

**Universidad Católica de Santa María**  
**Facultad de Ciencias e Ingenierías Físicas y**  
**Formales**  
**Programa Profesional de Ingeniería de Sistemas**



**Tema:**  
**“Mecanismo de autenticación de personas**  
**basado en la geometría de la mano”**

Tesis presentada por la Bachiller  
RODRIGUEZ PAREDES, SILVIA MARIA  
Para optar el Título Profesional de:  
INGENIERO DE SISTEMAS

**Arequipa**  
**-2013-**


## PRESENTACION

Sr. Director del Programa Profesional de Ingeniería de Sistemas.


Sres. Miembros del Jurado.

De conformidad con las disposiciones del Reglamento de Grados y Títulos del Programa Profesional de Ingeniería de Sistemas, pongo a vuestra consideración el presente trabajo de investigación titulado:

“Mecanismo de autenticación de personas basado en la geometría de la mano”, el mismo que de ser aprobado me permitirá optar el Título Profesional de Ingeniería de Sistemas.



Rodríguez Paredes, Silvia María



*“He tenido sueños y he tenido pesadillas. Superé mis pesadillas gracias a mis sueños.”*

**Jonas Salk**



*“A mis Padres Juan y Victoria, a mis hermanas Ángela y María Alejandra, que siempre me apoyan y me acompañan en todo este proceso y a mis familiares y amigos que siempre estuvieron para apoyarme”*

Mi más sincero agradecimiento a:

A mi asesor Dr. Guillermo Calderón Ruiz, por su acompañamiento y sus consejos en el desarrollo de mi tesis y por depositar su confianza en mi trabajo aquí expuesto.

Al Ing. Hanibal Sardon Paniagua por su apoyo y guía en mi tesis y por darme aliento en cada momento.

A mi madrina Antuca Nuñez y con especial cariño a José y a mis amigos Ginger, Natalie, Rosa, Steven, Fonseca, Jean, Jaime y Gonzalo por incentivar me siempre a seguir adelante y apoyarme en mi proceso de titulación.

## TABLA DE CONTENIDO

RESUMEN .....	15
ABSTRACT .....	16
INTRODUCCION .....	17
CAPITULO I: FUNDAMENTOS TEORICOS .....	19
1.1    MARCO CONCEPTUAL: .....	19
1.1.1    Identificación Vs. Autenticación .....	19
1.1.2    Biometría .....	19
1.1.3    Información Biométrica .....	19
1.1.4    Sistema Biométrico. ....	20
1.1.5    Paralelismo en la CPU .....	21
1.1.6    Multiprocesamiento .....	21
1.1.7    Intel Threading Building Blocks .....	23
1.1.8    Procesamiento digital de imágenes. ....	23
1.1.9    Métodos de digitalización de imágenes .....	33
1.2    ESTADO DEL ARTE: .....	40
CAPITULO II: IMPLEMENTACION DEL MECANISMO .....	44
2.1.    MATERIALES. ....	44
2.1.1    Captura de la imagen .....	44
2.1.2    Procesamiento de la imagen .....	46
2.2    OBTENCION DE LA IMAGEN Y SUS PARAMETROS .....	46
2.2.1    Almacenamiento. ....	46
2.2.2    Transformación. ....	46
2.3    PREPARACIÓN DE LA IMAGEN .....	47
2.4    OBTENCION DE CARACTERÍSTICAS .....	54
2.5    HERRAMIENTA DE PRUEBA. ....	89
2.5.1    Mecanismo Grafico. ....	89
2.5.2    Batch en Consola .....	91
2.5.3    Casos de Uso. ....	92

2.5.4	Pantallas del sistema.....	93
2.5.5	Distribución de indicadores.....	94
2.6	REQUISITOS DE DESARROLLO .....	98
2.6.1	Confiabilidad. ....	98
2.6.2	Validez en Contenido. ....	98
2.6.3	Validez en Criterio. ....	98
2.6.4	Validez de Constructo. ....	98
2.6.5	Objetiva. ....	99
CAPITULO III EVALUACION Y DISCUSION DE LOS RESULTADOS.....		100
3.1	Clasificación de las palmas. ....	100
3.2	Resultados de la variable de Grado de iluminación Vs. Espectro utilizado. ....	101
3.3	Segmentación de la imagen .....	102
3.4	Combinación de las variables Espectro-Iluminación–Segmentación.....	102
3.4.1	Error Tipo 1.....	103
3.4.2	Error Tipo 2.....	103
3.5	Tiempo de ejecución. ....	112
3.5.1	Equipos Utilizados.....	112
3.1.1	Resultados en tiempo de ejecución.....	113
CONCLUSIONES .....		115
RECOMENDACIONES .....		116
TRABAJOS FUTUROS.....		117
REFERENCIAS BIBLIOGRAFICAS .....		118
ANEXO A.....		120
PLANTEAMIENTO DE LA INVESTIGACION.....		120
1.1	Planteamiento del Problema.....	120
a.	Caracterización del problema. ....	120
1.2	Objetivos de la Investigación.....	120
a.	General.....	120
b.	Específicos. ....	121

1.3	Preguntas de Investigación. ....	121
1.4	Área Científica a la que corresponde el Problema. ....	121
1.5	Solución propuesta. ....	122
FUNDAMENTOS TEÓRICOS .....		126
2.1	Estado del arte. ....	126
2.2	Bases teóricas de la investigación. ....	129
METODOLOGÍAS A EMPLEAR .....		129
3.1	Tipo y nivel de la investigación. ....	129
a.	Tipo de Investigación. ....	129
b.	Nivel de la investigación. ....	130
3.2	Diseño de la investigación. ....	131
a.	<i>Formulación de la hipótesis:</i> .....	131
b.	<i>Diseño de las variables:</i> .....	131
c.	<i>Tipologías del diseño de la investigación.</i> .....	134
d.	<i>Contexto de la investigación.</i> .....	134
3.3	Universo o Población y Muestra. ....	134
a.	<i>Unidad de análisis.</i> .....	134
b.	Población. ....	134
c.	Tipo de muestra. ....	137
d.	Criterios de inclusión. ....	137
3.4	Técnicas e Instrumentos. ....	137
3.5	Procedimientos de recolección de datos. ....	138
3.6	Plan de análisis de datos .....	138
PLAN DE TRABAJO. ....		140
REFERENCIAS BIBLIOGRÁFICAS. ....		141
POSIBLE TEMARIO DEL INFORME FINAL .....		143
ANEXO B .....		144
UMBRAL DE ILUMINACION .....		144
ANEXO C .....		156

SEGMENTACION DE LA IMAGEN ..... 156

ANEXO D ..... 169

TABLAS RESUMEN DE LA EVALUACIÓN DE LA ILUMINACIÓN SEGÚN SU ESPECTRO..... 169

ANEXO E ..... 171

ANEXO F ..... 169



## Índice de Tablas.

TABLA 1 Características De Los Sistemas Biometricos.....	20
TABLA 2 Margen De Error Según Diferentes Algoritmos De Validación De Palmas.....	42
TABLA 3. Propiedades Generales De Las Imágenes A Evaluar.....	46
TABLA 4 Ejemplo - Arreglo De Posiciones Donde Se Encuentran Los Puntos De Inflexión.....	67
TABLA 5. Relación Del Tipo De Resultado Y Las Variables Independientes.....	95
TABLA 6. Umbrales Para El Reconocimiento De Manos Uniformes Según Espectro.....	101
TABLA 7. Lista De Equipos Utilizados En Las Pruebas.....	113



## Índice de Figuras.

<i>FIGURA 1</i> ESPECTRO ELECTROMAGNÉTICO.....	25
<i>FIGURA 2</i> REPRESENTACIÓN DE LOS ESPACIOS DE COLOR .....	25
<i>FIGURA 3</i> SISTEMA DE COORDENADAS CARTESIANAS DE RGB (ETXEBERRIA) .....	26
<i>FIGURA 4</i> ESPACIO DE COLOR EN RGB .....	28
<i>FIGURA 5</i> IMAGEN DIGITAL.....	29
<i>FIGURA 6</i> FUNCIÓN BIDIMENSIONAL DE INTENSIDAD DE LUZ $F(X,Y)$ . .....	30
<i>FIGURA 7</i> IMAGEN DISCRETIZADA-> MATRIZ.....	31
<i>FIGURA 8</i> FORMACIÓN DE LA IMAGEN: MUESTREO.....	32
<i>FIGURA 9</i> FORMACIÓN DE LA IMAGEN: CUANTIFICACIÓN.....	33
<i>FIGURA 10</i> SEGMENTACIÓN DE LA IMAGEN BASADA EN UMBRALIZACIÓN A LA IZQUIERDA LA IMAGEN ORIGINAL Y A LA DERECHA LA IMAGEN UMBRALIZADA. (ELABORACIÓN PROPIA).....	35
<i>FIGURA 11</i> MASCARA DE 3 X 3 PARA REALCE DEL PIXEL CENTRAL. ....	36
<i>FIGURA 12</i> MASCARA DE 3 X 3 PARA LA DETECCIÓN DE PUNTOS AISLADOS. ....	37
<i>FIGURA 13</i> MÁSCARAS DE LÍNEA.....	37
<i>FIGURA 14</i> OBTENCIÓN DEL BORDE DE LA PALMA DE LA MANO, A LA IZQUIERDA LA IMAGEN SEGMENTADA Y A LA DERECHA EL BORDE DE LA PALMA. ....	38
<i>FIGURA 15</i> MÁSCARA GAUSSIANA 3X3.....	40
<i>FIGURA 16</i> IMAGEN TOMADA PARA LA AUTENTICACIÓN POR MEDIO DE LA BIOMETRÍA DE LA MANO .....	41
<i>FIGURA 17</i> PUNTOS DE INFLEXIÓN Y DISTANCIAS TOMADAS .....	42
<i>FIGURA 18</i> DISPOSITIVO DE RECOLECCIÓN DE IMÁGENES.....	45
<i>FIGURA 19.</i> INDEXACION DE UNA IMAGEN REPRESENTADA EN UNA MATRIZ .....	48
<i>FIGURA 20.</i> PROCESO DE UMBRALIZACIÓN.....	48
<i>FIGURA 21.</i> MÁSCARA PARA LIMPIEZA DE RUIDO EN LA IMAGEN .....	49
<i>FIGURA 22.</i> REPRESENTACIÓN DE LA IMAGEN EN UNA MATRIZ, SOMBREANDO LOS CASILLEROS A EVALUAR. .....	51

