especificaciones del manipulador

¹La capacidad para desplazarse a un punto determinado del espacio con un error mínimo.

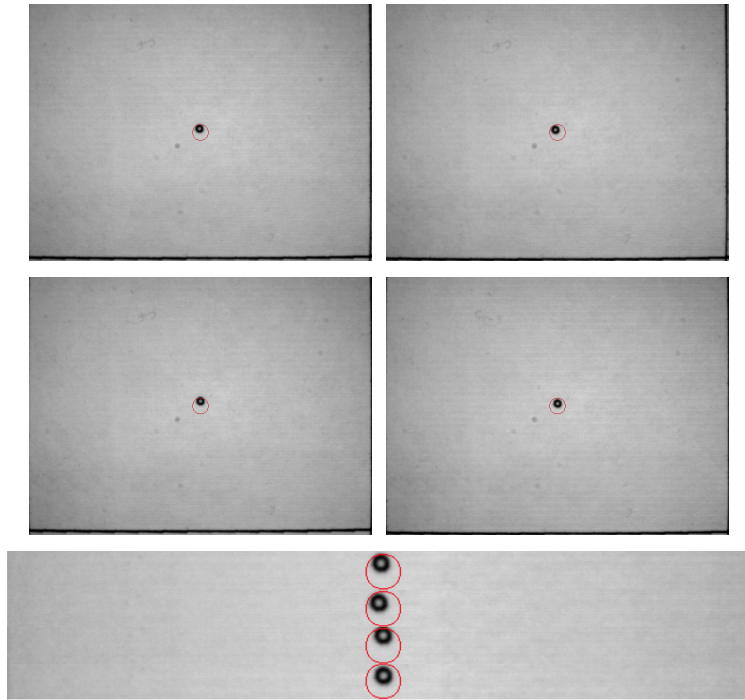


Figura 5.14: Resultado repetibilidad Posición 2.

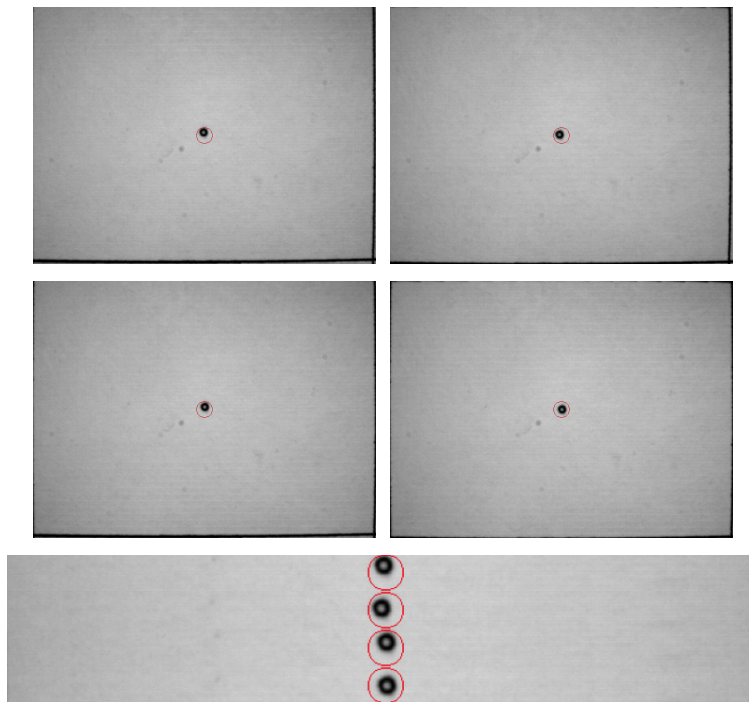


Figura 5.15: Resultado repetibilidad Posición 3.

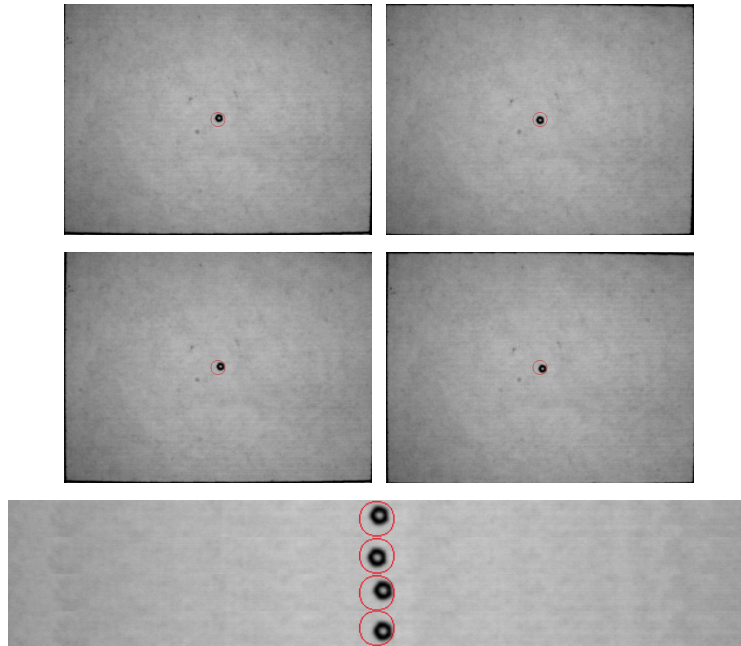


Figura 5.16: Resultado repetibilidad Posición 4.

5.3. Resultados obtenidos

El modelo propuesto fue aplicado para las 245 muestras con diferentes tipos de defectos. Se tomaron muestras de 640×480 *pixels* en escala de grises. Los parámetros que se mencionaron anteriormente fueron fijados a lo largo de los experimentos sobre todas las texturas muestras. El tiempo de cálculo en la detección de defectos en cada muestra es de aproximadamente de 1.5 s. Las imágenes de cada porción son analizadas y guardas para que al finalizar la toma de todas las muestras, estas sean unidas generando una imagen completa de la información general de la piel bajo inspección, esta imagen tiene una medida de 1200×1256 *pixels*. El programa muestra en el panel frontal por medio de indicadores en pantalla en que porciones de la piel se encontraron defectos(rojo) y en cuales no (verde).

A continuación se describen los defectos más significativos encontrados en el análisis de cada piel curtida de bovino:

- En la piel 1 el programa detectó un defecto, este tipo de defecto tiene profundidad y el programa de detección lo perfila de una forma correcta. Este defecto es resaltado y señalado en la muestra tomada (Figura 5.17).

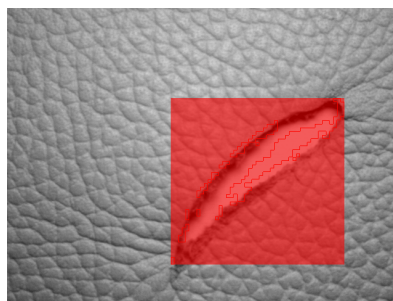


Figura 5.17: Defecto encontrado al analizar la Piel curtida No1.

- En la piel 2 al igual que en la piel anterior se detecto un defecto, pero a diferencia del anterior, este tipo de defecto no tiene profundidad, este defecto es detectado satisfactoriamente (Figura

5.18). En las figuras 5.25 y 5.26 se puede observar la piel curtida completa y el panel frontal después del análisis.

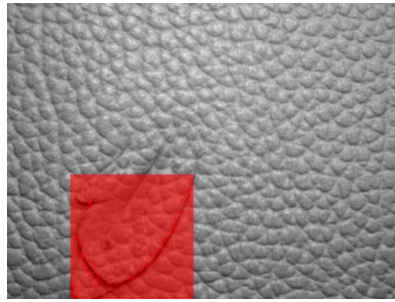


Figura 5.18: Defecto encontrado al analizar la Piel curtida No2.

- En la piel 3 se tiene una orilla este es un defecto con profundidad evidente (Figura5.19), el defecto es detectado y señalado en la imagen así como en el panel frontal (Figuras 5.27 y 5.28).

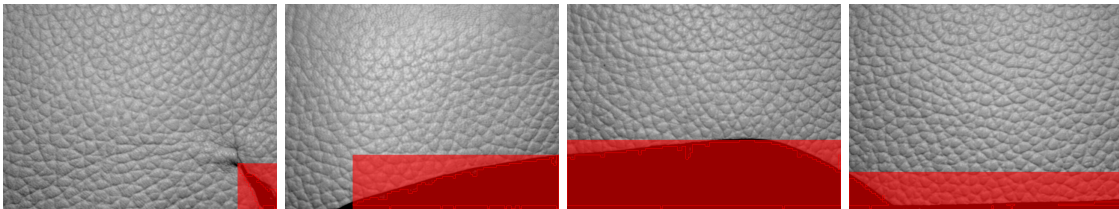


Figura 5.19: Defectos encontrado al analizar la Piel curtida No3.

- En la piel 4 se tiene tres tipos de defectos. El primero es un punto con un diámetro mayor a 1mm, una marca de control y orillas. En la Figura 5.20 se muestran los defectos más significativos. Para ver el resultado final ver Figuras (5.29 y 5.30).

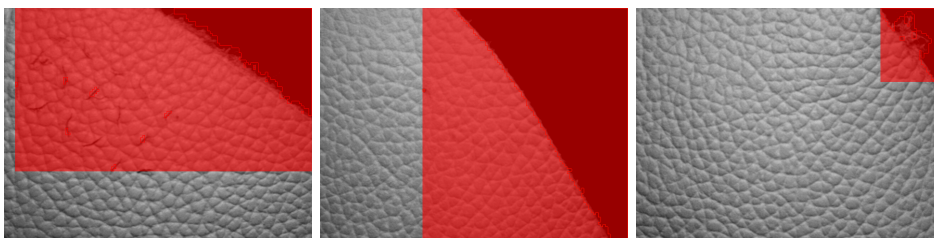


Figura 5.20: Defectos encontrados al analizar la Piel curtida No4.

- En la piel 5 se pueden observar cuatro tipos de defectos. Dos defectos sin profundidad, una marca que no se alcanza a perfilar y bordes. De estos el programa pudo detectar solo 3 de ellos (Figura 5.21).

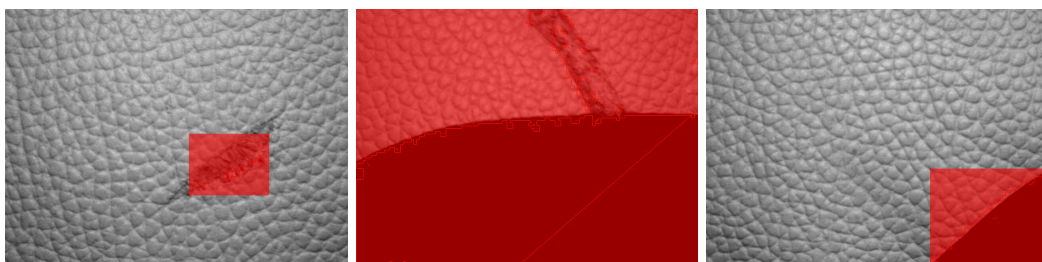


Figura 5.21: Defectos encontrados al analizar la Piel curtida No5.

- La piel 6 presenta una sección de pequeños defectos que forman una área considerable en la porción tomada (Figura 5.22). Estos son detectados y mostrados en pantalla (Figuras 5.33 y 5.34).

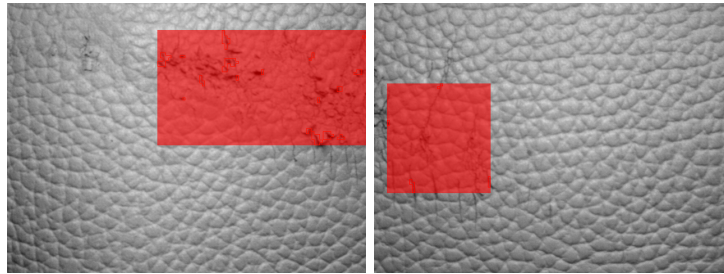


Figura 5.22: Defectos encontrados al analizar la Piel curtida No6.

A continuación se observa las imágenes finales después del escaneado de la piel curtida de bovino para cada piel, así como el panel frontal de cada una de ellas donde se muestran las posiciones de los defectos encontrados en cada prueba.