FIGURA 23. *DIFERENCIACIÓN DE PÍXELES QUE REPRESENTAN BORDES (AMARILLO)* ..... 52

FIGURA 24. SOMBREADO EN ROJO LOS PÍXELES QUE EVALÚA AL LIMPIAR EL PIXEL DE LA POSICIÓN 8. .... 53

FIGURA 25. (A) IMAGEN TAL Y COMO ES TOMADA (B) IMAGEN LISTA A SER EVALUADA..... 54

FIGURA 26. UBICANDO EL CENTRO DE LA IMAGEN EN LA PARTE INFERIOR DE LA IMAGEN.57

FIGURA 27. REPRESENTACIÓN DE UNA IMAGEN MAQUETADA EN UNA MATRIZ, REPRESENTANDO LAS COORDENADAS DEL PIXEL 38. .... 58

FIGURA 28. REPRESENTACIÓN DE UNA IMAGEN MAQUETADA EN UNA MATRIZ, REPRESENTANDO LAS COORDENADAS DEL PIXEL 31. .... 59

FIGURA 29. REPRESENTACIÓN DE UNA IMAGEN MAQUETADA EN UNA MATRIZ, REPRESENTANDO LAS COORDENADAS DEL PIXEL 24. .... 59

FIGURA 30. UBICACIÓN DE PUNTO MEDIO EN LA PARTE INFERIOR DE LA PALMA. .... 60

FIGURA 31. EVALUACIÓN DEL PIXEL 24 SOMBREANDO TODOS LOS PÍXELES RELACIONADOS A ESTE ..... 61

FIGURA 32. REPRESENTACIÓN DEL ORDEN DE EVALUACIÓN EN SENTIDO HORARIO DE LOS PÍXELES A SU ALREDEDOR..... 61

FIGURA 33. *SOMBREADO DE LOS PÍXELES A SER EVALUADOS.* ..... 62

FIGURA 34. ACERCAMIENTO DEL CONTORNO DE PALMA. .... 63

FIGURA 35. PUNTOS DE INFLEXIÓN DE LA PALMA IZQUIERDA. .... 66

FIGURA 36. PUNTOS DE INFLEXIÓN FALSOS..... 68

FIGURA 37. REPRESENTACIÓN GRÁFICA DEL CÁLCULO DEL PRIMER PUNTO DE INFLEXIÓN. .... 69

FIGURA 38. INSERCIÓN DEL PRIMER PUNTO DE INFLEXION FALSO. .... 69

FIGURA 39. INSERCIÓN DEL SEGUNDO PUNTO DE INFLEXIÓN FALSO (DEDO ÍNDICE). .... 70

FIGURA 40. INSERCIÓN DEL TERCER PUNTO DE INFLEXIÓN (DEDO PULGAR). .... 71

FIGURA 41. REPRESENTACIÓN GRÁFICA DE DONDE SE ENCUENTRAN LOS PUNTOS DE INFLEXIÓN. .... 71

<i>FIGURA 42. POSICIONES TOMADAS EN CUENTA PARA HALLAR EL CENTRO DEL DEDO MEÑIQUE</i> .....	75
<i>FIGURA 43. POSICIONES TOMADAS EN CUENTA PARA HALLAR EL CENTRO DEL DEDO ANULAR.</i> .....	76
<i>FIGURA 44. POSICIONES TOMADAS EN CUENTA PARA HALLAR EL CENTRO DEL DEDO MEDIO.</i> .....	77
<i>FIGURA 45. POSICIONES TOMADAS EN CUENTA PARA HALLAR EL CENTRO DEL DEDO ÍNDICE.</i> .....	78
<i>FIGURA 46. POSICIONES TOMADAS EN CUENTA PARA HALLAR EL CENTRO DEL DEDO PULGAR.</i> .....	79
<i>FIGURA 47. GENERACIÓN GRAFICA DE LOS PUNTOS DE INFLEXIÓN Y LOS CENTROS.</i> .....	80
<i>FIGURA 48. UBICACIÓN GRAFICA DE LOS PUNTOS DE INFLEXIÓN Y CENTROS ENUMERADOS.</i> .....	83
<i>FIGURA 49. DISTANCIAS GENERADAS PARA EL DEDO MEÑIQUE</i> .....	85
<i>FIGURA 50. SEÑALIZACIÓN DE LOS CUADRANTES GENERADOS AL GENERAR LAS DISTANCIAS DEL CENTRO DEL DEDO CON SUS RESPECTIVOS PUNTOS DE INFLEXIÓN.</i> .....	86
<i>FIGURA 51. REPRESENTACIÓN DE LAS 16 DISTANCIAS 8 POR CADA SEGMENTO DE CADA DEDO.</i> .....	86
<i>FIGURA 52. PUNTOS DE INFLEXIÓN NECESARIOS PARA HALLAR EL CENTRO DE LA PALMA DE LA MANO.</i> .....	87
<i>FIGURA 53. REPRESENTACIÓN GRÁFICA DE LAS DISTANCIAS GENERADAS A PARTIR DE LOS CENTROS.</i> .....	88
<i>FIGURA 54. REPRESENTACIÓN DE TODAS LAS DISTANCIAS (ROJO) DISTANCIA DEL CENTRO A LOS CONTORNOS DE LOS DEDOS (AZUL) GROSOR DE LOS DEDOS (VERDE) CENTRO DE LA PALMA.</i> .....	89
<i>FIGURA 55. PANTALLA CON RESULTADOS.</i> .....	90
<i>FIGURA 56. PANTALLA DE BATCH EN CONSOLA DE AUTENTIFICACIÓN</i> .....	91
<i>FIGURA 57. DIAGRAMA DE CASOS DE USO DEL MECANISMO IMPLEMENTADO.</i> .....	92
<i>FIGURA 58. PANTALLA INICIAL DEL MECANISMO.</i> .....	94

<i>FIGURA 59.</i> OPCIONES DE USO LIBRERÍA TBB.....	96
<i>FIGURA 60.</i> OPCIÓN DE NÚMERO DE SEGMENTACIONES. ....	97
<i>FIGURA 61.</i> OPCIÓN DEL GRADO DE ILUMINACIÓN.....	98
<i>FIGURA 62.</i> ESQUEMA DE LAS SESIONES TOMADAS PARA LA OBTENCION DE LA IMAGEN. ....	100
<i>FIGURA 63.</i> PICOS DEL CONTORNO DE LA PALMA. ....	102
<i>FIGURA 64.</i> ERRORES QUE DESVIAN EL VERDARERO CONTORNO DE LA PALMA. ....	103
<i>FIGURA 65.</i> EXISTEN PICOS QUE NO PERMITEN GENERAR LOS PUNTOS DE INFLEXIÓN COMPLETOS.....	104
<i>FIGURA 66.</i> PICOS QUE SON COMNFUNDIDOS POR DEDOS.....	104
<i>FIGURA 67.</i> GRÁFICO DE LÍNEAS QUE REPRESENTA LA CANTIDAD DE ERRORES SEGÚN ILUMINACIÓN - SEGMENTACIÓN DEL ESPECTRO 460.....	105
<i>FIGURA 68.</i> GRÁFICO DE LÍNEAS QUE REPRESENTA LA CANTIDAD DE ERRORES SEGÚN ILUMINACIÓN - SEGMENTACIÓN DEL ESPECTRO 630. ....	106
<i>FIGURA 69.</i> GRÁFICO DE LÍNEAS QUE REPRESENTA LA CANTIDAD DE ERRORES SEGÚN ILUMINACIÓN - SEGMENTACIÓN DEL ESPECTRO 700. ....	106
<i>FIGURA 70.</i> GRÁFICO DE LÍNEAS QUE REPRESENTA LA CANTIDAD DE ERRORES SEGÚN ILUMINACIÓN - SEGMENTACIÓN DEL ESPECTRO 850. ....	107
<i>FIGURA 71.</i> GRÁFICO DE LÍNEAS QUE REPRESENTA LA CANTIDAD DE ERRORES SEGÚN ILUMINACIÓN - SEGMENTACIÓN DEL ESPECTRO 940. ....	107
<i>FIGURA 72.</i> GRÁFICO DE LÍNEAS QUE REPRESENTA LA CANTIDAD DE ERRORES SEGÚN ILUMINACIÓN - SEGMENTACIÓN DEL ESPECTRO BLANCO. ....	108
<i>FIGURA 73.</i> MARGEN DE ERROR AL EVALUAR LAS PALMAS CON UN ESPECTRO DE 460 Y UN UMBRAL DE ILUMINACIÓN DE 50. ....	109
<i>FIGURA 74.</i> MARGEN DE ERROR AL EVALUAR LAS PALMAS CON UN ESPECTRO DE 630 Y UN UMBRAL DE ILUMINACIÓN DE 55. ....	109
<i>FIGURA 75.</i> MARGEN DE ERROR AL EVALUAR LAS PALMAS CON UN ESPECTRO DE 700 Y UN UMBRAL DE ILUMINACIÓN DE 45. ....	109
<i>FIGURA 76.</i> MARGEN DE ERROR AL EVALUAR LAS PALMAS CON UN ESPECTRO DE 700 Y UN UMBRAL DE ILUMINACIÓN DE 50. ....	110

*FIGURA 77.* MARGEN DE ERROR AL EVALUAR LAS PALMAS CON UN ESPECTRO DE 700 Y UN UMBRAL DE ILUMINACIÓN DE 55. .... 110

*FIGURA 78.* MARGEN DE ERROR AL EVALUAR LAS PALMAS CON UN ESPECTRO DE 850 Y UN UMBRAL DE ILUMINACIÓN DE 50. .... 110

*FIGURA 79.* MARGEN DE ERROR AL EVALUAR LAS PALMAS CON UN ESPECTRO DE 940 Y UN UMBRAL DE ILUMINACIÓN DE 90. .... 110

*FIGURA 80.* MARGEN DE ERROR AL EVALUAR LAS PALMAS CON UN ESPECTRO DE BLANCO Y UN UMBRAL DE ILUMINACIÓN DE 50. .... 111

*FIGURA 81.* PORCENTAJE DE ERROR EN COMPARACIÓN A TODAS LAS EVALUACIONES ESPECTRO ILUMINACIÓN Y SEGMENTACIÓN. .... 112

*FIGURA 82.* COMPARACIÓN EN EL TIEMPO TOTAL UTILIZADO EN LA SIMULACIÓN. .... 114



## RESUMEN

Hoy en día el proceso de autenticación de personas es de vital importancia para asegurar el correcto acceso a los recursos de una institución.

Las técnicas de autenticación basadas en claves y usuarios así como las técnicas basadas en tarjetas de acceso, no garantizan la autenticidad de la persona. Es así que las técnicas de autenticación biométricas surgen como una solución, las cuales permiten asegurar la identidad de la persona, utilizando para ello la medición de alguna característica biológica propia de la persona. Sin embargo el inconveniente del uso de estas técnicas es el alto consumo de recursos computacionales debido al gran volumen de información que se debe almacenar, pre-procesar y a partir de ella obtener la información más relevante a utilizar.

El presente proyecto propone el uso de técnicas de computación paralelas que frente a las técnicas seriales tienen la ventaja de generar una mejora en los tiempos de ejecución, debido a que se basan en la utilización de bloques de construcción de hebras de software y procesamiento de imágenes digitales.

Debido a que hoy en día las técnicas de computación paralelas han llegado a convertirse en un conjunto de tecnologías robustas. La presente propuesta tomando como base estas tecnologías pretende desarrollar un mecanismo de software que recibirá como entrada imágenes de manos humanas las cuales procesará y extraerá la información pertinente para realizar un proceso de autenticación con usuarios previamente registrados.

**Palabras Claves.** Técnicas De Autenticación, Autenticación Biométricas, Técnicas De Computación Paralelas, Bloques De Construcción De Hebras.

## ABSTRACT

Nowadays the process of authentication of people is vital to ensure proper access to the resources of an institution.

Authentication techniques based on keys and users, and the techniques based on access cards do not guarantee the authenticity of the person. Thus biometric authentication techniques emerge as a solution, which allows to ensure the identity of the individual, using the measurement of any biological characteristic of the person itself. However, the drawback of using these techniques is the high consumption of computational resources due to the large volume of information to be stored, pre-processing and from it get the most relevant information to use.

This draft proposes the use of parallel computing techniques versus serial techniques have the advantage of generating an improvement in execution time, because it is based on the use of threads building blocks and image processing software digital.

Because today parallel computing techniques have come has become a robust set of technologies. This proposal is based on these technologies is to develop a software prototype that receive as input images of human hands which will process and extract relevant information to perform an authentication process with previously registered users.

**keywords.** Authentication Techniques, Biometric Authentication Techniques, Parallel Computing Techniques, Threads Building Blocks.

## INTRODUCCION

Un sistema biométrico funciona almacenando y comparando características biométricas, en muchos casos estas características son almacenadas como imágenes, archivos de sonido, series de datos en el tiempo, etcétera. En este alcance dichas características preservan la información biométrica para ser utilizada en cualquier tipo de sistema. En un segundo alcance por razones de eficiencia más que utilizar características biométricas almacenadas directamente se realiza un procesamiento para codificar dichas características y de esta manera poder eficientemente almacenarlas y compararlas.

Cuando un individuo utiliza un sistema biométrico las características biométricas codificadas son almacenadas y referenciadas para una posterior comparación, esta referencia puede estar en una base de datos centralizada o sobre otro tipo de dispositivo de almacenamiento, dependiendo de las necesidades de la aplicación.

Cuando las características biométricas de un individuo tienen un grado de similitud muy alto con una referencia en la base de datos principal se realiza un proceso denominado “matching” el cual le indica al sistema la identidad del individuo que está utilizándolo.

Existen dos modos de reconocimiento biométrico, la verificación y la identificación. En la verificación el individuo ingresa una identidad y sus características biométricas, el proceso de “matching” debe limitarse a dar por validada esta entrada en la base de datos de referencia. En la Identificación no se necesita una identidad, el sistema realiza una búsqueda de rango amplio utilizando las características ingresadas para encontrar la identidad del individuo.

En el campo de la investigación biométrica, la combinación de diversas modalidades de procesamiento de imágenes, ha demostrado ser una opción prometedora en la mejora del reconocimiento de personas por medio de la biometría de mano.

Para la realización de pruebas en el sistema se utilizó una base de datos de imágenes obtenidas del Laboratorio Nacional Del Instituto De Reconocimiento De Patrones De Automatización, De La Academia De Ciencias De China, este laboratorio ha diseñado un dispositivo de imagen multi-espectral para capturar información de forma correlativa y complementaria de manos humanas denominado CASIA Multi-SpectralPalmpImageDatabase V1.0 (o CASIA-MS-PalmpImageV1) el cual ha sido puesto a disposición del público para promover la investigación y el progreso en las técnicas de imágenes multiespectrales dentro de las técnicas de modalidades biométricas.

La presente tesis propone el desarrollo de un mecanismo de sistema de software para la autenticación de individuos basándose en la geometría de la mano, este estudio permite explorar el uso de la geometría de la mano como una medida para diferenciar la identidad de una persona, en el primer capítulo se exponen los fundamentos teóricos necesarios para realizar la investigación, en el segundo capítulo se presentan las metodologías y herramientas utilizadas para el estudio, en el capítulo se presenta el desarrollo experimental del mecanismo de software, y por último se presentan las pruebas del mecanismo experimental, conclusiones, recomendaciones para trabajos futuros y las referencias de la tesis.

## CAPITULO I: FUNDAMENTOS TEORICOS

### 1.1 MARCO CONCEPTUAL:

#### 1.1.1 Identificación Vs. Autenticación

La identificación personal es asociar una identidad con un individuo. El problema del reconocimiento de un individuo se puede categorizar de las siguientes dos maneras (a) verificación, e (b) identificación.

La verificación (autenticación) se refiere al hecho de confirmar o denegar la solicitud de una persona que reclama su identidad (¿Soy la persona quien digo ser?). Y la identificación se refiere al hecho de establecer una identidad (¿Quién soy?). (Pankanti, 1999)

#### 1.1.2 Biometría.

Este término se puede utilizar tanto para describir una característica como para describir un proceso.

**a. Como Característica.** Es una medición biológica (anatómica y fisiológica) así como también de las características del comportamiento que pueden ser utilizadas para automatizar el reconocimiento.