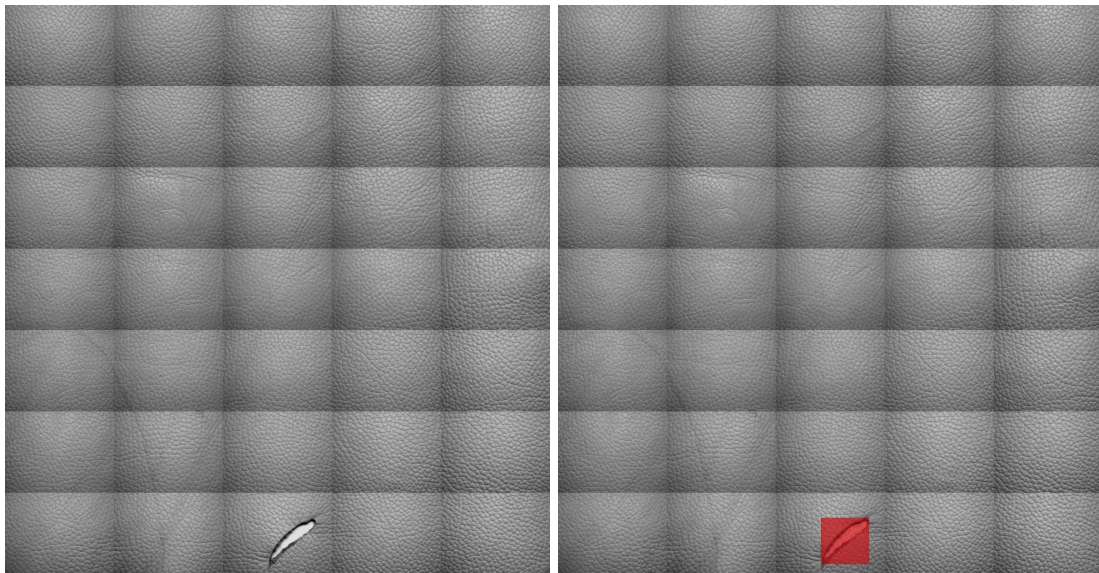


Figura 5.23: Piel curtida No 1 antes y después de ser analizada.

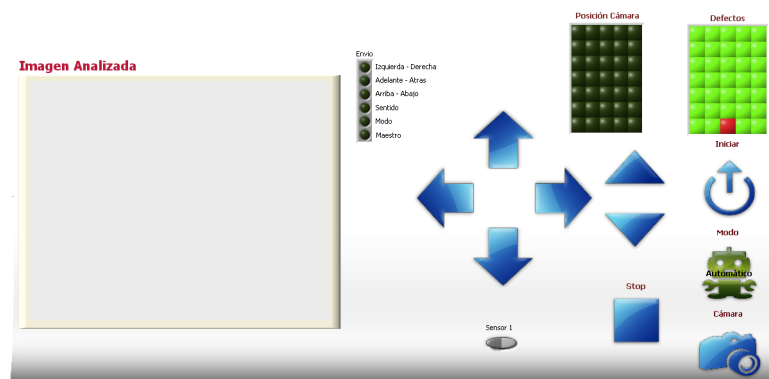


Figura 5.24: Panel Frontal después de terminar el análisis de la Piel curtida No1.

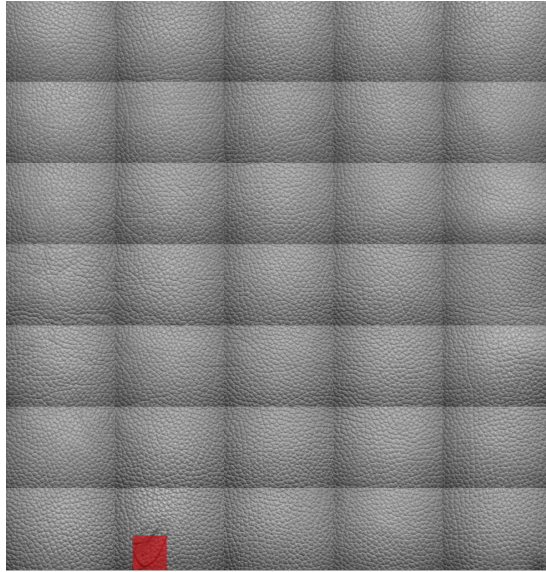


Figura 5.25: Piel curtida No2 después de ser analizada.

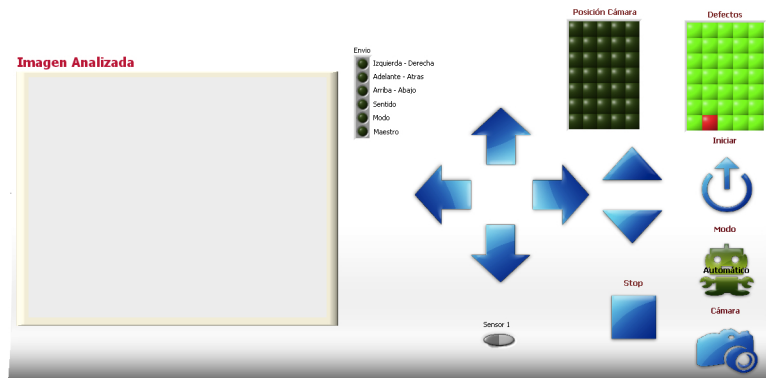


Figura 5.26: Panel Frontal después de terminar el análisis de la Piel curtida No2.

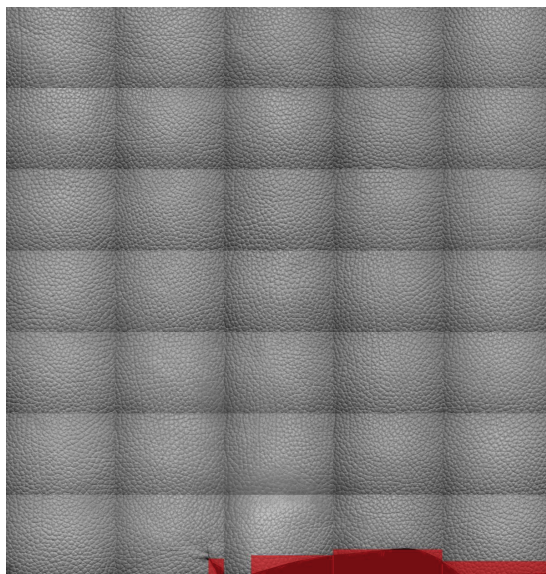


Figura 5.27: Piel curtida No3 después de ser analizada.

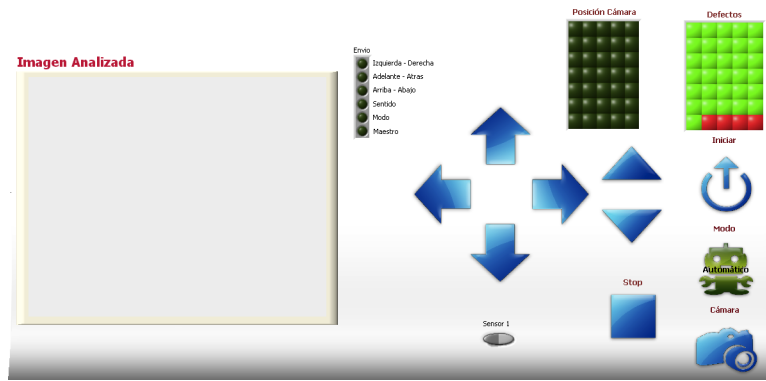


Figura 5.28: Panel Frontal después de terminar el análisis de la Piel curtida No3.

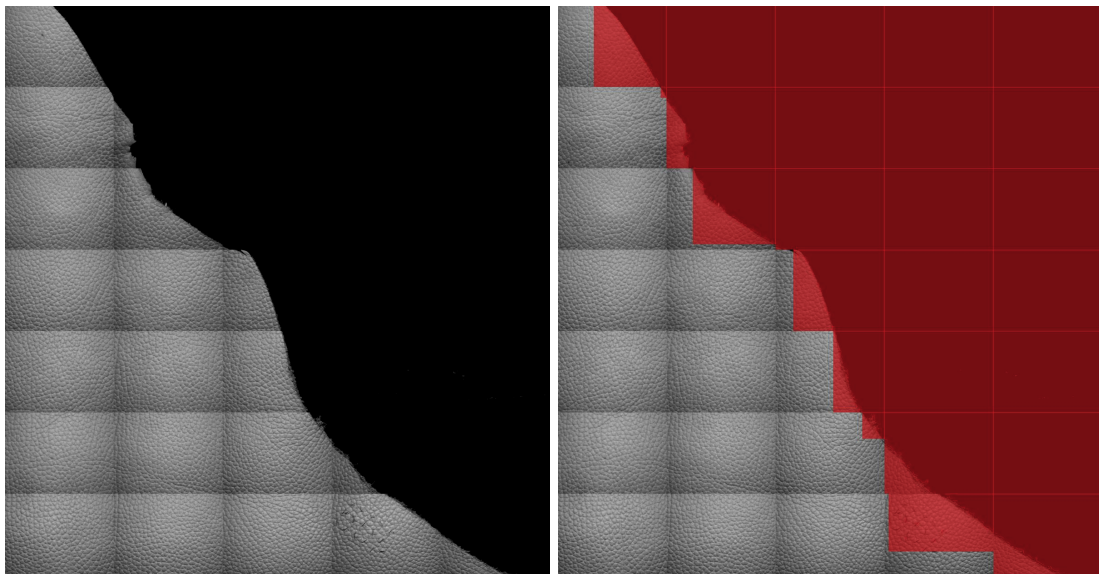


Figura 5.29: Piel curtida No4 antes y después de ser analizada.

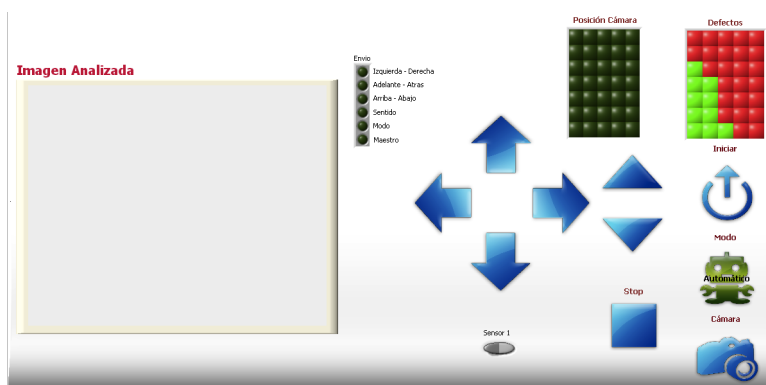


Figura 5.30: Panel Frontal después de terminar el análisis de la Piel curtida No4.

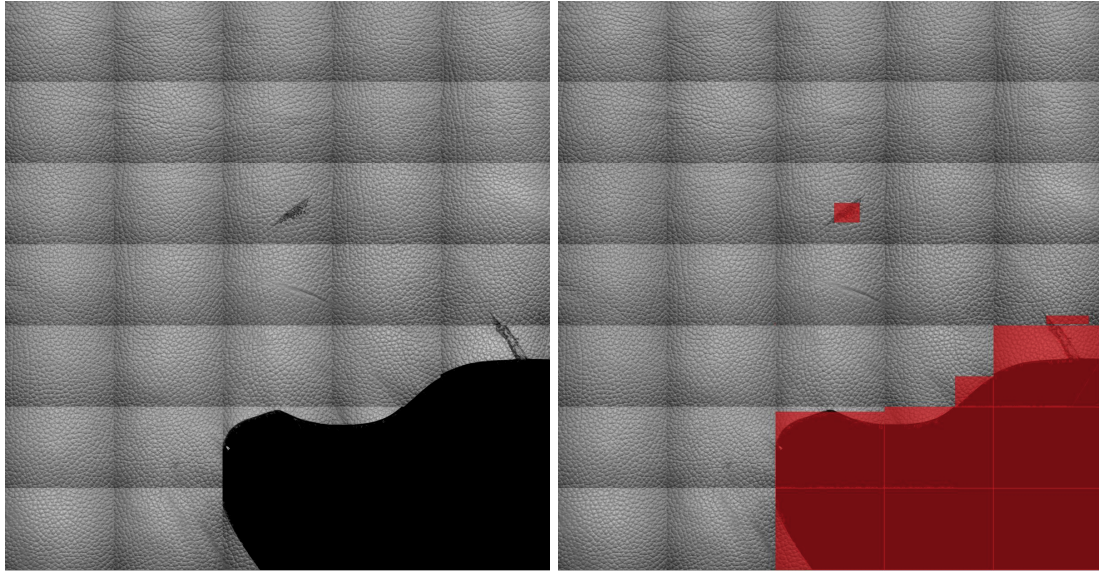


Figura 5.31: Piel curtida No5 antes y después de ser analizada.

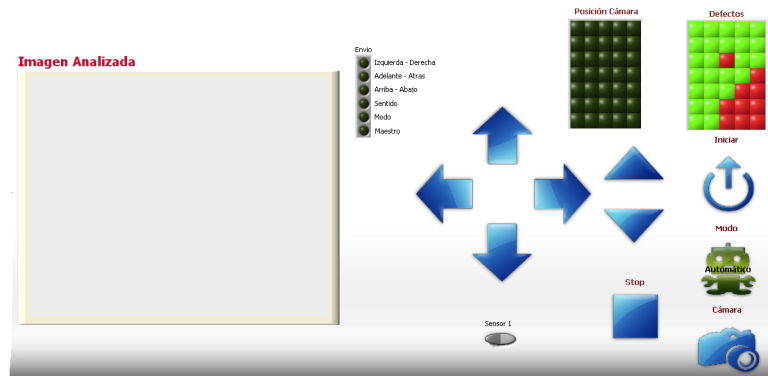


Figura 5.32: Panel Frontal después de terminar el análisis de la Piel curtida No5.