**b. Como Proceso.** Es un método automatizado para el reconocimiento de individuos basado en medidas biológicas y características del comportamiento.

#### 1.1.3 Información Biométrica.

Es la información que es usada para describir los datos recolectados durante los procesos de inscripción, verificación o identificación pero no se aplica para el enlace del nombre del usuario, información demográfica ni de autorización.

### 1.1.4 Sistema Biométrico.

Es un conjunto de múltiples componentes (tales como sensores, algoritmos y formulario de resultado) que hacen un sistema operacional completo. Un sistema automatizado de biometría debe ser capaz de:

- a. Obtener un conjunto de ejemplos de los usuarios finales.
- b. Extraer y procesar la información obtenida.
- c. Comparar la información biométrica con la información contenida en una o más referencias.

Para que un sistema biométrico sea eficiente, los indicadores o rasgos personales objeto de estudio deben reunir las siguientes cualidades:

- a. **Permanencia.** La característica no debe cambiar con el tiempo, o hacerlo muy lentamente.
- b. **Unicidad.** la existencia de dos personas con una característica idéntica debe tener una probabilidad muy pequeña.
- c. **Universalidad.** Cualquier persona debe poseer esa característica.
- d. **Cuantificación.** La característica puede ser medida de forma cuantitativa.

La Tabla 1 se muestra los diferentes sistemas biométricos con sus respectivas características.

Tabla 1.  
*Características de los Sistemas Biométricos*

	Ojo (Iris)	Ojo (Retina)	Huellas dactilares	Vascular dedo	Vascular mano	Geometría de la mano	Escritura y firma	Voz	Cara 2D	Cara 3D
<b>Fiabilidad</b>	Muy alta	Muy Alta	Muy Alta	Muy Alta	Muy Alta	Alta	Media	Alta	Media	Alta
<b>Facilidad de uso</b>	Media	Baja	Alta	Muy Alta	Muy Alta	Alta	Alta	Alta	Alta	Alta
<b>Prevención de ataques</b>	Muy alta	Muy Alta	Alta	Muy Alta	Muy Alta	Alta	Media	Media	Media	Alta
<b>Aceptación</b>	Media	Baja	Alta	Alta	Alta	Alta	Muy Alta	Alta	Muy alta	Muy alta
<b>Estabilidad</b>	Alta	Alta	Alta	Alta	Alta	Media	Baja	Media	Media	Alta

elaborado por: Marino Tapiador, Juan Alberto Sigüenza, "Tecnologías biométricas aplicadas a la seguridad"

### 1.1.5 Paralelismo en la CPU.

Aunque la ejecución de los programas en una computadora se percibe de forma secuencial (leyendo instrucciones de la memoria y ejecutando una por una), internamente se tiene un paralelismo en ciertas etapas de esta ejecución, a nivel de proceso un proceso escribe en disco, mientras otro utiliza el procesador, a nivel de micro operación.

Intel Threading Building Blocks (Intel TBB) explota las capacidades de paralelismo de los procesadores con arquitectura multinúcleo. Gestionado los hilos de ejecución, evitando la creación, sincronización y destrucción de los hilos que es explícita y dependiente del sistema.

Su librería trabaja en base a la descomposición recursiva de problemas y del robo de tareas.

### 1.1.6 Multiprocesamiento.

Es una tendencia significativa en el campo de la computación consiste en configurar un sistema de computación con varios procesadores, no es un enfoque nuevo pero sí posee grandes perspectivas en función del desarrollo de los microprocesadores, se podrían concebir sistemas contruidos por cientos o miles de microprocesadores.

Si un procesador falla, los restantes continúan operando, lo cual no es automático y requiere de un diseño cuidadoso, un procesador que falla habrá de informarlo a los demás de alguna manera, para que se hagan cargo de su trabajo, los procesadores en funcionamiento deben poder detectar el fallo de un procesador determinado, el Sistema Operativo debe percibir que ha fallado un procesador determinado y ya no podrá asignarlo y también debe ajustar sus estrategias de asignación de

recursos para evitar la sobrecarga del sistema que está degradado. Los multiprocesadores hacen posible la explotación del paralelismo.

Los sistemas de computación obtienen los beneficios del procesamiento concurrente más por la “multiprogramación” de varios procesos y menos por la explotación del “paralelismo” dentro de un solo proceso.

La detección del paralelismo es un problema complejo y la puede efectuar el programador, el traductor del lenguaje, el hardware o el Sistema Operativo. El paralelismo dentro de los programas puede ser “explícito” o “implícito”.

Las principales características del paralelismo explícito son las que se detallan a continuación:

- a. Es indicado de forma específica por un programador mediante una “construcción de concurrencia”.
- b. Se pueden utilizar procesadores separados para ejecutar cada una de las proposiciones.
- c. Es susceptible de errores de programación difíciles de detectar y depurar.
- d. El programador puede omitir tratar situaciones donde sería aplicable el paralelismo.

Las principales características del paralelismo implícito son las que se detallan a continuación:

- a. La verdadera esperanza está en la detección automática del paralelismo implícito.
- b. Es el paralelismo intrínseco del algoritmo pero no establecido explícitamente por el programador.

c. Los compiladores explotan el paralelismo implícito mediante las técnicas de “distribución de ciclos” y de “reducción de la altura del árbol”.

### **1.1.7 Intel Threading Building Blocks**

Intel Threading Building Blocks (Intel TBB) es una biblioteca basada en plantillas para C++ desarrollada por Intel para facilitar la escritura de programas que exploten las capacidades de paralelismo de los procesadores con arquitectura multinúcleo.

La biblioteca abstrae el acceso a los múltiples procesadores permitiendo que las operaciones sean tratadas como tareas que se reparten automática y dinámicamente entre los procesadores disponibles mediante un gestor en tiempo de ejecución.

Esta aproximación hace que Intel TBB se incluya en la familia de soluciones para la programación paralela que permiten desacoplar la programación de las características particulares de la máquina.

### **1.1.8 Procesamiento digital de imágenes.**

El termino procesamiento digital de imágenes va dirigido a las técnicas de manipulación y análisis de imágenes por computadora. Este procesamiento puede considerarse como un tipo especial del procesamiento digital en dos dimensiones, el cual se usa para revelar información sobre imágenes y que involucra hardware, software y soporte teórico. La imagen puede haber sido generada de muchas maneras, por ejemplo, fotográficamente, o electrónicamente por medio de monitores de televisión, este procesamiento también se puede hacer por medio de métodos ópticos o bien por medio de métodos digitales

utilizando una computadora.

Los principios fundamentales del procesamiento digital de imágenes están establecidos hace muchos años, pero no se llevaban a cabo debido a la falta de procesamiento en las computadoras pero con la aparición de las computadoras de alta capacidad y memoria, era natural que se comenzara a desarrollar este campo.

**a. Pixel.**

Acónimo formado con las palabras inglesas 'Picture' 'element'. Es el elemento simple de una imagen digital. Es la parte más pequeña que se muestra y que se le puede asignar un color.

**b. Pixel por pulgada (PPI – Pixel Per Inch).**

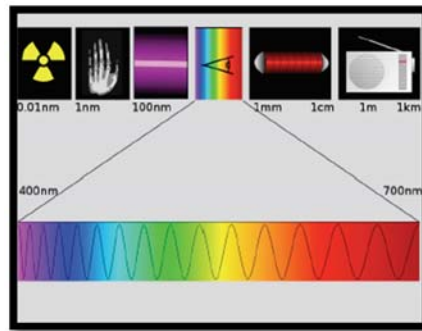
Es la medida de la resolución de una imagen digital. Mientras más alta sea su PPI más información tiene la imagen y también es mayor el tamaño de la imagen.

**c. Resolución.**

En el ámbito de la digitalización, la resolución se refiere al número de bits que componen la palabra digital.

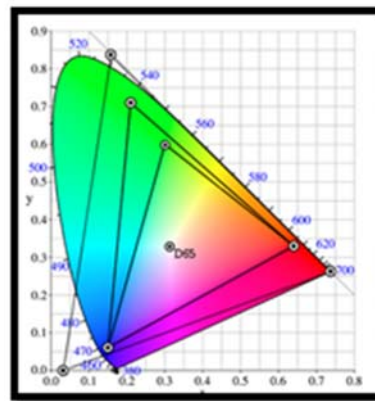
**d. Espacios de Color.**

El ojo humano es capaz de percibir una parte del espectro electromagnético. Es lo que se llamaba el espectro de luz visible, los rayos ultravioleta o los rayos gamma queda fuera de este espectro de luz visible (existen, pero no podemos verlos) como se muestra en la *Figura 1*.



*Figura 1* Espectro Electromagnético

Al igual que el ojo es incapaz de ver zonas del espectro electromagnético los dispositivos que habitualmente se utilizan para manipular y presentar imágenes tales como pantallas de televisión u ordenador e impresoras son incapaces de representar el espectro de luz visible.



*Figura 2* Representación de los espacios de color

Existen diferentes espacios de color y cada uno de ellos tiene capacidades de representación distintas, con mayor o menor limitación de la gama de colores que puede mostrar cada uno.

En la *Figura 2* vemos un diagrama cromático donde el triángulo interior son los colores representados por el espacio sRGB, el intermedio, los

representados por el Adobe RGB y el triángulo exterior indica los colores que ProPhoto RGB es capaz de representar. Como se puede ver, hay una gran mejora de uno a otro.

**e. Espacio de color RGB.**

En el Espacio de color RGB cada color aparece en sus componentes espectrales primarias: rojo, verde y azul. Este modelo está basado en el sistema de coordenadas cartesianas como se muestra en la Figura 3.

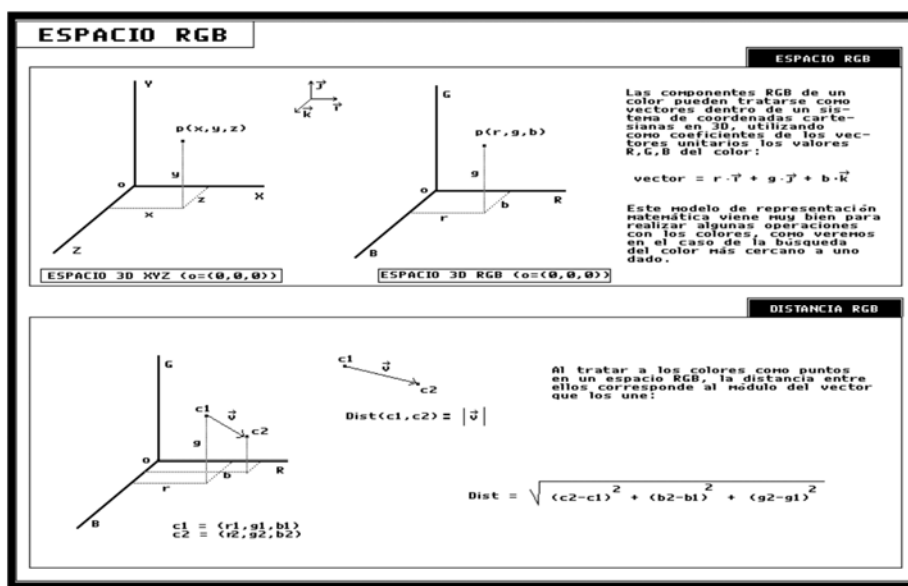


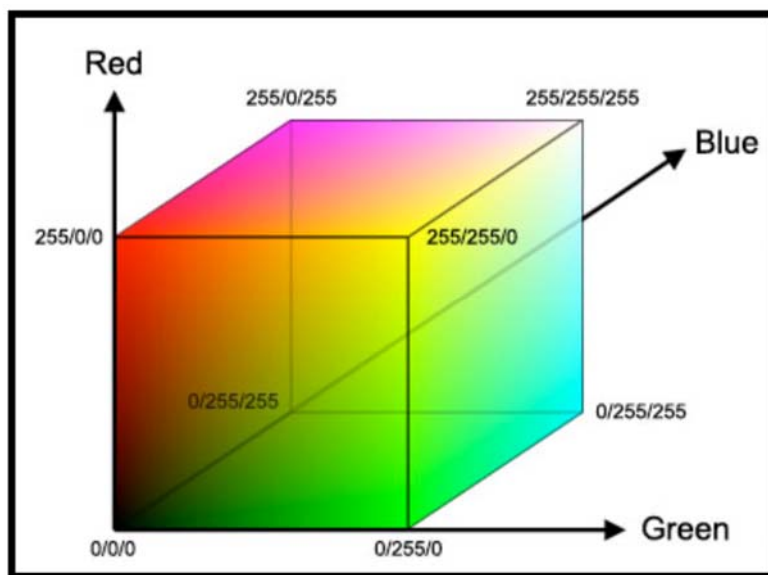
Figura 3 Sistema de coordenadas cartesianas de RGB (Etxeberria)

El sub-espacio de color de interés es el tetraedro mostrado en la Figura 4, en el cual los valores RGB están en tres vértices; cian, magenta y amarillo que se sitúan en otros tres vértices, el negro corresponde al origen y el blanco se sitúa en el vértice más alejado del origen. En este modelo, la escala de grises se extiende desde el negro al blanco a lo

largo de la diagonal que une esos dos puntos, y los colores son puntos dentro del tetraedro definidos por los vectores desde el origen. Por conveniencia, se asume que todos los vectores han sido normalizados, de modo que el tetraedro de la *Figura 4* es el tetraedro unitario, es decir, todos los valores de R, G y B están en el rango  $[0,1]$ . Las imágenes en este modelo se forman por la combinación en diferentes proporciones de cada uno de los colores primarios RGB. Para indicar con qué proporción mezclamos cada color, se asigna un valor a cada uno de los colores primarios, de manera, por ejemplo, que el valor 0 significa que no interviene en la mezcla y, a medida que ese valor aumenta, se entiende que aporta más intensidad a la mezcla. Aunque el intervalo de valores podría ser cualquiera (valores reales entre 0 y 1, valores enteros entre 0 y 37, etc.), es frecuente que cada color primario se codifique con un byte (8 bits), así de manera usual la intensidad de cada una de las componentes se mide según una escala que va del 0 al 255. (Sevilla)

Por lo tanto como se aprecia en la *Figura 4*, el rojo se obtiene con  $(255, 0, 0)$ , el verde con  $(0, 255, 0)$  y el azul con  $(0, 0, 255)$ , obteniendo, en cada caso un color resultante monocromático. La ausencia de color (negro) se obtiene cuando las tres componentes son 0,  $(0, 0, 0)$ . La combinación de dos colores a nivel 255 con un tercero en nivel 0 da lugar a tres colores intermedios. De esta forma el amarillo es  $(255, 255, 0)$ , el cian  $(0, 255, 255)$  y el magenta  $(255, 0, 255)$ . Obviamente, el color blanco se forma con los tres colores primarios a su máximo nivel  $(255, 255, 255)$ . El conjunto de todos los colores se puede

representar en forma del cubo de la *Figura 4*. Cada color es un punto de la superficie o del interior de éste. La escala de grises estaría situada en la diagonal que une al color blanco con el negro.