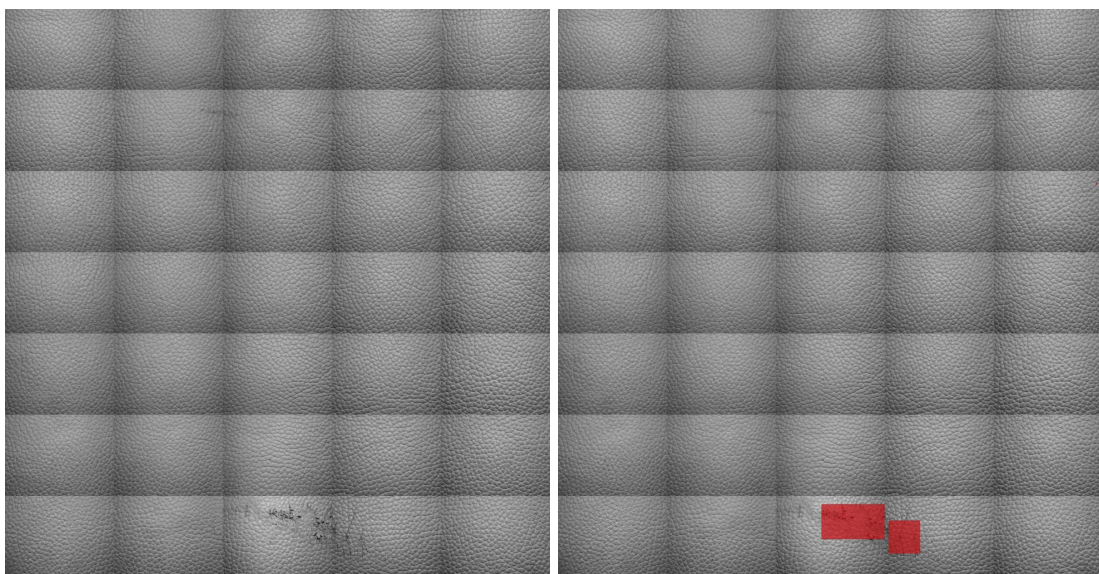


Figura 5.33: Piel curtida No6 antes y después de ser analizada..

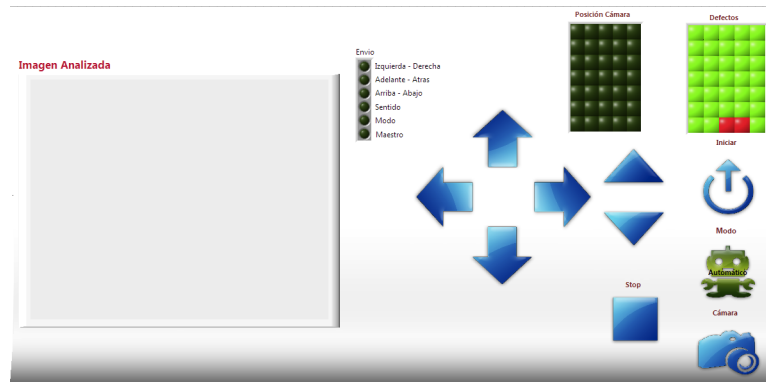


Figura 5.34: Panel Frontal después de terminar el análisis de la Piel curtida No6.

Se puede observar que los defectos en las pieles curtidas de bovino se detectaron satisfactoriamente, pero en el caso de la piel curtida No5 se observa un defecto que es visible pero el algoritmo no lo detecta, se recurrirá a el entrenador *NI training interface* para observar sus características de haralick y determinar por qué este defecto no fue detectado.



Figura 5.35: Defecto proveniente de la piel curtida No5 que no fue detectado por el algoritmo.

Como se puede ver en las características de haralick el defecto no se alcanza a diferenciar en las características de haralick, razón por la cual al ser analizada esta piel curtida de bovino, dicho defecto no pueda ser detectado. El problema se centra en que si se aumentan las características del clasificador para poder detectar este defecto se tengan falsos negativos no deseados durante el análisis.

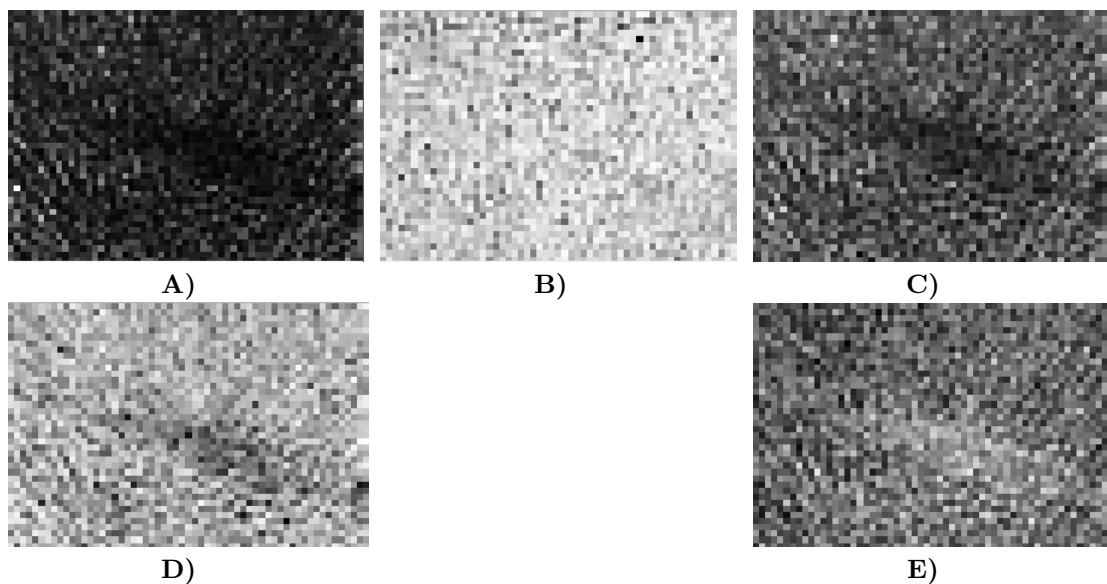


Figura 5.36: Características de Haralick. A) Contraste. B) Correlación. C) Disimilitud. D) Entropía. E) Homogeneidad.

Capítulo 6

Conclusiones y trabajo a futuro

En esta capítulo se presentan las conclusiones de la tesis y las principales aportaciones que ofrece, por último se propone algunas líneas de investigación en las que sería interesante profundizar en el futuro.

6.1. Conclusiones

Como conclusiones al proyecto partimos del objetivo principal, que era crear un sistema mecánico para la detección de defectos en pieles curtidas de bovino a través de visión artificial, se puede decir que se ha realizado con éxito. A pesar de las limitaciones del hardware, fue posible utilizar este mismo y adaptarlo a la necesidad del proyecto obteniendo resultados satisfactorios en la experimentación.

La tarea de detectar defectos en texturas es un trabajo desafiante, mucho más cuando este tipo de texturas son del tipo aleatorio, se partió de este gran problema y se simplificó a través de investigación, encontrado un módulo de LabVIEW[®] cuyo desarrollo es reciente y no se tiene investigaciones en las que se haya aplicado, el problema se centró en la investigación del uso de este programa para caracterizar las pieles curtidas de bovino, así como su implementación en la programación en LabVIEW[®] para que este estuviera incorporado en el programa general y así tener una integración total entre el software y hardware del proyecto.

Hablando de la optimización, el programa cumple con esta característica, ya que al dividir a toda la piel curtida de bovino puesta en el espacio de trabajo en pequeñas muestras, que después fueron unir las en una sola imagen mostrando los defectos encontrados, se evitó procesar gran cantidad de información dándole así mayor resolución a cada imagen tomada en cada porción de la piel curtida. Con respecto al software desarrollado para este proyecto se entrega y despliega información de partes con defectos en pantalla, obteniendo al finalizar del proceso de análisis información de alto nivel (binaria) la cual puede servir para procesos posteriores del análisis de la piel curtida.

La iluminación es de vital importancia en cualquier proyecto que implica un sistema de visión artificial. Por lo tanto se desarrolló una cámara de aislamiento para evitar las variaciones en cada imagen y que estas tuvieran una iluminación homogénea no importando en que posición se encuentre el manipulador. Hablando de la iluminación, no se encontró mucha información al iniciar el proyecto sobre esta aplicada a texturas naturales, por lo cual se recurrió a la investigación y experimentación. Se encontró que la iluminación en forma directa sobre la textura resaltan las texturas, relieves y defectos del objeto iluminado, todo esto debido a que cualquier relieve por muy pequeño que sea, produce una sombra muy definida. Se experimentó con esta iluminación con diferentes ángulos para comprobar que con esta iluminación los defectos se intensificaron en la adquisición. Al tener una correcta iluminación la imagen que se obtuvo fue adecuada, por lo cual no se tuvo la necesidad de emplear filtros, aumentando la velocidad del proceso de inspección.

Con respecto a los resultados experimentales fue posible comprobar que a través del algoritmo utilizado, el cual combina el análisis multi-escala y la matriz de co-ocurrencia se pudieron extraer

características que se calculan en un tiempo aceptable (1.5 s) y con la cual la mayoría de los defectos que se presentaron en las pieles curtidas fueron detectados de manera satisfactoria, con excepción de los puntos de alfiler, los cuales son muy pequeños (menores a .1 mm) y que con la resolución de la cámara no fue suficiente para poder detectarlos. En relación al otro defecto que se presentó en la piel No5, que se mencionó con detalle, que no fue posible detectar, este tiene una característica particular, no se logró generar una sombra que contrastara este defecto, por lo cual este no pudo ser visto en las características de haralick, evitando su detección al ser analizado.

6.2. Principales aportaciones

A continuación se enumeran de manera resumida las principales aportaciones de esta tesis.

- Se presenta un modelo multiresolución combinado con estadística de segundo orden basado en las características de haralick para la detección de defectos desarrollado por LabVIEW[®], donde la caracterización se realiza en un ambiente interactivo y de viable navegación.
- Se ha realizado un estudio de los métodos de detección de defectos en texturas aleatorias así como métodos enfocados a pieles curtidas de bovino, esto resultó vital para este trabajo ya que proporcionó algunos de los fundamentos teóricos empleados, brindó una idea de los posibles aportes y contribuciones de esta investigación y, además, ayudó a plantear el trabajo a futuro.
- Se investigó y experimentaron diferentes tipos de iluminación, como se mencionó esta es de vital importancia para cualquier proyecto que involucre un sistema de visión. Con respecto a esto no se encontró investigación enfocada a texturas. Se implementó un sistema de iluminación de forma directa, esta ayudó a mejorar la detección de defectos en pieles curtidas de bovino, contrastando los defectos generando una sombra que nos ayudó posteriormente a su detección.
- La piel curtida de bovino fue dividida en porciones más pequeñas para su análisis, ayudando a tener más píxeles por cada milímetro y así poder detectar la mayoría de los defectos. Estas porciones fueron unidas para generar una imagen general donde se resaltan los defectos encontrados en ella.
- El desarrollo de este proyecto conllevó a numerosas etapas de investigación, ajustes y pruebas con la finalidad de alcanzar el algoritmo final, basado en métodos y en las respectivas consideraciones de las diferentes problemáticas suscitadas en el desarrollo.
- El sistema de detección de pieles curtidas de bovino funcionó acorde a las limitaciones del proyecto: se trabajó con pieles curtidas de bovino, se utilizó un sistema de iluminación que ayudó a resaltar los defectos que se encontraban en ella, todo esto aplicado en tiempo real.
- La elección de LabVIEW[®] como herramienta de programación fue acertada, ya que en ella se pueden encontrar muchos instrumentos virtuales, que con un nivel de conocimiento intermedio acerca del procesamiento de imágenes, permite al diseñador implementar de manera fácil y dinámica las soluciones.
- Se consiguió el objetivo de realizar todo el proceso de manera automática, realizando el análisis de la piel curtida de bovino de una forma automática.
- Se diseñó una interfaz, la cual facilita el uso del sistema de inspección, y su adaptación para otros tipos de pieles curtidas de bovino.
- Aunque el sistema este diseñado para la piel curtida de bovino especificada, la metodología presentada para la caracterización de defectos, puede ser considerada para la inspección de otros tipos de pieles, es decir, el sistema puede ser utilizado para otros tipos de pieles siguiendo los pasos que en este proyecto se marcan.