*Figura 4* Espacio de color en RGB

#### **f. Imagen Digital**

El procesamiento digital de imágenes se efectúa dividiendo la imagen en un arreglo rectangular de elementos, cada elemento de la imagen así dividida se conoce con el nombre de pixel, el siguiente paso es asignar un valor numérico a la luminosidad promedio de cada pixel, así los valores de la luminosidad de cada pixel con sus coordenadas que indican su posición, definen completamente la imagen.

El término imagen (representación bidimensional de un objeto tridimensional) se utiliza para describir a una función bidimensional de intensidad de luz  $f(x,y)$ , donde  $x$  e  $y$  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 (Iturrate, 1998) ya que La imagen no depende sólo de la escena visual sino también de las condiciones de iluminación.

Donde un modelo sencillo de imagen visual se puede representar de la siguiente manera:

$$f(x, y) = i(x, y)r(x, y)$$

Donde

$0 < i(x, y) < \infty$ , iluminación

$0 < r(x, y) < 1$ , reflectancia

En la Figura 5 se aprecia una imagen digital en color, en este caso cada par de coordenadas discretas  $(x, y)$  tiene un valor de color en el espacio elegido, si el espacio de color fuera RGB los valores serían las tres componentes de este espacio.

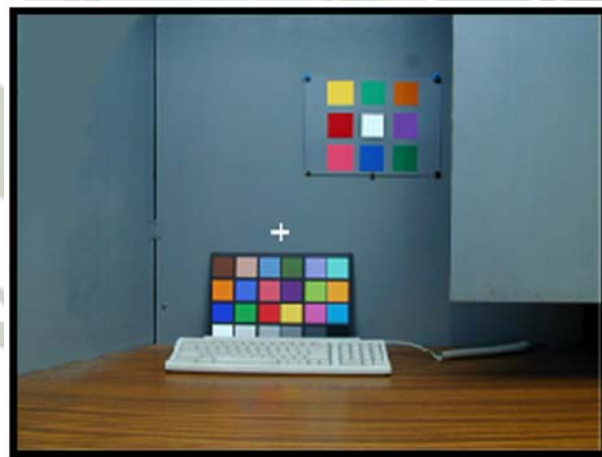


Figura 5 Imagen Digital

Como se observa en la Figura 6, una imagen digital es una función  $f(x, y)$  que ha sido discretizada en coordenadas espaciales y en brillo y se puede considerar como una matriz cuyos índices de fila y columna identifican a un punto en la imagen y el correspondiente valor del

elemento de la matriz que identifica el nivel de intensidad de luz en ese punto como se muestra en la Figura 7, los elementos de tal arreglo digital son llamados elementos de imagen, elementos de pintura, pixel o pel (estos dos últimos son abreviaturas del inglés pictureelements), el nivel de brillo, o valor de cada pixel, es representado por códigos binarios enteros positivos (el brillo no puede ser negativo). (Fraga, 7 de mayo de 2001)

Es decir:

- $x, y$ : Son las coordenadas espaciales.
- $f$ : Es el valor proporcional al brillo.

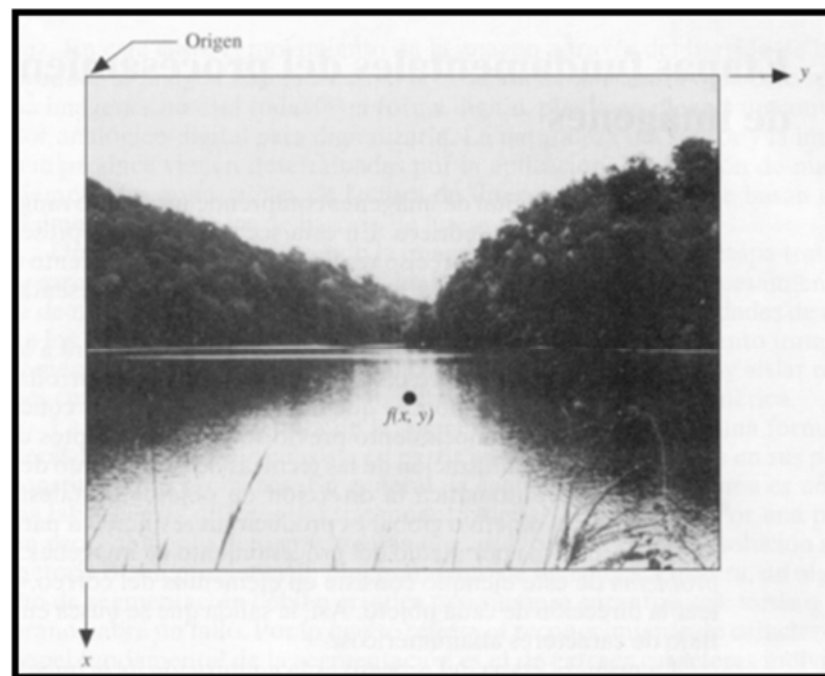


Figura 6 función bidimensional de intensidad de luz  $f(x,y)$ .

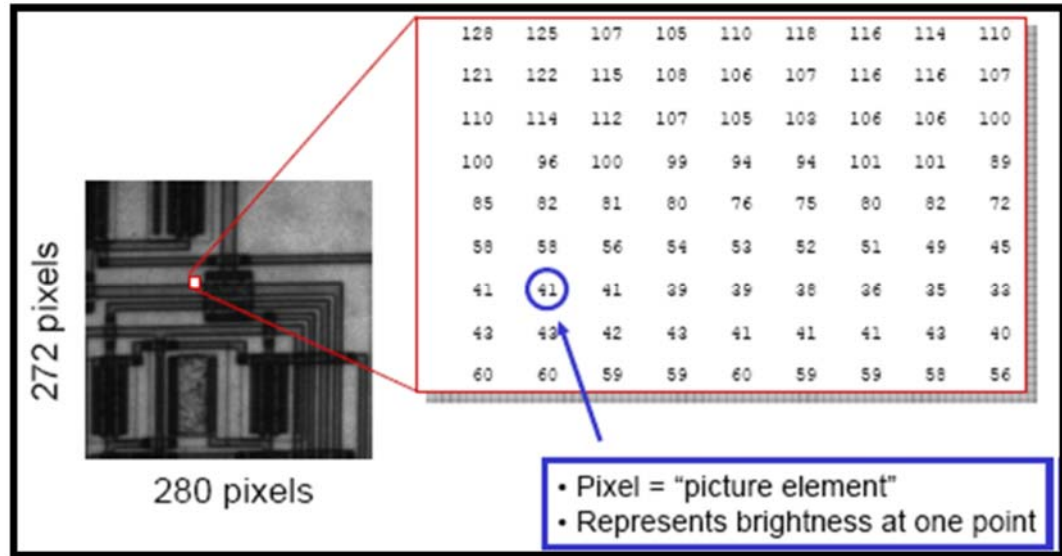


Figura 7 Imagen discretizada-> Matriz

Con 5 bits en la intensidad se pueden representar 32 niveles de gris ( $2^5 = 32$ ). Para tener una imagen de buena calidad para el ojo humano es necesario tener como mínimo 64 niveles de gris. (EMMANUELLE, 2007). Para una apreciación suave se usa una representación estándar a 8 bits, esto es,  $2^8=256$  niveles de gris. A una imagen que utilice este estándar se le conoce como imagen en tonos de gris. La representación de la imagen a 8 bits (1 byte) se dice estándar porque es el mínimo número de bits que puede ser procesado directamente por la mayoría de los microprocesadores. Como se muestra en la Figura 8.

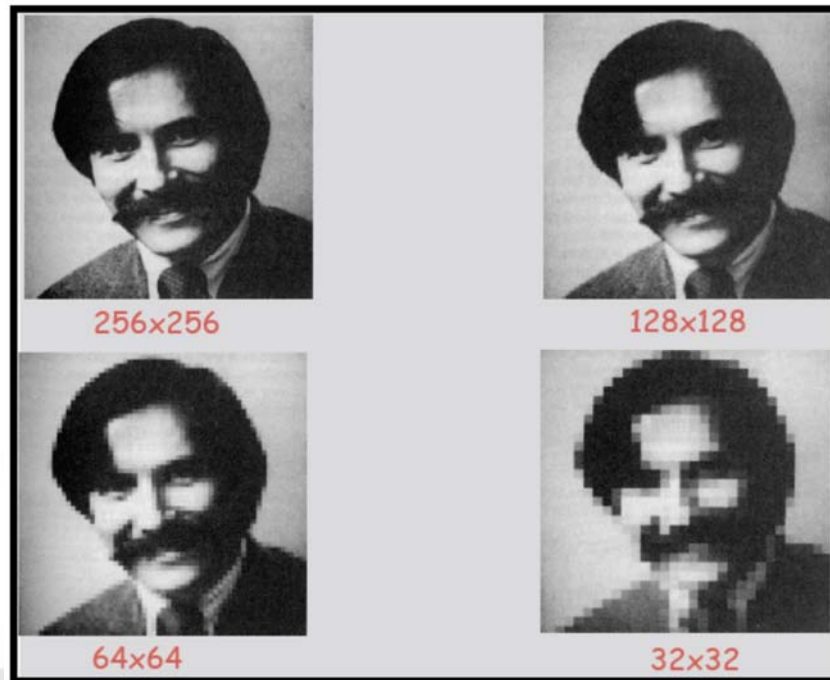


Figura 8 Formación de la imagen: Muestreo

Una imagen binaria es una imagen monocromática que tiene representación de 1 bit por pixel, esto es, dos niveles, blanco o negro. En el proceso y análisis de imagen, la binarización se emplea para separar las regiones u objetos de interés en una imagen del resto. Las imágenes binarias se usan en operaciones booleanas o lógicas para identificar individualmente objetos de interés o para crear máscaras sobre regiones.

En muchos casos, una imagen binaria es el resultado de una segmentación por niveles de gris o de una segmentación por selección de un rango de color determinado. En otros casos, una imagen binaria es simplemente el resultado de una selección interactiva de regiones de interés, las cuales se utilizarán como mascararas de comparación o referencia.



Figura 9 Formación de la imagen: Cuantificación

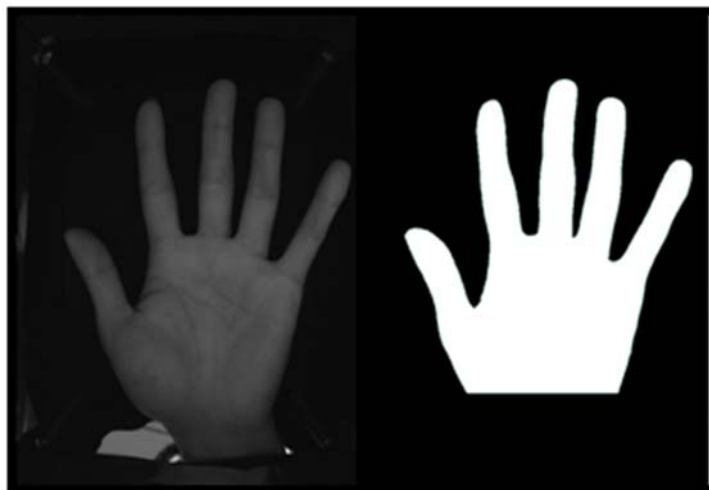
### 1.1.9 Métodos de digitalización de imágenes.