6.3. Líneas de trabajo a futuro

A pesar de los buenos resultados obtenidos con el algoritmo implementado, que han quedado demostrados en el capítulo 5, es posible realizar ciertas mejoras:

- El desarrollo tecnológico nos ofrece cada día modelos de cámaras más robustas (de mayor resolución), se recomienda en el futuro experimentar con una cámara de mejores características.
- La mejora más interesante a partir del trabajo expuesto quizá será la aplicación del proceso a imágenes de texturas a color. Los pasos a seguir serían los mismos salvo que habría de hacerlo por triplicado, una vez para cada plano de color. Así mismo, debe estudiarse que sistema de representación de color es el más adecuado: RGB (R-rojo, G-verde, B-azul), XYZ (Y-luminancia, X y Z - media del color), HSI (H-Tono, S-saturación, I-intensidad), HSV (H-Tono, S-saturación, V-brillo).
- Se recomienda el uso de LabVIEW[®] para el desarrollo de aplicaciones de visión, ya que permite usar y modificar de manera sencilla los diferentes algoritmos, filtros y análisis que forman parte del procesamiento digital.
- Aunque el procesamiento de cada imagen era muy rápido (1.5 s para cada imagen), para completar el proceso de análisis de todas las porciones se necesitaban 7 minutos, esto debido a los motores a paso que a pesar de proporcionar una fuerza de avance manteniendo una gran rigidez, genera una velocidad baja, por lo tanto se recomienda investigar otros tipos de motores para corregir este problema.
- Para que este sistema pueda funcionar en diferentes pieles curtidas se podría agregar un estado en el código, donde antes de inicial el análisis se pueda leer por medio de un código puesto en la piel y así identificar el tipo de piel que es, a partir de esto el sistema podrá swichear los algoritmos para su correcto funcionamiento.

Referencias

- [1] R. Haralick and L. Shapiro, *Computer and robot vision*, vol. 1. Addison-Wesley, 1992.
- [2] F. Pernkopf, "Detection of surface defects on raw steel blocks using bayesian network classifiers," *Pattern Analysis & Applications*, vol. 7, no. 3, pp. 333–342, 2004.
- [3] J. Liu and J. MacGregor, "Estimation and monitoring of product aesthetics: Application to manufacturing of engineered stone countertops," *Machine Vision and Applications*, vol. 16, no. 6, pp. 374–383, 2006.
- [4] V. Murino, M. Bicego, and I. Rossi, "Statistical classification of raw textile defects," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 4, pp. 311–314, IEEE, 2004.
- [5] L. Siew, R. Hodgson, and E. Wood, "Texture measures for carpet wear assessment," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 10, no. 1, pp. 92–105, 1988.
- [6] X. Xianghua and M. Mirmehdi, "Texems: Texture exemplars for defect detection on random textured surfaces," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 8, pp. 1454–1464, 2007.
- [7] M. Niskanen, O. Silvén, and H. Kauppinen, "Color and texture based wood inspection with non-supervised clustering," in *PROCEEDINGS OF THE SCANDINAVIAN CONFERENCE ON IMAGE ANALYSIS*, pp. 336–342, 2001.
- [8] M. Niskanen, H. Kauppinen, and O. Silvén, "Real-time aspects of som-based visual surface inspection," *Machine Vision Applications in Industrial Inspection*, vol. 4664, pp. 123–134, 2002.
- [9] J. Kittler, R. Marik, M. Mirmehdi, M. Petrou, and J. Song, "Detection of defects in colour texture surfaces," *IAPR Machine Vision Applications*, pp. 558–567, 1994.
- [10] D. Tsai and C. Hsieh, "Automated surface inspection for directional textures," *Image and Vision Computing*, vol. 18, no. 1, pp. 49–62, 1999.
- [11] X. Xie and M. Mirmehdi, "Localising surface defects in random colour textures using multiscale texem analysis in image eigenchannels," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 3, pp. III–1124, IEEE, 2005.
- [12] W. Amorim, H. Pistori, M. Pereira, and M. Jacinto, "Attributes reduction applied to leather defects classification," in *Graphics, Patterns and Images (SIBGRAPI), 2010 23rd SIBGRAPI Conference on*, pp. 353–359, IEEE, 2010.
- [13] J. Sobral, "Leather inspection based on wavelets," *Pattern Recognition and Image Analysis*, pp. 359–394, 2005.
- [14] P. Villar, M. Mora, and P. Gonzalez, "A new approach for wet blue leather defect segmentation," *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp. 591–598, 2011.
- [15] National Instruments, *NI Vision Concepts Help*, 3729161-01 ed., Junio 2011.
- [16] C. Yeh and D. Perng, "A reference standard of defect compensation for leather transactions," *The International Journal of Advanced Manufacturing Technology*, vol. 25, no. 11, pp. 1197–1204, 2005.

- [17] C. Kwak, J. Ventura, and K. Tofang-Sazi, "Automated defect inspection and classification of leather fabric," *Intelligent Data Analysis*, vol. 5, no. 4, p. 355, 2001.
- [18] R. Gonzalez and R. Woods, *Tratamiento digital de imagenes*. Addison-Wesley Longman, 1996.
- [19] A. Rosenfeld and A. Kak, "Digital picture processing. volumes 1 & 2 //(book)," *New York, Academic Press, 1982,*, 1982.
- [20] "Ieee standard glossary of image processing and pattern recognition terminology," *IEEE Std 610.4 1990*, 1990.
- [21] A. Jain, *Fundamentals of digital image processing*. Prentice-Hall, Inc., 1989.
- [22] A. Materka, M. Strzelecki, *et al.*, "Texture analysis methods-a review," *Technical University of Lodz, Institute of Electronics, COST B*, vol. 11, pp. 9–11, 1998.
- [23] C. Chen and C. Chen, "Filtering methods for texture discrimination," *Pattern Recognition Letters*, vol. 20, no. 8, pp. 783–790, 1999.
- [24] T. Randen and J. Husoy, "Filtering for texture classification: A comparative study," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 4, pp. 291–310, 1999.
- [25] R. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.
- [26] L. Van Gool, P. Dewaele, and A. Oosterlinck, "Texture analysis anno 1983," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 3, pp. 336–357, 1985.
- [27] F. Vilnrotter, R. Nevatia, and K. Price, "Structural analysis of natural textures," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 1, pp. 76–89, 1986.
- [28] S. Arivazhagan and L. Ganesan, "Texture classification using wavelet transform," *Pattern Recognition Letters*, vol. 24, no. 9, pp. 1513–1521, 2003.
- [29] J. Zhang and T. Tan, "Brief review of invariant texture analysis methods," *Pattern recognition*, vol. 35, no. 3, pp. 735–747, 2002.
- [30] P. Maillard, "Comparing texture analysis methods through classification," *Photogrammetric engineering and remote sensing*, vol. 69, no. 4, pp. 357–368, 2003.
- [31] I. Epifanio López, *Descripción de texturas. Aplicaciones a su comprensión y clasificación*. PhD thesis, Universitat de València, 2002.
- [32] J. Mao and A. Jain, "Texture classification and segmentation using multiresolution simultaneous autoregressive models," *Pattern recognition*, vol. 25, no. 2, pp. 173–188, 1992.
- [33] V. Nalwa, *A guided tour of computer vision*. 1994.
- [34] E. Trucco and A. Verri, *Introductory techniques for 3-D computer vision*, vol. 93. Prentice Hall Upper Saddle River eNJ NJ, 1998.
- [35] D. Forsyth and J. Ponce, *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002.
- [36] M. Groover, M. Weiss, and R. Nagel, *Industrial Robotics: Technology, Programming and Application*. McGraw-Hill Higher Education, 1986.
- [37] S. AGNES, "Manual de prácticas de visión por computador," *Barcelona: Infaimon*, 2008.
- [38] "Tipos de iluminacion," Noviembre 2005.
- [39] DCM, "Formas de iluminacion," 2012.
- [40] X. Xie, "A review of recent advances in surface defect detection using texture analysis techniques," *ELCVIA*, vol. 7, no. 3, 2008.

- [41] R. Chin, “Automated visual inspection: 1981 to 1987,” *Computer Vision, Graphics, and Image Processing*, vol. 41, no. 3, pp. 346–381, 1988.
- [42] T. Newman and A. Jain, “A survey of automated visual inspection,” *Computer vision and image understanding*, vol. 61, no. 2, pp. 231–262, 1995.
- [43] Y. Li and P. Gu, “Free-form surface inspection techniques state of the art review,” *Computer-Aided Design*, vol. 36, no. 13, pp. 1395–1417, 2004.
- [44] M. Pietikainen, T. Maenpaa, and J. Viertola, “Color texture classification with color histograms and local binary patterns,” in *Workshop on Texture Analysis in Machine Vision*, pp. 109–112, Citeseer, 2002.
- [45] R. Broadhurst, J. Stough, S. Pizer, and E. Chaney, “Histogram statistics of local image regions for object segmentation,” in *International Workshop on Deep Structure, Singularities, and Computer Vision*, Citeseer, 2005.
- [46] X. Xie, M. Mirmehdi, and B. Thomas, “Colour tonality inspection using eigenspace features,” *Machine Vision and Applications*, vol. 16, no. 6, pp. 364–373, 2006.
- [47] I. Tsai, C. Lin, J. Lin, *et al.*, “Applying an artificial neural network to pattern recognition in fabric defects,” *Textile Research Journal*, vol. 65, no. 3, pp. 123–130, 1995.
- [48] H. Ng, “Automatic thresholding for defect detection,” *Pattern Recognition Letters*, vol. 27, no. 14, pp. 1644–1649, 2006.
- [49] N. Otsu, “A threshold selection method from gray-level histograms,” *Automatica*, vol. 11, pp. 285–296, 1975.
- [50] J. Iivarinen, J. Rauhamaa, and A. Visa, “Unsupervised segmentation of surface defects,” in *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, vol. 4, pp. 356–360, IEEE, 1996.
- [51] A. Latif-Amet, A. Ertüzün, and A. Erçil, “An efficient method for texture defect detection: sub-band domain co-occurrence matrices,” *Image and Vision Computing*, vol. 18, no. 6-7, pp. 543–553, 2000.
- [52] Y. Huang and K. Chan, “Texture decomposition by harmonics extraction from higher order statistics,” *Image Processing, IEEE Transactions on*, vol. 13, no. 1, pp. 1–14, 2004.
- [53] A. Monadjemi, *Towards efficient texture classification and abnormality detection*. PhD thesis, University of Bristol, 2004.
- [54] T. Mäenpää, M. Turtinen, and M. Pietikäinen, “Real-time surface inspection by texture,” *Real-Time Imaging*, vol. 9, no. 5, pp. 289–296, 2003.
- [55] J. Chen and A. Jain, “A structural approach to identify defects in textured images,” in *Systems, Man, and Cybernetics, 1988. Proceedings of the 1988 IEEE International Conference on*, vol. 1, pp. 29–32, IEEE, 1988.
- [56] W. Wen and A. Xia, “Verifying edges for visual inspection purposes,” *Pattern recognition letters*, vol. 20, no. 3, pp. 315–328, 1999.
- [57] C. Chen, *Handbook of pattern recognition and computer vision*. World Scientific, 2010.
- [58] C. Neubauer, “Segmentation of defects in textile fabric,” in *Pattern Recognition, 1992. Vol. I. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on*, pp. 688–691, IEEE, 1992.
- [59] D. Tsai and T. Huang, “Automated surface inspection for statistical textures,” *Image and Vision Computing*, vol. 21, no. 4, pp. 307–323, 2003.
- [60] C. Chan and G. Pang, “Fabric defect detection by fourier analysis,” *Industry Applications, IEEE Transactions on*, vol. 36, no. 5, pp. 1267–1276, 2000.