#### a. Segmentación de Imágenes.

El proceso de segmentación es el que consume más recursos computacionales y tiempo en una aplicación de procesamiento de imágenes, es así que existen diversos alcances para lograr este objetivo, el algoritmo de segmentación Mean-Shift (Dorin Comaniciu, 2002) que busca los picos que representan los centros o valores

ejemplares de cada tipo de segmento, determinar cuántos son y agrupar o etiquetar los píxeles según el centro al que se parezcan más, este algoritmo ha sido muy utilizado en la comunidad de visión artificial ya que permite crear clusters sin la necesidad de un parámetro inicial, este algoritmo tiene dos pasos fundamentales, el filtrado y la clusterización, el algoritmo de Segmentación eficiente basada en grafos (P. Felzenszwalb, 2004) donde Cada píxel se toma como un nodo del grafo y los arcos indican el grado de similitud entre los nodos. Que a diferencia del algoritmo Mean-Shift este no necesita un paso de filtrado previo, y su éxito radica en el método denominado umbralización adaptativa, existe un método híbrido que combina los algoritmos de algoritmo Mean-Shift y el algoritmo de Segmentación eficiente basada en grafos denominado Algoritmo de Segmentación híbrido (Caroline Pantofaru, 2005) el cual utiliza el proceso de filtrado del Mean-Shift y luego se aplica la clusterización de la segmentación eficiente basada en grafos.

En la presente tesis se utiliza un paso de segmentación basado en umbralización para separar el objeto principal (palma de la mano) del fondo de la imagen; para este objetivo se utilizan los niveles de color de cada componente de los píxeles, tomando como referencia que el fondo es de color negro se utiliza un umbral para discernir si un píxel pertenece al fondo de la imagen o a la palma de la mano. En la *Figura 10* se muestra el proceso de segmentación basado en umbralización para obtener la región que conforma la palma de la mano.



*Figura 10* Segmentación de la imagen basada en umbralización a la izquierda la imagen original y a la derecha la imagen umbralizada. (Elaboración propia)

#### **b. Detección de discontinuidades.**

El método más común de buscar discontinuidades es la correlación de la imagen con una máscara (Martin, 21 de mayo de 2002). En la *Figura 11* se puede ver un caso general de máscara de  $3 \times 3$ . En este procedimiento se realiza el producto de los elementos de la máscara por el valor de gris correspondiente a los píxeles de la imagen encerrados por la máscara. La respuesta a la máscara de cualquier píxel de la imagen viene dado por:

$$R = \sum_{i=1}^9 w_i z_i$$

Donde  $Z_i$  es el nivel de gris asociado al píxel de la imagen con coeficiente de la máscara  $W_i$ , como suele ser habitual, la respuesta de la máscara viene referida a su posición central. Cuando la máscara esté centrada en un píxel de borde de la imagen, la respuesta se determina empleando el vecindario parcial apropiado. La elección del

tamaño de la máscara que filtra la imagen depende de la calidad del filtro necesario y de la eficiencia con la que se realiza este proceso, ambas características son inversas, es decir, si se desea mejorar el proceso en calidad el tiempo crece y viceversa.

-1	-1	-1
0	6	0
-1	-1	-1

Figura 11 Mascara de 3 x 3 para realce del pixel central.

### c. **Detección de puntos Aislados.**

La detección de puntos aislados es inmediata. Empleando la máscara de la *Figura 12*, si se ha detectado un punto en la posición en la cual está centrada la máscara básicamente se mide la diferencia entre el pixel central y sus vecinos, puesto que un pixel será un punto aislado siempre que sea suficientemente distinto de sus vecinos. Solamente se considerarán puntos aislados aquellos cuya diferencia con respecto a sus vecinos sea significativa.

Para esto se utiliza un valor umbral, si el resultado R del filtrado del pixel central con su vecindario es mayor a dicho umbral este punto no se considera un punto aislado.

$$R = \sum_{i=0}^9 W_i Z_i$$

Si  $R > \text{umbral}$ , entonces el punto no es aislado.

Si  $R < \text{umbral}$ , entonces el punto es aislado.

Donde  $Z_i$  es el nivel de gris asociado al pixel de la imagen con coeficiente de la máscara  $W_i$ , como suele ser habitual, la respuesta de la máscara viene referida a su posición central.

-1	-1	-1
-1	9	-1
-1	-1	-1

Figura 12 Mascara de 3 x 3 para la detección de puntos aislados.

#### d. Detección de líneas.

En este caso se consideran las máscaras de la *Figura 13*. Si pasamos la primera de las máscaras a lo largo de la imagen, tendrá mayor respuesta para líneas de ancho un pixel orientadas horizontalmente. Siempre que el fondo sea uniforme, la respuesta será máxima cuando la línea pase a lo largo de la segunda fila de la máscara. La segunda máscara de la *Figura 13* respondería mejor a líneas orientadas a  $45^\circ$ ; la tercera máscara a líneas verticales; y la última a líneas orientadas a  $-45^\circ$ . Estas direcciones se pueden establecer observando que para la dirección de interés las máscaras presentan valores mayores que para otras posibles direcciones.

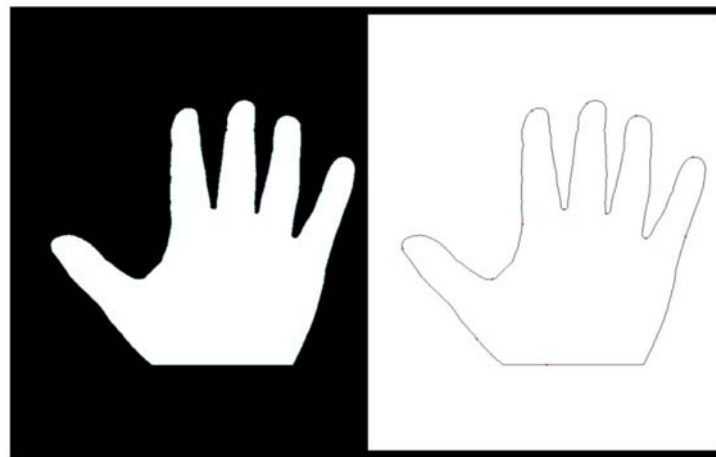
<table border="1"> <tbody> <tr><td>-1</td><td>-1</td><td>-1</td></tr> <tr><td>2</td><td>2</td><td>2</td></tr> <tr><td>-1</td><td>-1</td><td>-1</td></tr> </tbody> </table>	-1	-1	-1	2	2	2	-1	-1	-1	<table border="1"> <tbody> <tr><td>-1</td><td>-1</td><td>2</td></tr> <tr><td>-1</td><td>2</td><td>-1</td></tr> <tr><td>2</td><td>-1</td><td>-1</td></tr> </tbody> </table>	-1	-1	2	-1	2	-1	2	-1	-1	<table border="1"> <tbody> <tr><td>-1</td><td>2</td><td>-1</td></tr> <tr><td>-1</td><td>2</td><td>-1</td></tr> <tr><td>-1</td><td>2</td><td>-1</td></tr> </tbody> </table>	-1	2	-1	-1	2	-1	-1	2	-1	<table border="1"> <tbody> <tr><td>2</td><td>-1</td><td>-1</td></tr> <tr><td>-1</td><td>2</td><td>-1</td></tr> <tr><td>-1</td><td>-1</td><td>2</td></tr> </tbody> </table>	2	-1	-1	-1	2	-1	-1	-1	2
-1	-1	-1																																					
2	2	2																																					
-1	-1	-1																																					
-1	-1	2																																					
-1	2	-1																																					
2	-1	-1																																					
-1	2	-1																																					
-1	2	-1																																					
-1	2	-1																																					
2	-1	-1																																					
-1	2	-1																																					
-1	-1	2																																					
Horizontal	$45^\circ$	Vertical	$-45^\circ$																																				

Figura 13 Máscaras de línea.

**e. Detección de bordes.**

La detección de bordes es el procedimiento empleado más habitualmente para la detección de discontinuidades. Un borde se define como la frontera entre dos regiones con nivel de gris relativamente diferente, suponiendo que las regiones de interés son suficientemente homogéneas de modo que la transición entre dichas regiones se puede determinar empleando exclusivamente las discontinuidades en el nivel de gris.

En la presente tesis se realizó un paso de obtención del borde la palma de la mano a partir de la imagen segmentada del fondo, primero se obtuvo el primer píxel del borde más inferior y se realizó un procedimiento de búsqueda de píxeles del borde tomando como premisa que un borde contiene el contacto de dos píxeles blanco y negro, en la se observa el resultado de obtener el borde de la palma de la mano con esta técnica.



*Figura 14* Obtención del borde de la palma de la mano, a la izquierda la imagen segmentada y a la derecha el borde de la palma.

**f. Limpieza de ruido.**

El término dominio-espacial se refiere al conjunto de puntos que componen una imagen y los métodos en el dominio espacial son procedimientos que operan directamente sobre los píxeles. Las transformaciones de procesamiento de imágenes en el dominio espacial se pueden expresar como:

$$g(x,y) = T\{ f(x,y) \}$$

Donde  $f(x,y)$  es la imagen de entrada,  $g(x,y)$  es la imagen procesada, y  $T$  es el operador definido sobre alguna vecindad del punto  $(x,y)$ . La principal técnica usada para definir una vecindad de  $(x,y)$  es usar una sub-imagen cuadrada centrada en  $(x,y)$ . El centro de la sub-imagen se mueve de punto a punto empezando en la esquina superior izquierda y al aplicar el operador en cada punto obtenemos el valor de la nueva imagen  $g(x,y)$  en cada punto.

Una de las formas de realizar este proceso es mediante máscaras. Básicamente una máscara es una matriz bidimensional pequeña (por ejemplo 3x3) cuyo valor de los elementos son escogidos para detectar una propiedad de la imagen. Pudiéndose formar máscaras de cualquier tamaño o incluso que no sean cuadradas teniendo máscaras rectangulares, circulares, etc.

Suponiendo que una imagen de intensidad constante que contiene puntos aislados cuyas intensidades difieren considerablemente del fondo. En la *Figura 15* se observa una máscara 3x3, los valores que tiene esta máscara gaussiana permiten difuminar bordes utilizando el vecindario del pixel.

-1	-1	-1
-1	8	-1
-1	-1	-1

Figura 15 Máscara Gaussiana 3x3

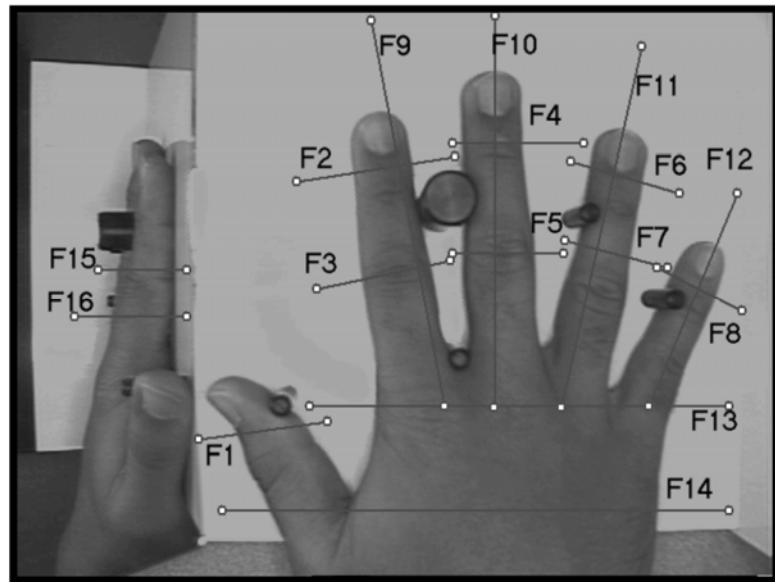
El centro de la máscara se mueve a lo largo de la imagen. En cada posición de la imagen multiplicamos cada punto que está contenido en la imagen por su correspondiente coeficiente de la máscara; es decir, el punto en el centro de la máscara es multiplicado por 8, mientras que sus 8 vecinos se multiplican por el valor de 1. El resultado de estas 9 multiplicaciones se suma. Si todos los puntos bajo la máscara tienen el mismo valor de intensidad (fondo constante), la suma será cero. Sin embargo, si el centro de la máscara se ha colocado en un punto aislado de una intensidad diferente a su fondo, la suma será diferente de cero.

## 1.2 ESTADO DEL ARTE:

En el momento que el reconocimiento de la palma fue una biometría fiable, se generaron múltiples estudios, investigaciones y pruebas siendo las tomadas en esta investigación:

En 1999 con (Anil K. Jain & Arun Ross , Sharath Pankanti, March 22-24, 1999.)Que presentan un prototipo para la verificación de personas por medio de la biometría de su mano, este prototipo en cuestión adquiere las imágenes de la mano izquierda tomadas en una superficie plana con 5 tarugos para asegurar su correcta posición y a su vez cuenta con un espejo para obtener el perfil de la mano. Con esta imagen el prototipo es capaz de obtener 16 distancias características como se muestra en la *Figura 16* que permiten verificar la autenticidad de una persona. Pero su mayor inconveniente fue la de lograr

siempre la misma posición de la mano al momento de su verificación no logrando tener un precisión suficiente.



*Figura 16* Imagen tomada para la autenticación por medio de la biometría de la mano

Es en el 2005 con la publicación de (Vandana Roy and C. V. Jawahar, December 2005) quienes propusieron el uso de análisis discriminario (BDA) no parametrizado, junto con el uso de algoritmo de vecinos cercanos que al igual (Anil K. Jain & Arun Ross , Sharath Pankanti, March 22-24, 1999.) Mantiene el uso de tarugos al momento de capturar la imagen para asegurar una que los dedos se encuentren separados y se la posición sea siempre similar a la tomada para el registro extrayendo los contornos y tomando 24 distancias como se muestra en la *Figura 17*.