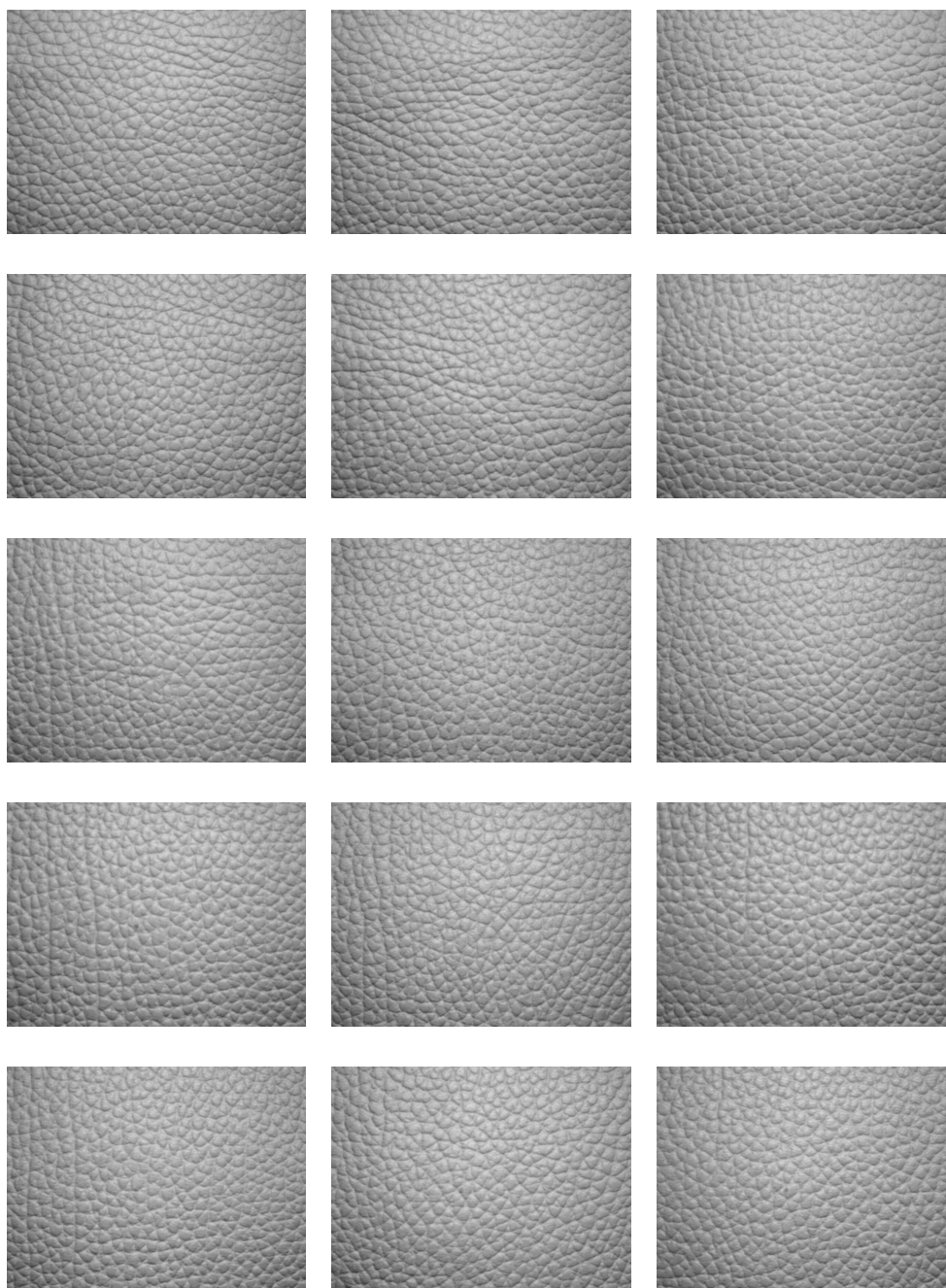
- [61] H. Sari-Sarraf and J. Goddard Jr, "Vision system for on-loom fabric inspection," *Industry Applications, IEEE Transactions on*, vol. 35, no. 6, pp. 1252–1259, 1999.
- [62] J. Scharcanski, "Stochastic texture analysis for monitoring stochastic processes in industry," *Pattern recognition letters*, vol. 26, no. 11, pp. 1701–1709, 2005.
- [63] S. Kim, M. Lee, and K. Woo, "Wavelet analysis to fabric defects detection in weaving processes," in *Industrial Electronics, 1999. ISIE'99. Proceedings of the IEEE International Symposium on*, vol. 3, pp. 1406–1409, IEEE, 1999.
- [64] F. Truchetet and O. Laligant, "Industrial applications of the wavelet and multi-resolution-based signal/image processing: a review," in *Proceedings of SPIE*, vol. 6356, p. 63560H, 2007.
- [65] B. Mandelbrot, *The fractal geometry of nature*. Wh Freeman, 1983.
- [66] F. Cohen, Z. Fan, and S. Attali, "Automated inspection of textile fabrics using textural models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 13, no. 8, pp. 803–808, 1991.
- [67] X. Xie and M. Mirmehdi, "Texture exemplars for defect detection on random textures," *Pattern Recognition and Image Analysis*, pp. 404–413, 2005.
- [68] A. Lerch and D. Chetverikov, "Knowledge-based line-correction rules in a machine-vision system for the leather industry," *Engineering Applications of Artificial Intelligence*, vol. 4, no. 6, pp. 433–438, 1991.
- [69] L. Jian, H. Wei, and H. Bin, "Research on inspection and classification of leather surface defects based on neural network and decision tree," in *Computer Design and Applications (ICCD), 2010 International Conference on*, vol. 2, pp. V2–381, IEEE, 2010.
- [70] R. Viana, R. Rodrigues, M. Alvarez, and H. Pistori, "Svm with stochastic parameter selection for bovine leather defect classification," *Advances in Image and Video Technology*, pp. 600–612, 2007.
- [71] J. Sobral, "Optimised filters for texture defect detection," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 3, pp. III–565, IEEE, 2005.
- [72] A. Kumar and G. Pang, "Defect detection in textured materials using gabor filters," *Industry Applications, IEEE Transactions on*, vol. 38, no. 2, pp. 425–440, 2002.
- [73] K. Krastev, L. Georgieva, and N. Angelov, "Leather features selection for defects recognition using fuzzy logic," *Energy*, vol. 2, p. 3, 2004.
- [74] F. He, W. Wang, and Z. Chen, "Automatic defects detection based on adaptive wavelet packets for leather manufacture," in *Technology and Innovation Conference, 2006. ITIC 2006. International*, pp. 2024–2027, IET, 2006.
- [75] D. Pérez Martínez, "Modelos psicofísicos de discriminación de texturas visuales: evolución y aspectos críticos," *Anuario de psicología*, no. 64, pp. 3–20, 1995.
- [76] J. Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles," *Vision research*, vol. 20, no. 10, pp. 847–856, 1980.
- [77] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 11, no. 7, pp. 674–693, 1989.
- [78] A. Laine and J. Fan, "Frame representations for texture segmentation," *Image Processing, IEEE Transactions on*, vol. 5, no. 5, pp. 771–780, 1996.
- [79] R. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 3, no. 6, pp. 610–621, 1973.
- [80] M. Unser, "Texture classification and segmentation using wavelet frames," *Image Processing, IEEE Transactions on*, vol. 4, no. 11, pp. 1549–1560, 1995.

- [81] B. Julész, E. Gilbert, L. Shepp, H. Frisch, *et al.*, “Inability of humans to discriminate between visual textures that agree in second-order statistics-revisited,” *Perception*, vol. 2, no. 4, pp. 391–405, 1973.
- [82] V. Hodge and J. Austin, “A survey of outlier detection methodologies,” *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, 2004.
- [83] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, “Support vector method for novelty detection,” *Advances in neural information processing systems*, vol. 12, no. 3, pp. 582–588, 2000.
- [84] V. Vapnik, “Statistical learning theory. 1998,” 1998.
- [85] A. Aizerman, E. Braverman, and L. Rozoner, “Theoretical foundations of the potential function method in pattern recognition learning,” *Automation and remote control*, vol. 25, pp. 821–837, 1964.

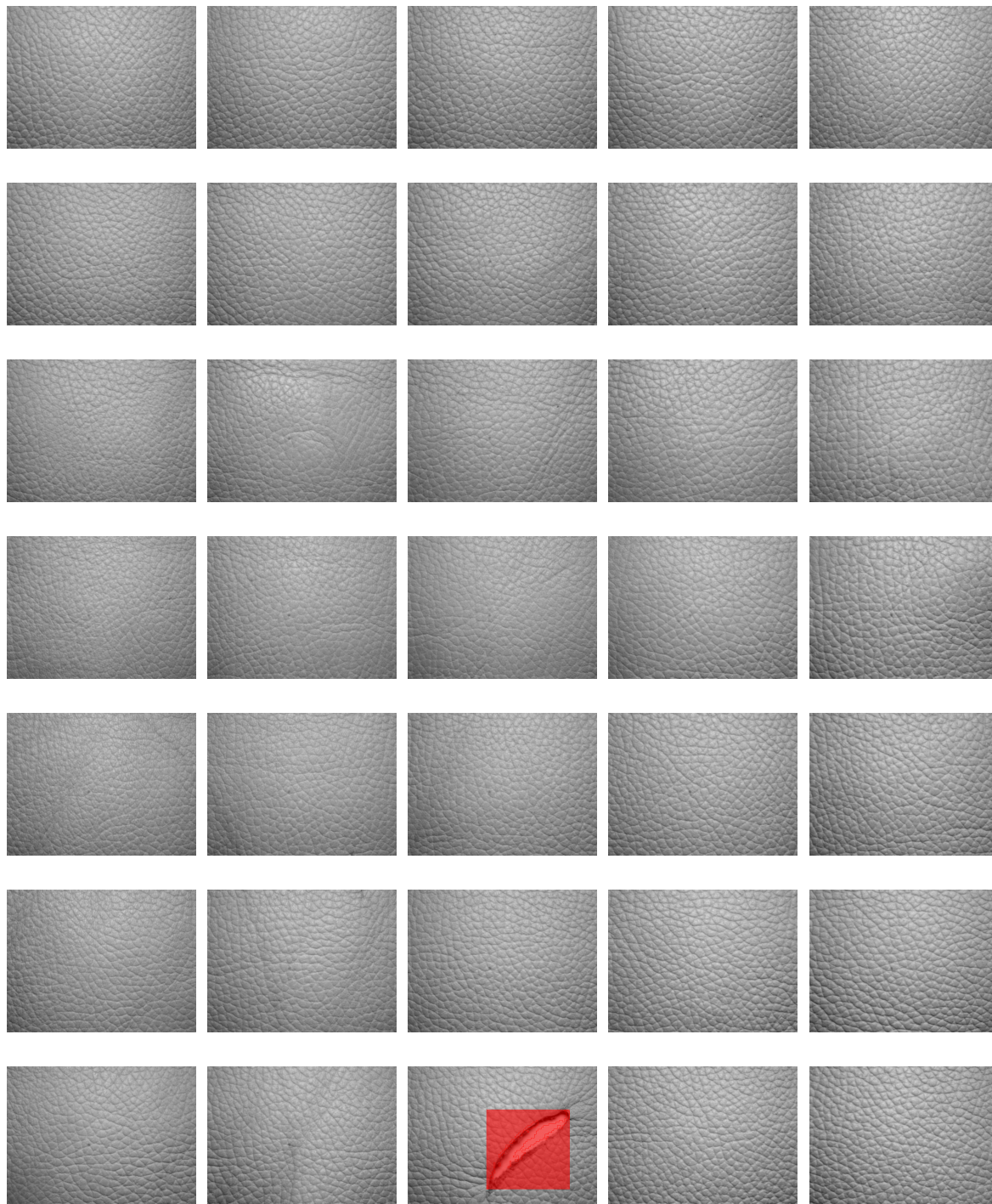
Apéndice A

Pieles curtidas analizadas

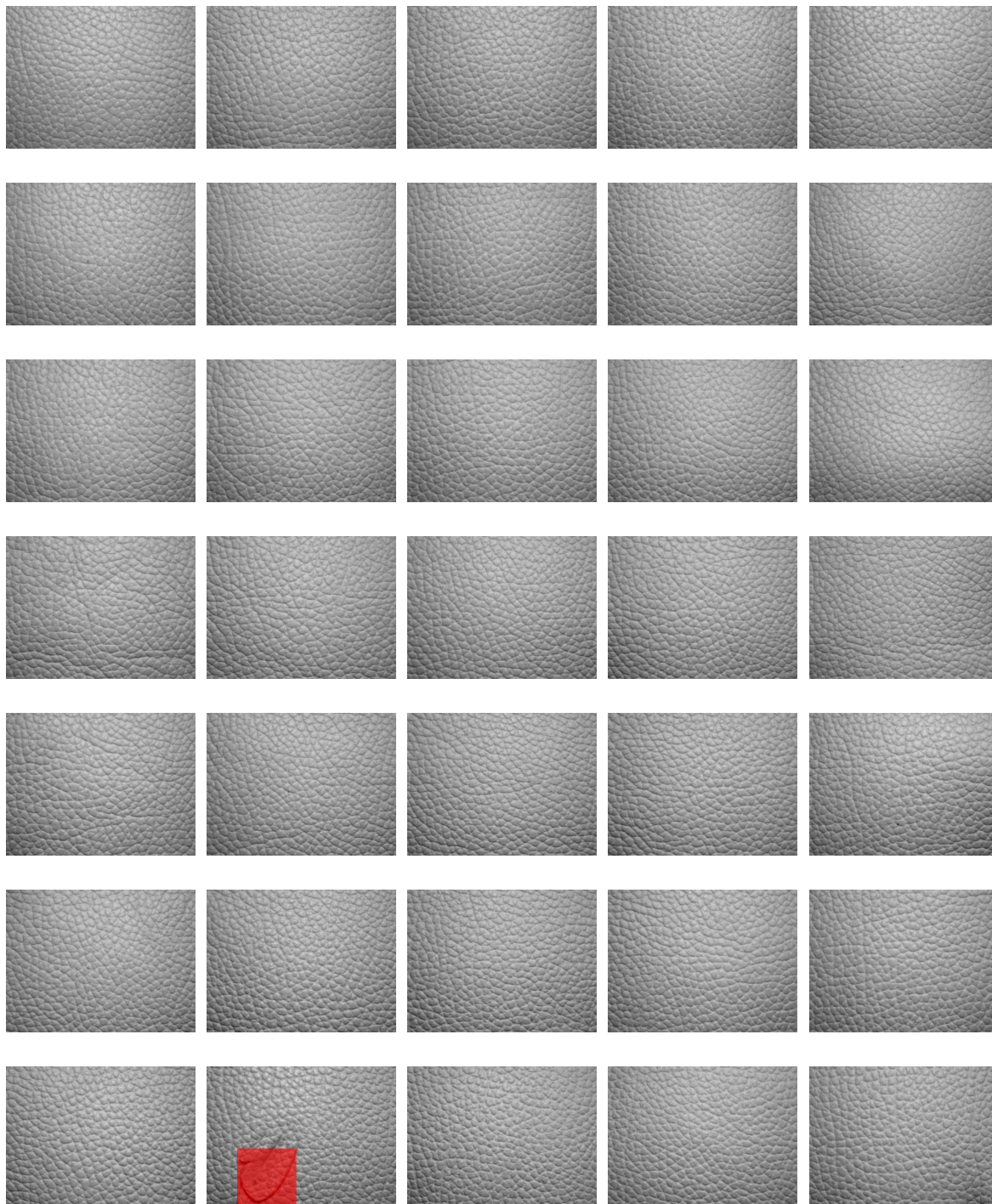
A.1. Texturas muestras en buen estado



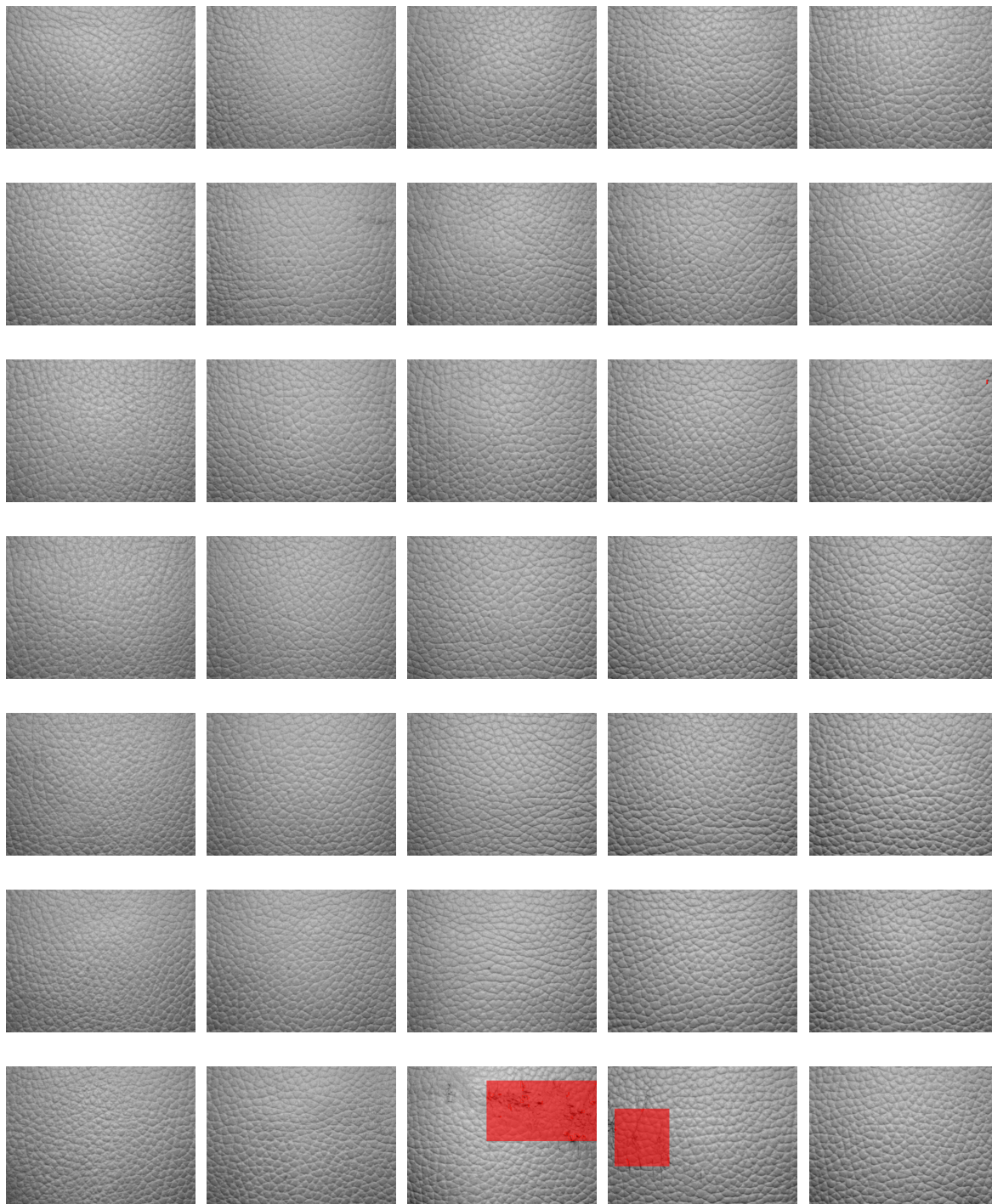
A.2. Piel No 1



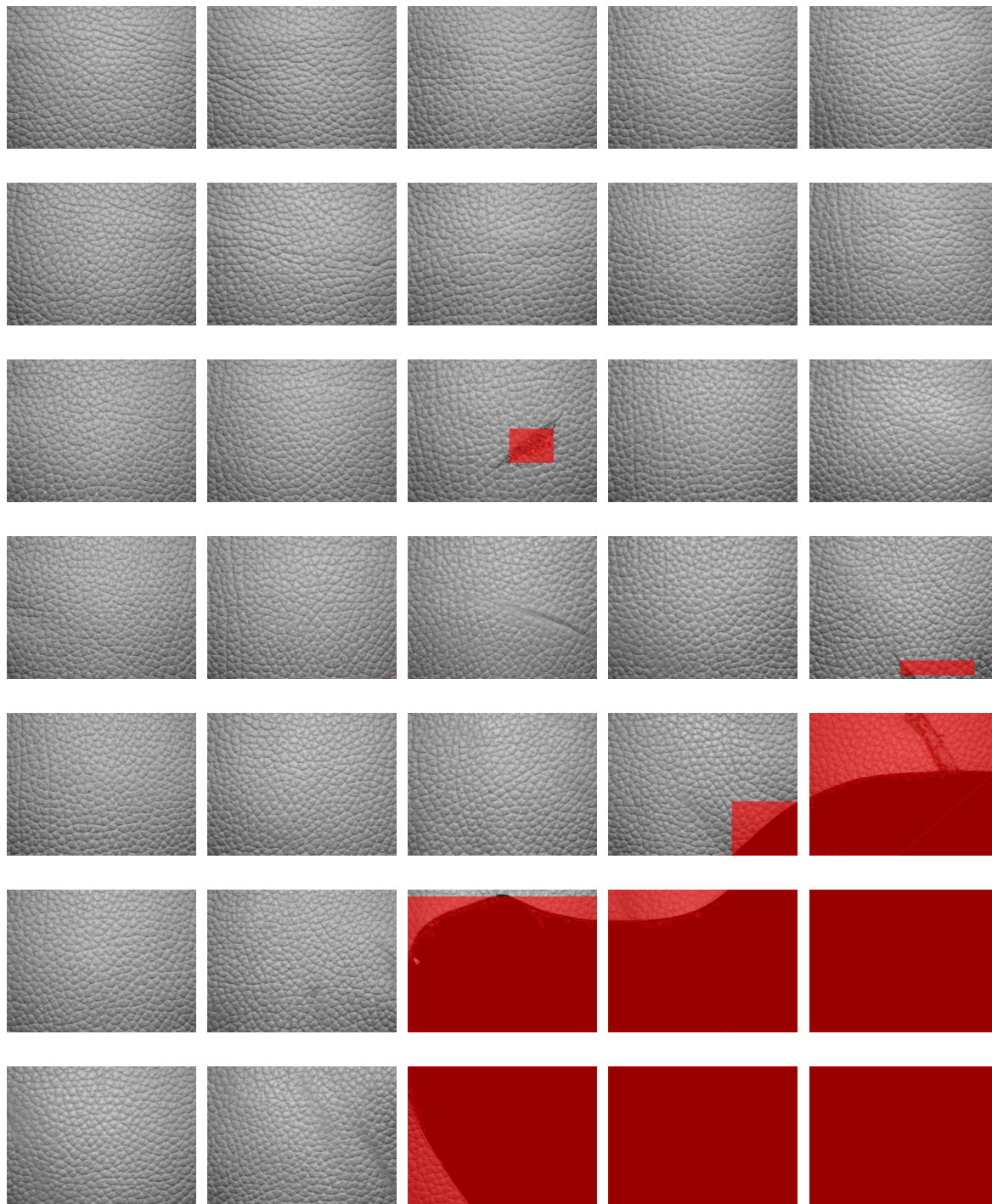
A.3. Piel No 2



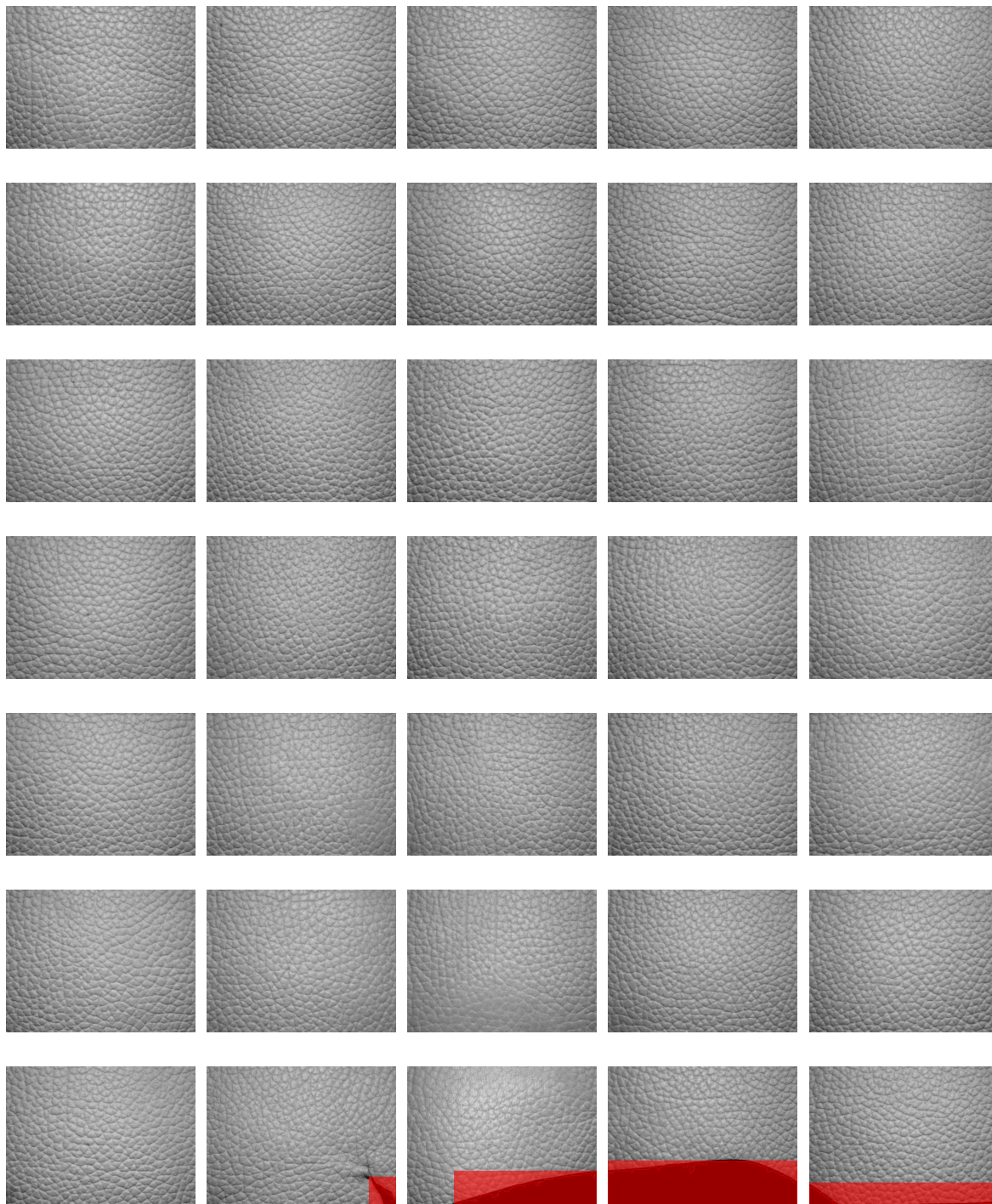
A.4. Piel No 3



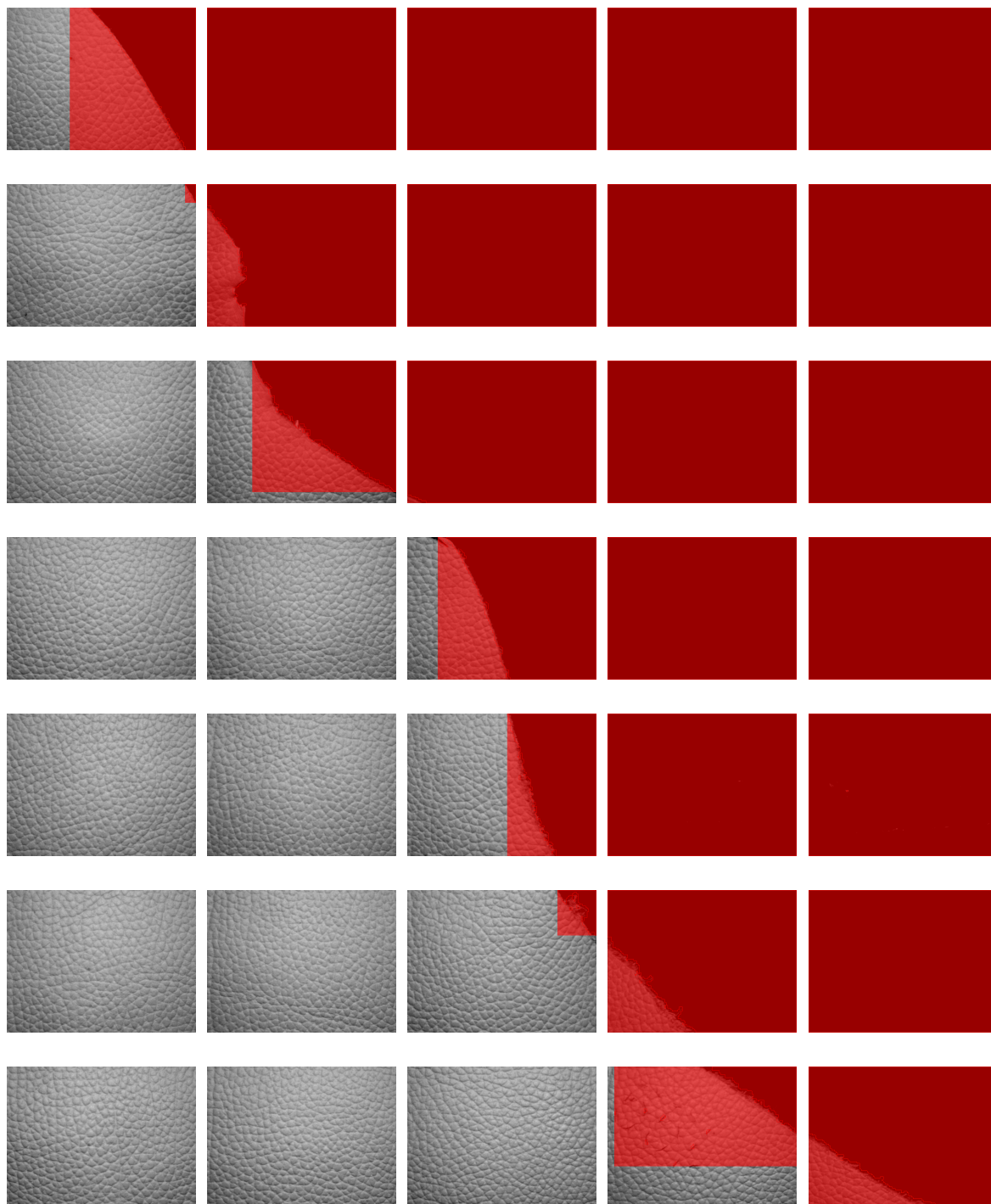
A.5. Piel No 4



A.6. Piel No 5



A.7. Piel No 6



Apéndice B

Hoja de datos

1. Puertos de conexión del bloque conector CB-68LP
2. Hoja de especificaciones cámara JAI CV-M50
3. Hoja de especificaciones motor minebea 23KM-K
4. Datasheets sensor H21A1

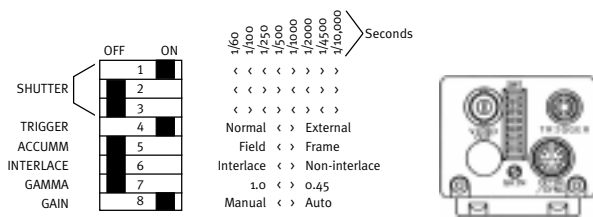
ACH8	34	68	ACH0
ACH1	33	67	AIGND
AIGND	32	66	ACH9
ACH10	31	65	ACH2
ACH3	30	64	AIGND
AIGND	29	63	ACH11
ACH4	28	62	AISENSE
AIGND	27	61	ACH12
ACH13	26	60	ACH5
ACH6	25	59	AIGND
AIGND	24	58	ACH14
ACH15	23	57	ACH7
DAC0OUT1	22	56	AIGND
DAC1OUT1	21	55	AOGND
RESERVED	20	54	AOGND
DIO4	19	53	DGND
DGND	18	52	DIO0
DIO1	17	51	DIO5
DIO6	16	50	DGND
DGND	15	49	DIO2
+5 V	14	48	DIO7
DGND	13	47	DIO3
DGND	12	46	SCANCLK
PF10/TRIG1	11	45	EXTSTROBE*
PF11/TRIG2	10	44	DGND
DGND	9	43	PF12/CONVERT*
+5 V	8	42	PF13/GPCTR1_SOURCE
DGND	7	41	PF14/GPCTR1_GATE
PF15/UPDATE*	6	40	GPCTR1_OUT
PF16/WFTRIG	5	39	DGND
DGND	4	38	PF17/STARTSCAN
PF18/GPCTR0_GATE	3	37	PF18/GPCTR0_SOURCE
GPCTR0_OUT	2	36	DGND
FREQ_OUT	1	35	DGND

Specifications for CV-M50

Specifications	CV-M50C	CV-M50E
Scanning system	625 lines 25 frames/sec.	525 lines 30 frames/sec.
CCD sensor	Monochrome 1/2" Hyper HAD IT CCD	
Sensing area	6.6 mm (h) x 4.8 mm (v)	
Picture elements effective	752 (h) x 582 (v)	768 (h) x 494 (v)
Elements in video out	737 (h) x 575 (v)	758 (h) x 486 (v)
Cell size	8.6 (h) x 8.3 (v) μm	8.4 (h) x 9.8 (v) μm
Resolution (horizontal)	560 TV lines	570 TV lines
Resolution (vertical)	575 TV lines	485 TV lines
Sensitivity	0.5 Lux, F1.4	
S/N ratio	>56 dB (AGC off, Gamma 1)	
Video output	Composite VBS signal 1.0 Vpp, 75 Ohm	
Gamma	0.45 - 1.0	
Gain	Manual - Automatic Potmeter - AGC	
Scanning	2:1 interlace - non-interlace	
Accumulation	Field - frame	
Synchronization	Int. X-tal. Ext HD/VD or random trigger	
HD sync. input/output	4V, 75 Ohm*	
VD sync. input/output	4V, 75 Ohm*	
Trigger input	4V, 75 Ohm*	
Trigger input duration	>HD interval	
WEN output (write enable)	4V, 75 Ohm	
EEN output (exposure enable)	4V, 75 Ohm	
Pixel clock out (optional)	4V, 75 Ohm sine	