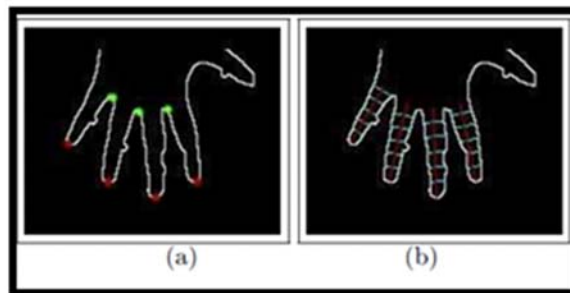


Figura 17 Puntos de inflexión y distancias tomadas

En la conferencia de Data Mining y Data Warehouse se presentó un trabajo de (Karlovec, 12 October, 2010) donde utiliza dos métodos de autenticación (1) basado en una máquina de aprendizaje, (2) funciones de distancia. En la que almacena 48 pares de vectores obtenidos por medio de algoritmos de extracción. Para la validación usa diferentes métodos sobre una muestra de 10 personas dando un resultado de margen de error por cada método. Como se muestra en la Tabla 2, donde determina que el algoritmo de los vecinos más cercanos es el que presenta un menor margen de error pero tiene un costo de tiempo muy elevado.

Tabla 2

Margen de error según diferentes algoritmos de validación de palmas

METHOD CER	(%)
K-Nearest-Neighbor	2,13
AbsoluteDistance (AD)	3,33
Weightedabsolutedistance (WAD)	5,88
Euclidean distance (ED)	5,88
WeightedEuclidean distance (WED)	5,88
D1 distance (D1)	5,88

(elaborador por (Karlovec, 12 October, 2010))

La investigación realizada en el 2012 por (R. Gayathri & P. Ramamoorthy, 2012) quien realizó una investigación para integrar el reconocimiento de la

palma y su huella digital donde remarca que la biometría de la mano por sí sola no presenta una identificación robusta por lo que generan un algoritmo para extraer su textura líneas y apariencias, características que son tomadas desde diferentes enfoques.

El sistema del departamento de defensa (DOD) del gobierno de Estados Unidos fue implementado para el seguimiento de la seguridad nacional, Este sistema incluye una colección de 10 huellas digitales y el menos 5 fotos policiales de distintos ángulos y un hisopo bucal para la recolección del ADN. La Biometría implica la identificación de individuos basado en su fisiología o en su comportamiento, donde se demuestra que las bases del uso la biometría se encuentran muy bien establecidas.



## CAPITULO II: IMPLEMENTACION DEL MECANISMO

La presente tesis propone un algoritmo para el reconocimiento del sujeto dueño de la palma de la mano a autentificar, existen pasos previos al algoritmo de reconocimiento, primero existe un pre-procesamiento de las imágenes de entrada, este pre-procesamiento limpia ruido y elimina las partes con mucho brillo de los contornos de la palma de la mano, luego se realiza un proceso de umbralización, una extracción del contorno de la palma y la obtención de distancias para aplicar técnicas biométricas.

### 2.1. MATERIALES.

Para el desarrollo del presente trabajo se han empleado aparatos e instrumentos de diferente naturaleza:

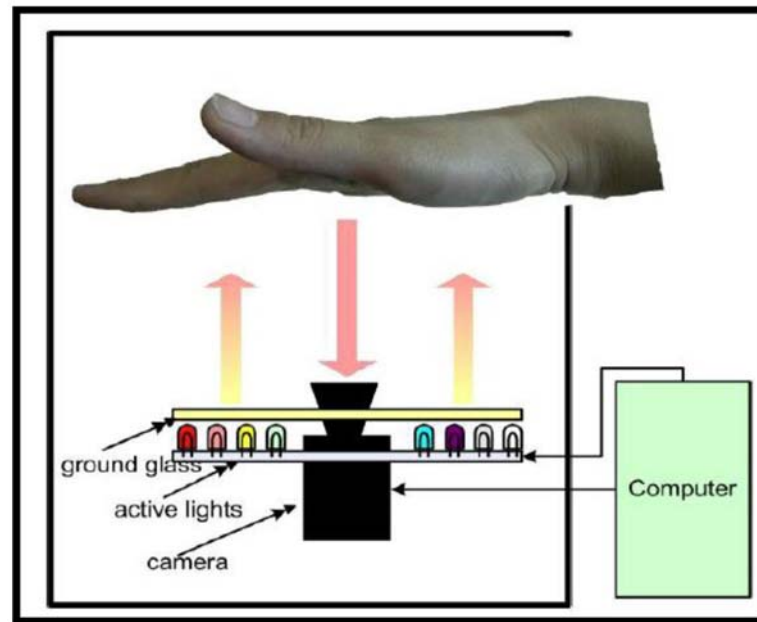
#### 2.1.1 Captura de la imagen

##### ***a. Aparatos de iluminación de longitud de onda determinada.***

De acuerdo con la teoría electromagnética, El rango de ondas hertzianas que van desde el espectro de luz visible hasta el infrarrojo cercano, proporciona un marcado incremento en la penetrabilidad de estas en relación a los objetos sobre los cuales inciden. Para la biometría de la mano, una fuente de luz multi-espectral puede penetrar tejidos cutáneos y subcutáneos a diferentes profundidades en la región palmar y forman imágenes de ambas superficies de la piel, la textura de la piel (epidermis) y la dermis (que incluye las venas de la palma).

Las longitudes ondas de la fuente de iluminación corresponde a seis longitudes de onda: (a) 460nm, (b)630nm, (c)700nm, (d)850nm, (e)940nm y luz blanca, respectivamente, tal como se observa en la

*Figura 18.*



*Figura 18* Dispositivo de Recolección de Imágenes.

El sistema de muestreo admite un cierto grado de tolerancia con respecto a la postura de la mano. Lo que permite lograr una mayor diversidad de muestras y la simulación durante el uso práctico.

#### ***b. Cámara fotográfica digital CCD.***

El dispositivo suministra una iluminación distribuida uniformemente y captura imágenes utilizando una cámara CCD fijada en la parte inferior del dispositivo. El dispositivo posee un circuito de control para ajustar automáticamente los espectros.

Todas las imágenes de las manos son de 8 bits de nivel de gris almacenadas en archivos JPEG.

## 2.1.2 Procesamiento de la imagen

Visual Studio 2010 Profesional C++

Visual Studio 2010 Profesional C#.

Librerías OpenCV, ImageRGB.

## 2.2 OBTENCION DE LA IMAGEN Y SUS PARAMETROS.

### 2.2.1 Almacenamiento.

- La ubicación de la imagen en una cadena de caracteres.
- El valor del umbral de la iluminación en un entero.
- El número de ciclos en un entero.
- Obtiene el valor del Flag que determina si se utilizara o no el algoritmo TBB.

### 2.2.2 Transformación.

La imagen se transforma en una matriz de datos creando un puntero de tipo *IplImage*.

- Al cargar la imagen se le indica que debe ser leída en colores.
- Las propiedades más importantes de este objeto son las que se muestran en **Tabla 3** como también los valores que son iguales para todas las imágenes de la muestra.

Tabla 3.

*Propiedades generales de las imágenes a evaluar*

Propiedad Principales De La Imagen	Valor
Canales	3
Alto	768
Ancho	576
widthStep	1728

*(elaboración Propia)*

- Una vez representada la imagen esta se indexa guardando por cada pixel sus valores de la cantidad de color de rojo, azul y verde.

## 2.3 PREPARACIÓN DE LA IMAGEN.

El siguiente paso es llamado proceso de umbralización, donde se tiene por objetivo identificar plenamente la zona donde se encuentra la palma y la zona del fondo de la imagen de entrada, utilizando para esto el fondo de la caja (que debe de ser de color negro tal como en las imágenes de la base de datos utilizada) o de un fondo oscuro de preferencia uniforme, el proceso de umbralización utiliza un valor umbral respecto al valor del brillo del pixel a analizar en ese momento, recorriendo pixel a pixel toda la imagen y decidiendo que pixel es parte de la mano (colocando el valor del blanco más alto - 255) y que pixel es parte del fondo de la imagen (colocando el valor negro - 0), como se indica en el Algoritmo 1 que Convierte la imagen a una representación a blanco y negro, en la Figura n se puede apreciar la palma de la mano umbralizada.

---

### Algoritmo 1. Umbralización a blanco y negro

---

ENTRADA:

*Matriz*: /\*Array bidimensional de la representación de la imagen\*/

*Umbral\_luminosidad*: Real /\*Parámetro determinante entre blanco y negro\*/

VARIABLES:

*Alto, Ancho*: entero

*Value*: Real /\* Valor calculado de la iluminación\*/

INICIO

LEE *Alto, Ancho*

CALCULAR *Tamaño* = *Alto* \* *Ancho*

PARA *i* = 0 HASTA *Tamaño* CON INCREMENTO +1

/\*Obtiene la luminosidad del pixel\*/

$$Value = \sqrt{matriz[i].blue^2 + matriz[i].red^2 + matriz[i].green^2}$$

/\*Si la luminosidad está por debajo del umbral entonces el pixel se convierte en negro de lo contrario el pixel se convierte en blanco\*/

SI *Value* < *Umbral\_luminosidad* ENTONCES

$$matriz[i].blue = matriz[i].red = matriz[i].green = 0$$

SI NO

$$matriz[i].blue = matriz[i].red = matriz[i].green = 255$$

FIN\_SI

/\*Evalúa si el pixel evaluado es parte del borde izquierdo para colocarlo en negro para poder tenerlo como referencia. ejem.1\*/

SI *i* % *Ancho* = 0 ENTONCES

$$matriz[i].blue = matriz[i].red = matriz[i].green = 0$$

FIN\_SI

FIN\_PARA

FIN

---

*Ejemplo 1:*

Al recorrer una matriz que representa una imagen tal como se muestra en la

*Figura 19* donde:

Ancho = 6  
Alto = 7

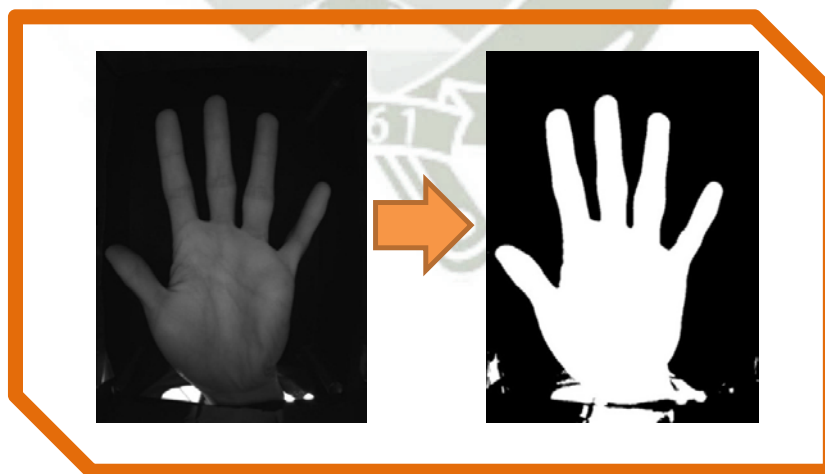
Es decir tiene 48 pixeles enumerados desde 0 como se muestra en la figura.

Al recorrer la matriz para reconocer el borde izquierdo, el pixel que se está evaluando debe ser múltiplo del ancho de la imagen. En este ejemplo los pixeles del borde son las posiciones que son múltiplo de 7 incluyendo el 0.

0	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	32	33	34
35	36	37	38	39	40	41
42	43	44	45	46	47	48

*Figura 19.* Indexación de una imagen representada en una matriz

Al terminar este proceso la imagen que se obtiene se muestra en la *Figura 20*.



*Figura 20.* Proceso de Umbralización

Una vez convertida la imagen de escala de grises a blanco y negro se hace una limpieza de imagen.

Esta limpieza de ruido se hace utilizando el filtro de la mediana, la donde máscara de filtro utilizada es de tamaño 3 x 3, la máscara utilizada en la presente tesis es presentada en la Figura 21.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Figura 21. *Máscara para limpieza de ruido en la imagen*

Esta máscara realiza un proceso de convolución discreto con respecto a la imagen en general, es decir, realiza un recorrido pixel a pixel en la imagen promediando el valor de los valores de cada pixel y su vecindario y colocando en valor en el pixel actual, como se indica en el Algoritmo 2.

Con la utilización de este filtro se elimina ruido del tipo sal y pimienta, siendo esto un prerequisite para el siguiente paso, "La umbralización".