Internal shutter	Off, 1/100, 1/250, 1/500, 1/1000, 1/2000, 1/4500, 1/10,000 sec.	
Trigger shutter	1/60, 1/100, 1/250, 1/500, 1/1000, 1/2000, 1/4500, 1/10,000 sec.	
Long time exposure	1 field to ∞ Duration between ext. VD pulses	
Operating temperature	-5°C to +45°C	
Humidity	20 - 80% non-condensing	
Storage temp./humidity	-25°C to 60°C/20 - 90%	
Power	12V DC \pm 10%. 2.5W	
Lens mount	C-mount	
Dimensions	40 x 50 x 80 mm (HxWxD)	
Weight	245 g	

* HD sync., VD sync. (Trigger) input or output by internal jumpers.
Factory setting: HD/VD input. Inputs TTL or 75 Ohm terminated by internal jumpers. Factory setting: 75 Ohm terminated.

Switch Setting

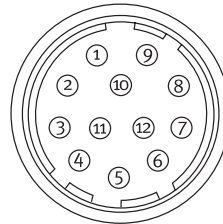


Ordering Information

CV-M50C 1/2" Monochrome Camera CCIR.
CV-M50E 1/2" Monochrome Camera EIA.

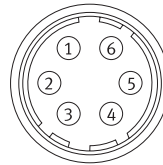
Connection Description

DC-IN/SYNC.



- Pin 1 Ground
- 2 +12V DC
- 3 Ground
- 4 Video output
- 5 Ground
- 6 HD in/(HD out)
- 7 VD in/(VD out)
- 8 Ground
- 9 Pixel clock out (sine) optional**
- 10 Ground
- 11 +12V DC
- 12 Ground

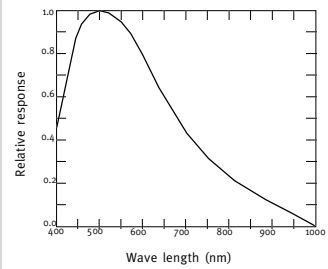
TRIGGER



- Pin 1 N.C.
- 2 EEN out
- 3 Ground
- 4 N.C.
- 5 Trig. in/VD out
- 6 WEN pulse out

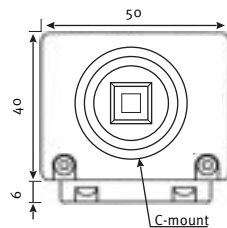
** Pixel clock output by internal jumper setting.

Spectral Sensitivity

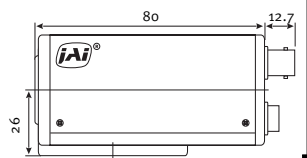


Dimensions

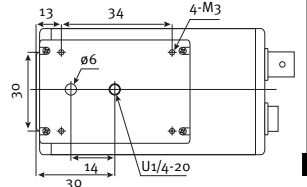
Front view



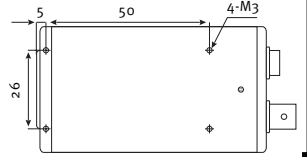
Side view



Bottom view



Top view



JAI Corporation, Japan
German Industry Center
1-18-2 Hakusan, Midori-ku
Yokohama,
Kanagawa 226-0006, Japan
Phone +81 45 933 5400
Fax +81 45 931 6142
www.jai-corp.co.jp

JAI A-S, Denmark
Camera Solutions
Produktionsvej 1, 2600 Glostrup
Copenhagen, Denmark
Phone +45 4491 8888
Fax +45 4491 8880
www.jai.dk

JAI UK Ltd., England
Audley House
Northbridge Road
Berkhamsted
Herts HP4 1EH, England
Phone +44 1442 879 669
Fax +44 1442 879 281

Costar, USA
43517 Ridge Park Drive
Temecula, CA 92590
United States of America
Phone +1 909 699 9000
Fax +1 909 308 9188
www.costar-usa.com



THE MECHADEMIC COMPANY

Visit our web site on www.jai.dk

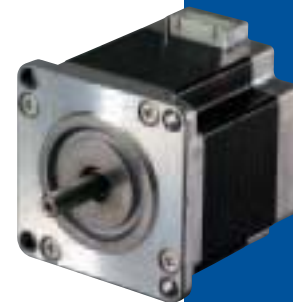
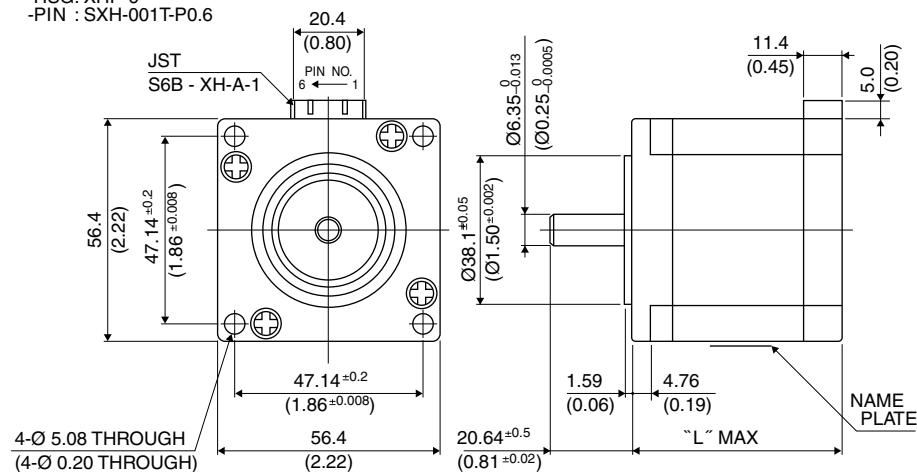
56 □ 1.8°

23KM-K

1.8°

■ 外観図 Outline

- LEAD WIRE SIDE
- HSG: XHP-6
- PIN : SXH-001T-P0.6



Hybrid

*シャフト径 $\text{Ø}8_{-0.013}$ mmも対応可能です。 *Shaft diameter of $\text{Ø}8_{-0.013}$ mm is also available.