**Algoritmo 2.** Limpieza de la imagen

ENTRADA:

*Matriz:* /\*Arreglo bidimensional de la representación de la imagen\*/  
*Num\_repeticiones:* entero /\*Parámetro determinante entre blanco y negro\*/

VARIABLES:

*Alto, Ancho:* entero  
*w, i, j:* Entero  
*Value:* Entero /\* Valor calculado de la iluminación\*/  
*R[]:* Array

INICIO

LEE *Alto, Ancho* /\*Al calcular el alto se le resta 150 ya que se le corta a la imagen la base donde se encuentra el inicio de la palma es decir la muñeca. Ej. 2\*/

CALCULAR *Tamaño = (Alto -150)\* Ancho*

*i = 0*

*w= Ancho* /\*En el arreglo R se guarda el valor Rojo de todos los pixeles\*/

*R[]= Matriz.Rojo[]*

PARA *j= 1* HASTA *Num\_repeticiones* CON INCREMENTO +1

PARA *i* HASTA *Tamaño* CON INCREMENTO +1 /\*Evaluara el pixel si este no es un pixel de contorno. Ej. 3\*/

SI *i % Ancho > 0* Y *i / Ancho > 0* ENTONCES

$$value = \frac{(R[i-w-1]+R[i-w]+R[i-w+1]+R[i-1]+R[i]+R[i+1]+R[i+w-1]+R[i+w]+R[i+w+1])}{255}$$

/\*Evalúa los pixeles vecinos donde si la mayoría son blancos el pixel es blanco de lo contrario lo pinta negro. Ej. 4\*/

SI *value > 4* ENTONCES

*matriz[i].blue = matriz[i].red = matriz[i].green = 255*

SI NO

*matriz[i].blue = matriz[i].red = matriz[i].green = 0*

FIN\_SI

FIN\_SI

FIN\_PARA

CALCULAR *Tamaño = Alto \* Ancho*

PARA *i* HASTA *Tamaño* CON INCREMENTO +1

*matriz[i].blue = matriz[i].red = matriz[i].green = 0*

FIN\_PARA

FIN\_PARA

FIN

*Ejemplo 2:*

Si la imagen representada en una matriz representada en la Figura 22 con las siguientes dimensiones:

Para no evaluar la última fila es decir la base de la imagen le restamos al alto el número de filas que no se desean evaluar, en este ejemplo no se evaluará la última fila por lo que para hallar el tamaño del recorrido este será:

$$\text{Tamaño} = (\text{Alto} - 1) * \text{Ancho}$$

Entonces:

$$\text{Tamaño} = (6 - 1) * 7 = 35$$

0	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	32	33	34
35	36	37	38	39	40	41

Figura 22. Representación de la imagen en una matriz, sombreando los casilleros a evaluar.

En la Figura 22 se muestran sombreados todos los píxeles a evaluar.

*Ejemplo 3:*

Si tomamos como ejemplo la matriz del ejemplo 2. Y solo se desea evaluar todos los pixeles que no sean contorno superior ni contorno izquierdo la posición tiene que cumplir las siguientes condiciones.

Dónde:

$$\begin{aligned} \text{posicion \% Ancho} &> 0 \\ \text{posicion / Ancho} &> 0 \end{aligned}$$

Solo entran a la condicional los casilleros que no son contornos es decir q su posición al dividirlo su residuo sea mayor a cero (es decir no sea su borde izquierdo) y q también al dividirlo el resultado sea mayor a cero (es decir no sea el borde superior)

Como se muestra en la Figura 23 donde solo se evaluarán los recuadros sombreados de verde reconocidos como bordes los casilleros amarillos

0(R:0,D:0)	1(R:1,D:0)	2(R:2,D:0)	3(R:4,D:0)	4(R:7,D:0)	5(R:7,D:0)	6(R:8,D:0)
7(R:0,D:1)	8(R:1,D:1)	9(R:1,D:1)	10(R:1,D:1)	11(R:1,D:1)	12(R:1,D:1)	13(R:1,D:1)
14(R:0,D:2)	15(R:1,D:1)	16(R:1,D:1)	17(R:1,D:1)	18(R:1,D:1)	19(R:1,D:1)	20(R:1,D:1)
21(R:0,D:3)	22(R:1,D:1)	23(R:1,D:1)	24(R:1,D:1)	25(R:1,D:1)	26(R:1,D:1)	27(R:1,D:1)
28(R:0,D:4)	29(R:1,D:1)	30(R:1,D:1)	31(R:1,D:1)	32(R:1,D:1)	33(R:1,D:1)	34(R:1,D:1)
35	36	37	38	39	40	41

Figura 23. Diferenciación de pixeles que representan bordes (amarillo)

*Ejemplo 4:*

Si continuamos con la matriz de los ejemplos 3 y 4 la evaluación de los pixeles empezaría por el pixel número 8. Para evaluar este pixel se evalúan sus 8 vecinos, como se muestra en la figura 24.

0	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	32	33	34
35	36	37	38	39	40	41

*Figura 24.* Sombreado en rojo los pixeles que evalúa al limpiar el pixel de la posición 8.

Al ser una imagen en blanco y negro sus valores rojo, azul y verde son iguales si su valor es 255 el pixel es blanco o si su valor es 0 el pixel es negro.

La fórmula siguiente determina cuantos pixeles de los 8 más el pixel evaluado son blancos.

*value*

$$= \frac{(R[i - w - 1] + R[i - w] + R[i - w + 1] + R[i - 1] + R[i] + R[i + 1] + R[i + w - 1] + R[i + w] + R[i + w + 1])}{255}$$

R[]: valor del color Rojo.  
 i: posición en el bucle.  
 w: ancho de la imagen.

Dónde:

Si el valor obtenido es menor a 4 entonces los valores de blue, red, y Green los pone a 255( blanco) de lo contrario les da valor a todos los 3 valores son 0(negro). con lo que logra limpiar el ruido de la imagen.

Entonces:

Para el pixel de la posición 8 se vería de la siguiente manera.

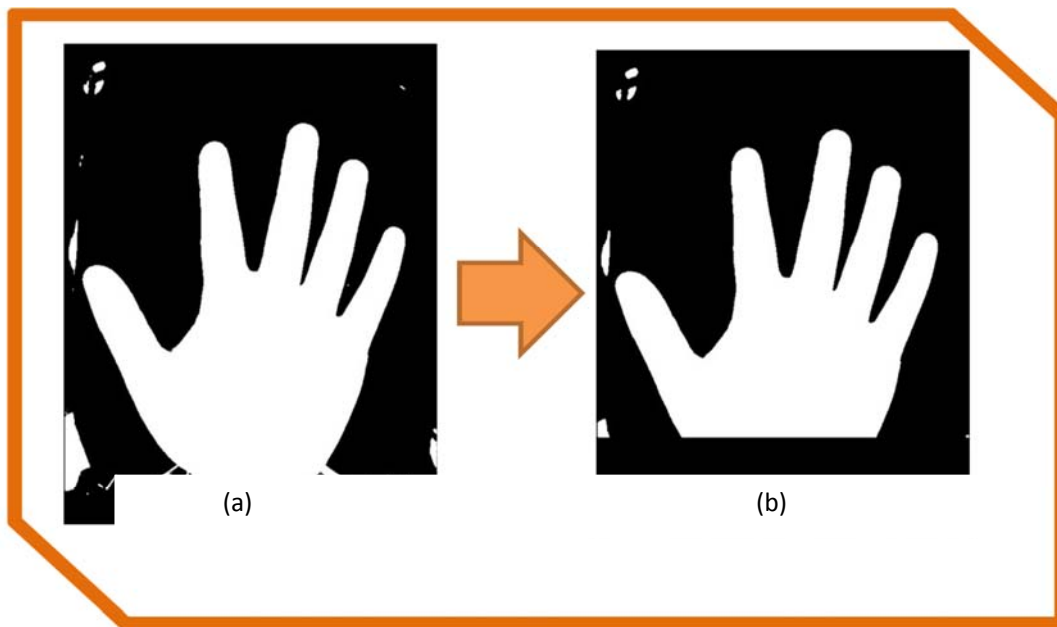
*value*

$$= \frac{(R[8-7-1] + R[8-7] + R[8-7+1] + R[8-1] + R[8] + R[8+1] + R[8+7-1] + R[8+7] + R[8+7+1])}{255}$$

$$value = \frac{(R[0] + R[1] + R[2] + R[7] + R[8] + R[9] + R[14] + R[15] + R[16])}{255}$$

Para el caso de los extremos del borde derecho de la matriz parece no tomar correctamente a sus vecinos pero si recordamos rellenamos de negro todos los pixeles del borde izquierdo.

Al terminar este proceso la imagen que se obtiene se muestra en la *Figura 25*.



*Figura 25.* (a) Imagen tal y como es Tomada (b) Imagen Lista a ser Evaluada.

## 2.4 OBTENCION DE CARACTERÍSTICAS.

Para obtener las características es necesario primero extraer los contornos de la palma a evaluar como se muestra en el Algoritmo 3 para lo cual.

- Crea un objeto de tipo ExtractContour exCo.

- En un vector que almacena objetos de tipo point2D (esta estructura tiene como propiedades dos números enteros x e y) guarda el resultado de la función extract(img) del objeto exCo que recibe como parámetro la ubicación de la imagen.

---

**Algoritmo 3.** Extracción de Contornos
 

---

ENTRADA:

*Matriz: /\*Array bidimensional de la representación de la imagen\*/*

SALIDA:

Vector\_Contornos: vector<Enteros> /\*Array donde se guardan las coordenadas del contorno de la mano \*/

VARIABLES:

Vector: vector<point2D>

Alto, Ancho: entero

x,y,Opcion : Entero

R[]: Array

**INICIO**

LEE Alto, Ancho

CALCULAR  $x = \text{Ancho} / 2$  /\*En x va el valor de la mitad del largo de la imagen. Ejemplo 5\*/

*/\*En y va el alto de la imagen, la resta es debido a que al estar almacenado a un vector este empieza desde la numeración 0 por lo que el alto máximo es la altura menos 1, de lo contrario habría desbordamiento que se encuentra en la mitad del largo de la imagen. Ejemplo 5\*/*

CALCULAR  $y = \text{Alto} - 1$

CALCULAR Opcion = 1

R[] = Matriz.Rojo[]

MIENTRAS  $y > 0$  HACER /\*Al ser la base de la mano negra este sube de fila hasta encontrar la base blanca. Ejemplo 6\*/

SI  $R[y * \text{Ancho} + x] = 255$  ENTONCES

FIN\_MIENTRAS

FIN\_SI

$y = y - 1$

FIN\_MIENTRAS

HACER

INSERTAR ATRAS Vector\_Contornos(x, y)

MIENTRAS Opcion <> -1 /\*Pregunta si se puede mover, Determinando a que posición. Ejemplo 7\*/

SI  $R[(y - 1) * \text{Ancho} + (x + 1)] = 255$  Y Tiene un vecino Negro ENTONCES

Opcion = 0

SI\_NO\_SI  $R[(y) * \text{Ancho} + (x + 1)] = 255$  Y Tiene un vecino Negro ENTONCES

Opcion = 1

SI\_NO\_SI  $R[(y + 1) * \text{Ancho} + (x + 1)] = 255$  Y Tiene un vecino Negro ENTONCES

Opcion = 2

SI\_NO\_SI  $R[(y + 1) * \text{Ancho} + (x)] = 255$  Y Tiene un vecino Negro ENTONCES

Opcion = 3

SI\_NO\_SI  $R[(y + 1) * \text{Ancho} + (x - 1)] = 255$  Y Tiene un vecino Negro ENTONCES

Opcion = 4

SI\_NO\_SI  $R[(y) * \text{Ancho} + (x - 1)] = 255$  Y Tiene un vecino Negro ENTONCES

Opcion = 5

SI\_NO\_SI  $R[(y - 1) * \text{Ancho} + (x - 1)] = 255$  Y Tiene un vecino Negro ENTONCES

Opcion = 6

SI\_NO\_SI  $R[(y - 1) * \text{Ancho} + (x)] = 255$  Y Tiene un vecino Negro ENTONCES

Opcion = 7

```

SI_NO
    Opcion = -1
FIN_SI
SI Opcion <> -1 ENTONCES /*Diferencia el pixel que es borde, Ejemplo 7*/
    SEGÚN Opcion HACER
        CASO, =0
            x = x + 1
            y = y - 1
            R[(y) * Ancho + (x)] = 125

        CASO, =1
            x = x + 1
            R[(y) * Ancho + (x)] = 125

        CASO, =2
            x = x + 1
            y = y + 1
            R[(y) * Ancho + (x)] = 125

        CASO, =3
            y = y + 1
            R[(y) * Ancho + (x)] = 125

        CASO, =4
            x = x - 1
            y = y + 1
            R[(y) * Ancho + (x)] = 125

        CASO, =5
            x = x - 1
            R[(y) * Ancho + (x)] = 125

        CASO, =6
            x = x - 1
            y = y - 1
            R[(y) * Ancho + (x)] = 125

        CASO, =7
            y = y - 1
            R[(y) * Ancho + (x)] = 125

    FIN_SEGUN
    INSERTAR ATRAS Vector_Contornos(x , y)
FIN_SI
FIN_MIENTRAS
FIN

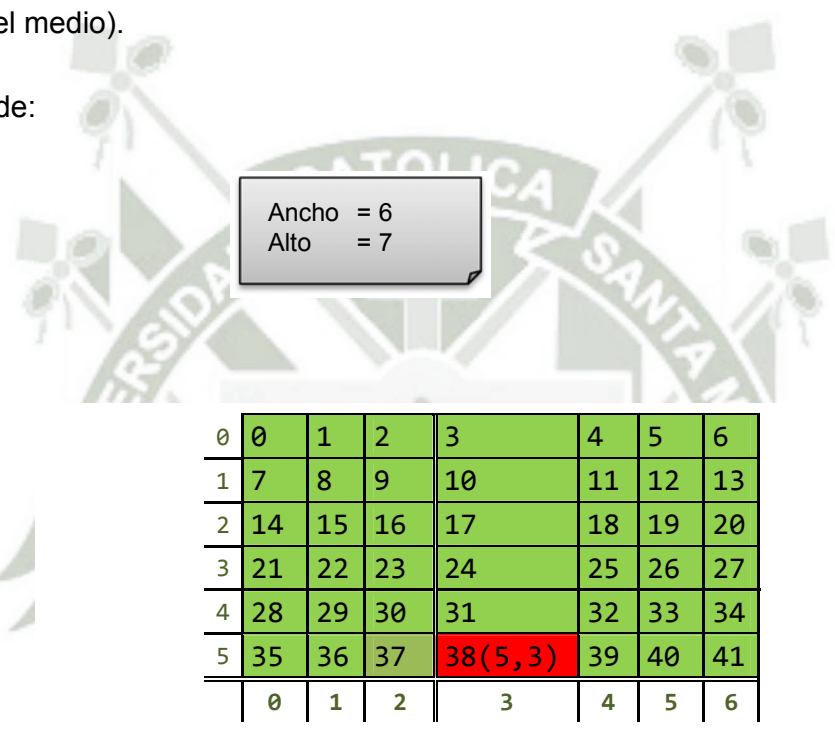
```

*Ejemplo 5:*

Para poder encontrar el pixel del punto medio en la base de la imagen como se muestra en la Figura 26 se toma como coordenadas:

- Coordenada x: El valor del alto de la imagen -1 (La resta se da debido al que el arreglo empieza desde la posición 0)
- Coordenada y: El valor del Ancho de la imagen/2 (Dividido para encontrar el medio).