UNIT: mm (inch)

結線表 PIN NO. vs. PHASE

相 (PHASE)	A	A COM	\bar{A}	B	B COM	\bar{B}
ピン番号 (PIN NO.)	1	2	3	4	5	6

	"L"
23KM-K2**	42 (1.65)
23KM-K3**	50 (1.97)
23KM-K0**	54 (2.13)
23KM-K1**	67 (2.64)
23KM-K7**	76 (2.99)

■ 仕様 Specifications

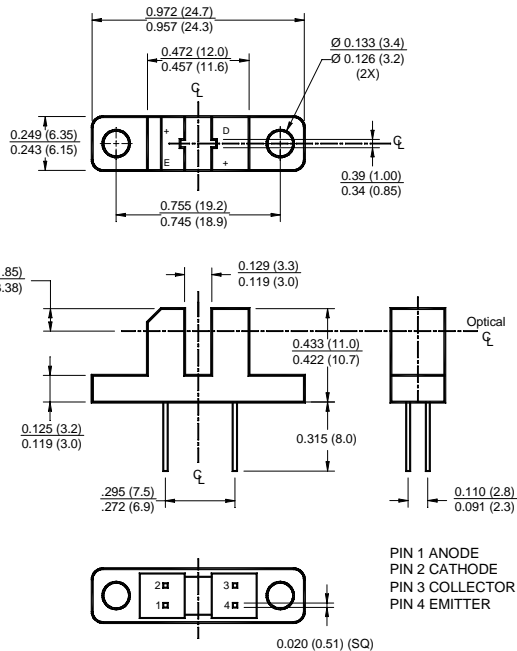
型式	ステップ角	ドライブ方式	定格電流	巻線抵抗	ホールディングトルク	インダクタンス	ロータイナーシャ	デイトントルク	質量
Model	Step Angle	Drive Sequence	Rated Current	Resistance	Holding Torque	Inductance	Rotor Inertia	Detent Torque	Mass
	(deg)		(A)	(Ohms)	(mNm)	(mH)	(g·cm ²)	(mNm)	(g)
23KM-K044U	1.8	UNI-POLAR	3	0.85	760	1.8	200	25	680
23KM-K055U	1.8	UNI-POLAR	1.5	3.3	760	6.7	200	25	680
23KM-K144U	1.8	UNI-POLAR	3	1.0	1050	2.2	290	49	900
23KM-K155U	1.8	UNI-POLAR	1.5	3.9	1050	8.0	290	49	900
23KM-K249U	1.8	UNI-POLAR	2	1.2	420	1.9	120	20	470
23KM-K255U	1.8	UNI-POLAR	1.5	2.3	420	3.6	120	20	470
23KM-K349U	1.8	UNI-POLAR	2	1.5	650	2.9	180	29	590
23KM-K355U	1.8	UNI-POLAR	1.5	2.9	650	5.5	180	29	590
23KM-K744U	1.8	UNI-POLAR	3	1.15	1200	2.7	360	51	1050
23KM-K755U	1.8	UNI-POLAR	1.5	4.6	1200	10.2	360	51	1050
23KM-K044B	1.8	BI-POLAR	2.2	1.7	1000	7.2	200	25	680
23KM-K144B	1.8	BI-POLAR	2.1	2.0	1250	8.7	290	49	900
23KM-K249B	1.8	BI-POLAR	1.4	2.4	540	7.7	120	20	470
23KM-K349B	1.8	BI-POLAR	1.4	3.0	800	11.8	180	29	590
23KM-K744B	1.8	BI-POLAR	2	2.3	1500	10.6	360	51	1050

H21A1 / H21A2 / H21A3

PHOTOTRANSISTOR

OPTICAL INTERRUPTER SWITCH

PACKAGE DIMENSIONS

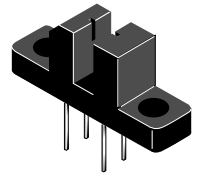


NOTES:

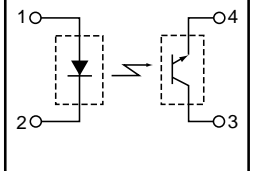
- Dimensions for all drawings are in inches (mm).
- Tolerance of $\pm .010 (.25)$ on all non-nominal dimensions unless otherwise specified.

DESCRIPTION

The H21A1, H21A2 and H21A3 consist of a gallium arsenide infrared emitting diode coupled with a silicon phototransistor in a plastic housing. The packaging system is designed to optimize the mechanical resolution, coupling efficiency, ambient light rejection, cost and reliability. The gap in the housing provides a means of interrupting the signal with an opaque material, switching the output from an "ON" to an "OFF" state.



SCHEMATIC



FEATURES

- Opaque housing
- Low cost
- .035" apertures
- High $I_{C(ON)}$

1. Derate power dissipation linearly 1.33 mW/°C above 25°C.
2. RMA flux is recommended.
3. Methanol or isopropyl alcohols are recommended as cleaning agents.
4. Soldering iron tip 1/16" (1.6mm) minimum from housing.

ABSOLUTE MAXIMUM RATINGS ($T_A = 25^\circ\text{C}$ unless otherwise specified)

Parameter	Symbol	Rating	Unit
Operating Temperature	T_{OPR}	-55 to +100	°C
Storage Temperature	T_{STG}	-55 to +100	°C
Soldering Temperature (Iron) ^(2,3 and 4)	T_{SOL-I}	240 for 5 sec	°C
Soldering Temperature (Flow) ^(2 and 3)	T_{SOL-F}	260 for 10 sec	°C
INPUT (EMITTER)			
Continuous Forward Current	I_F	50	mA
Reverse Voltage	V_R	6	V
Power Dissipation ⁽¹⁾	P_D	100	mW
OUTPUT (SENSOR)			
Collector to Emitter Voltage	V_{CEO}	30	V
Emitter to Collector Voltage	V_{ECO}	4.5	V
Collector Current	I_C	20	mA
Power Dissipation ($T_C = 25^\circ\text{C}$) ⁽¹⁾	P_D	150	mW

Apéndice C

Caracterización del motor

Para calcular cuantos pasos tiene que dar los motores, se diseñó una plantilla en tamaño real de la separación del espacio de trabajo. Como se puede apreciar en la Figura (C.1) en el centro de cada cuadro se tiene una circunferencia que indica el centro de cada espacio, esta cuadrícula fue fijada en el espacio de trabajo empezando desde el home del manipulador.

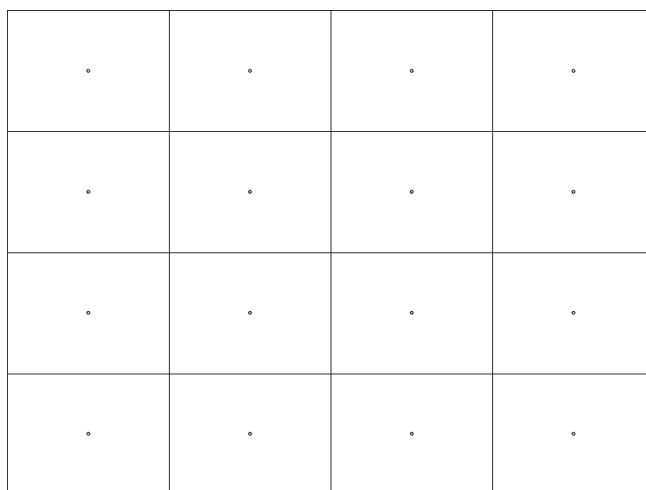


Figura C.1: Fragmento de la cuadrícula diseñada para el cálculo del número de pasos.

Para realizar esta experimentación se diseñaron dos programas, uno en LabVIEW[®] y otro en arduino.

- El programa en LabVIEW[®] genera una máscara en forma de punto en el centro del video tomado en tiempo real. Este sirve como referencia para saber si al terminar de dar los pasos, el manipulador ha llegado a la siguiente posición. Esto se verifica comparando los 2 puntos (En la cuadrícula y en la pantalla).
- El programa en arduino se generó para mover el motor un determinado número de pasos.

Durante la experimentación se corrió estos dos programas al mismo tiempo empezando desde el home del manipulador. Se variaron los ciclos de pasos hasta encontrar los valores correctos. Este experimento se repitió, para el motor 1 y motor 2 obteniendo los siguientes resultados.

Motor 1	Motor 2
1450 secuencias	925 secuencias

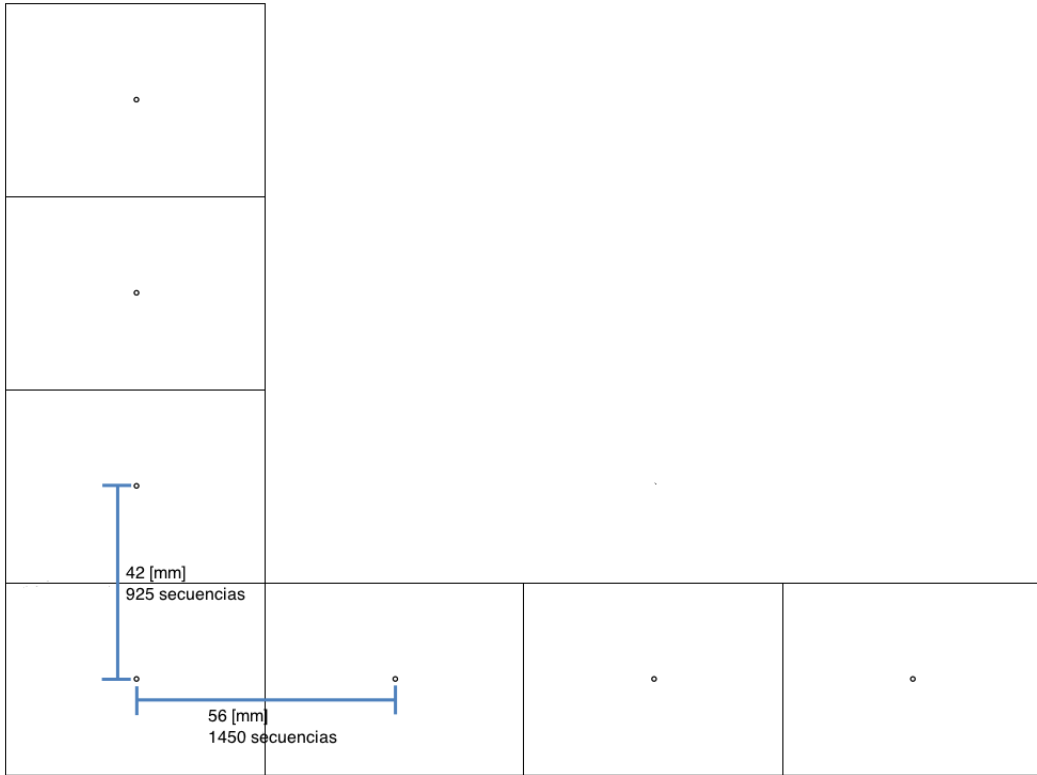


Figura C.2: Definición del número de secuencias para el avance de los motores.

Apéndice D

Cálculo de repetibilidad

Para realizar las mediciones, se imprimió cada una de las imágenes en una escala 4:1 y la distancia entre los puntos fue medida. Los datos obtenidos para cada posición fueron los siguientes:

Posición 1	Δx	Δy
1	0.25 [mm]	0.25 [mm]
2	0.25 [mm]	0.25 [mm]
3	0	0.50 [mm]
4	0	0.25 [mm]
Valor medio	0.125 [mm]	0.3125 [mm]

Posición 2	Δx	Δy
1	0.25 [mm]	0.50 [mm]
2	0.25 [mm]	0.50 [mm]
3	0	0.50 [mm]
4	0	0.50 [mm]
Valor medio	0.125 [mm]	0.50 [mm]

Posición 3	Δx	Δy
1	0.25 [mm]	0.50 [mm]
2	0.25 [mm]	0.25 [mm]
3	0.25 [mm]	0.50 [mm]
4	0.25 [mm]	0 [mm]
Valor medio	0.25 [mm]	0.3125 [mm]

Posición 4	Δx	Δy
1	0.25 [mm]	0.25 [mm]
2	0	0.25 [mm]
3	0.5 [mm]	0.25 [mm]
4	0.5 [mm]	0.25 [mm]
Valor medio	0.3125	0.25

Para especificar la repetibilidad del manipulador se explicará brevemente este concepto.

Repetibilidad: Radio de la esfera que abarca los puntos alcanzados por el robot tras suficientes movimientos, al ordenarle ir al mismo punto de destino programado. Analizando cada una de las tablas podemos observar que la mayor distancia del centro de la imagen al centro del punto de la cuadrícula diseñada es de 0.5 [mm] por lo tanto el radio que abarca todos los puntos alcanzados será de 0.5 [mm].

Precisión: Distancia entre el punto programado y el valor medio de los puntos realmente alcanzados al repetir el movimiento varias veces con carga. Para esto se tomará el valor medio mayor de todas las pruebas, esto es: $\Delta x = 0.3125$ $\Delta y = 0.3125$.

Repetibilidad [mm]	Precisión [mm]
0.5	$\Delta x = 0.3125$ $\Delta y = 0.3125$