Dónde:



**Figura 26.** Ubicando el centro de la imagen en la parte inferior de la imagen.

*Ejemplo 6:*

El objetivo de esta parte del algoritmo es encontrar el contorno de la base, recordando que al momento de la limpieza de la imagen esta pinto de negro la base de la mano (muñeca) es necesario reconocer el inicio de la palma. Por este motivo se recorre de abajo hasta arriba hasta encontrar la línea de pixeles blancos q determinan el inicio de la palma.

Continuando con el ejemplo anterior. Se tiene como valor inicial de las coordenadas:

El valor a recorrer es el alto de la imagen en este caso la coordenada y, que va a ir disminuyendo de uno en uno hasta encontrar el borde.

Como vemos en la *Figura 27* el pixel de coordenadas (5,3) es decir (posicion =  $(5 * 7) + 3 = 38$ ) es donde inicia la búsqueda.

0	0	1	2	3	4	5	6
1	7	8	9	10	11	12	13
2	14	15	16	17	18	19	20
3	21	22	23	24	25	26	27
4	28	29	30	31	32	33	34
5	35	36	37	38(5,3)	39	40	41
	0	1	2	3	4	5	6

*Figura 27.* Representación de una imagen maquetada en una matriz, representando las coordenadas del pixel 38.

Según la *Figura 27* el pixel [38] es negro por lo que procede a restar el valor de y. entonces las variables serian:

Estas vienen a dar la Posicion =  $(4 * 7) + 3 = 31$  como se muestra en la *Figura*

x	= 3 /*Ancho*/
y	= 5 /*Alto*/

28.

0	0	1	2	3	4	5	6
1	7	8	9	10	11	12	13
2	14	15	16	17	18	19	20
3	21	22	23	24	25	26	27
4	28	29	30	31(4,3)	32	33	34
5	35	36	37	38	39	40	41
	0	1	2	3	4	5	6

Figura 28. Representación de una imagen maquetada en una matriz, representando las coordenadas del pixel 31.

Según la *Figura 28* este pixel es negro por lo que procede a restar el valor de y. entonces las variables serian:

Estas vienen a dar la Posicion =  $(3 * 7) + 3 = 24$  como se muestra en la *Figura*

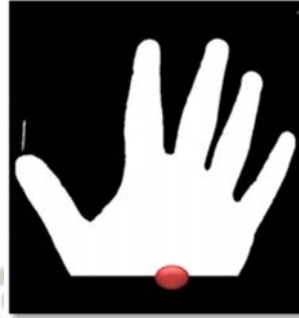
$$y = 3 * \text{Ancho} /$$

0	0	1	2	3	4	5	6
1	7	8	9	10	11	12	13
2	14	15	16	17	18	19	20
3	21	22	23	24(3,3)	25	26	27
4	28	29	30	31	32	33	34
5	35	36	37	38	39	40	41
	0	1	2	3	4	5	6

Figura 29. Representación de una imagen maquetada en una matriz, representando las coordenadas del pixel 24.

29.

Según la *Figura 29* este pixel es Blanco es decir este pixel es contorno por lo que el bucle termina con las coordenadas (3,3). Como dato inicial que viene a ser el punto medio en la parte inferior de la palma como se muestra en la *Figura 30*.



*Figura 30.* Ubicación de punto medio en la parte inferior de la palma.



Ejemplo 7:

0	0	1	2	3	4	5	6
1	7	8	9	10	11	12	13
2	14	15	16	17	18	19	20
3	21	22	23	24(3,3)	25	26	27
4	28	29	30	31	32	33	34
5	35	36	37	38	39	40	41
	0	1	2	3	4	5	6

Figura 31. Evaluación del pixel 24 sombreando todos los pixeles relacionados a este.

Una vez iniciado, e ingresado el primer contorno se evalúa todos pixeles que rodean este. Seguimos con el ejemplo anterior q vemos en la *Figura 31* que se inicia en la posición 24 con letras verdes los pixeles a evaluar.

Los pixeles son escogidos según la siguiente formula (recordando que los valores de x y y son 3y 3 correlativamente):

$$\begin{aligned}
 R[(y - 1) * Ancho + (x + 1)] &= R[(3 - 1) * 7 + (3 + 1)] &&= R[18] \\
 R[(y) * Ancho + (x + 1)] &= R[(3) * 7 + (3 + 1)] &&= R[25] \\
 R[(y + 1) * Ancho + (x + 1)] &= R[(3 + 1) * 7 + (3 + 1)] &&= R[32] \\
 R[(y + 1) * Ancho + (x)] &= R[(3 + 1) * 7 + (3)] &&= R[31] \\
 R[(y + 1) * Ancho + (x - 1)] &= R[(3 + 1) * 7 + (3 - 1)] &&= R[30] \\
 R[(y) * Ancho + (x - 1)] &= R[(3) * 7 + (3 - 1)] &&= R[23] \\
 R[(y - 1) * Ancho + (x - 1)] &= R[(3 - 1) * 7 + (3 - 1)] &&= R[16] \\
 R[(y - 1) * Ancho + (x)] &= R[(3 - 1) * 7 + (3)] &&= R[17]
 \end{aligned}$$

Como se puede ver la evaluación de los pixeles que bordean el pixeles en sentido horario. Como se ve en la *Figura 32*.

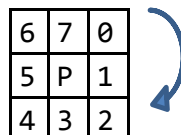


Figura 32. Representación del orden de evaluación en sentido horario de los pixeles a su alrededor.

Una vez encontrado el primer pixel blanco se pregunta si este tiene un pixel negro en sus adyacentes verticales u horizontales como se muestra en la *Figura 33*. Que según el ejemplo el primero a evaluar seria el pixel numero 18 al que se registra como pixel blanco, por lo que se evalúa los vecinos horizontales y verticales de este que vienen a ser los pixeles 17 y 19 como horizontales y los pixeles 11 y 25 como los verticales.(pixeles con letras rojas).

0	0	1	2	3	4	5	6
1	7	8	9	10	11	12	13
2	14	15	16	17	18	19	20
3	21	22	23	24	25	26	27
4	28	29	30	31	32	33	34
5	35	36	37	38	39	40	41
	0	1	2	3	4	5	6

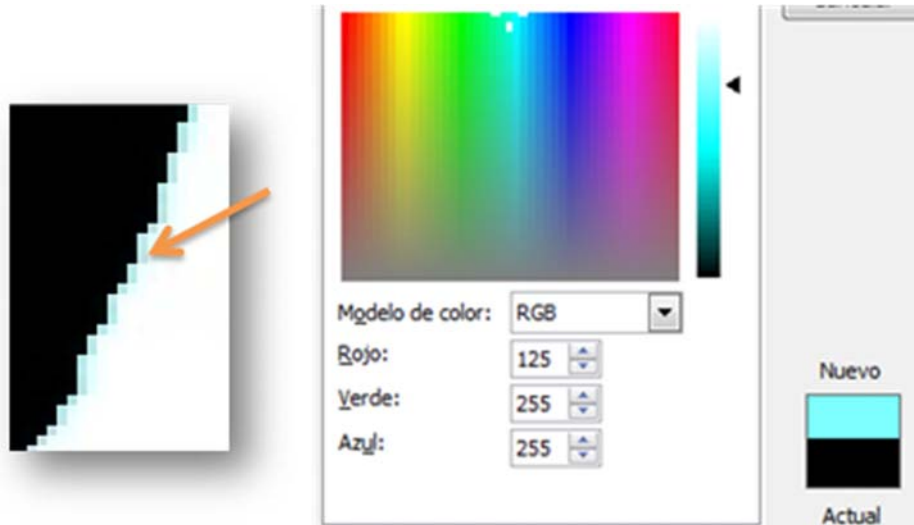
*Figura 33. Sombreado de los pixeles a ser evaluados.*

En el caso del ejemplo, ninguno de sus vecinos horizontales y verticales del pixel #18 son negros por lo que pasa a evaluar al siguiente pixel según lo estipulado que viene a ser el pixel #25, que cumple la primera condición de ser blanco para la segunda condición evaluar sus vecinos horizontales #24 y #26 así como también sus vecinos verticales #18 y #32, siendo los pixel #26 y #32 negros.

Una vez encontrado el siguiente pixel del contorno las coordenadas x e y apuntan a este pixel y este es pintado de otro color como se muestra en la *Figura 33* para reconocerlo como ya explorado y evitar entrar a un bucle infinito, para luego evaluar los contornos de este pixel así hasta lograr identificar todo el contorno de la mano.

Esta evaluación va en sentido anti-horario. Al no encontrar los contornos negros alado de los pixeles blancos porque ya todos los contornos son celestes retorna -1 terminando así la evaluación.

Como se muestra en la *Figura 34* el contorno de la palma toma un color celeste.



*Figura 34.* Acercamiento del contorno de palma.

Una vez determinado el contorno de la palma se procede a reconocer los Puntos de Inflexión.

- Los puntos de inflexión de la palma de la mano son los representados en la *Figura 35*.

Que son registrados según el siguiente algoritmo.

**Algoritmo 4.** Reconocimiento de los Puntos de inflexión.

ENTRADA:

*VContor*: vector <point2D> /\*Array donde se guardan las coordenadas del contorno de la mano \*/

SALIDA:

*VInflex*: vector <Enteros> /\*Array que almacena las coordenadas de los puntos de inflexión\*/

VARIABLES:

*Delta*: Entero /\*Valor del salto de pixel a pixel\*/*l*, *Previo*: Entero*Current*: Flotante*R*[]): Array*Flag*: Booleano /\*Identifica si la inflexión va de subida o de bajada\*/**INICIO***Delta* = 10*Previo* = 0INSERTAR ATRAS *VInflex*(0)PARA *i* = 1 HASTA *VContor.Tamaño()* CON INCREMENTO +*Delta*

/\*Hace una validación que determina si la imagen tomada no tiene la característica regulares\*/

SI *VInflex*[0] ≥ *VContor.Tamaño()* ENTOCES*i* = *VContor.Tamaño()*

SI\_NO

CALCULAR

*Current* =

$$\sqrt{(VContor[i].x - VContor[VInflex[0]].x)^2 + (VContor[i].y - VContor[VInflex[0]].y)^2}$$

/\*El contorno va de subida por lo que se busca el punto de inflexión que cambia la dirección hacia abajo\*/.

/\*TRUE dirección de la curva va hacia arriba\*/ /\*FALSE dirección de la curva va hacia abajo\*/

SI *Flag* = true ENTOCES

/\*Si el cálculo hallado es menor al anterior es por q se encontró el punto de inflexión y la curva empieza a ir hacia abajo\*/.

SI *Current* - *Previo* < 0 ENTOCESINSERTAR ATRAS *VInflex*(*i*)*Flag* = false

FIN\_SI

SI\_NO

/\*Si el cálculo hallado es mayor al anterior entonces este es el punto de inflexión que cambia la dirección hacia arriba\*/.

SI *Current* - *Previo* > 0 ENTOCESINSERTAR ATRAS *VInflex*(*i*)*Flag* = True

FIN\_SI

FIN\_SI

*Previo* = *Current*

FIN\_SI

FIN\_PARA /\*Al terminar el bucle se debe haber registrado 10 puntos inflexión ejemplo 8\*/

SI *VInflex.Tamaño()* < 10 ENTOCESLIMPIAR *VInflex*[ ]

SI\_NO

SI *VInflex.Tamaño()* > 10 ENTOCES*VInflex.Tamaño* = 100

SI\_NO

/\*Una vez registrados los diez puntos se agregan 3 puntos en función espejo con respecto a los ya ingresados, Ejemplo 9\*/

RECORRER 1 POSICIONES DESDE -> *VInflex*(1)INSERTAR *VeInflex*[1] = *VInflex*[1] - (*VInflex*[2] - *VInflex*[1])RECORRER 1 POSICIONES DESDE -> *VInflex*(9)INSERTAR *VeInflex*[9] = *VInflex*[8] + (*VInflex*[8] - *VInflex*[7])

```
RECORRER 1 POSICIONES DESDE -> VInflex(12)  
INSERTAR VeInflex[11] = VInflex[11] + (VInflex[11] - VInflex[10])  
FIN_SI  
FIN_SI  
RETORNAR VeInflex  
FIN
```

---

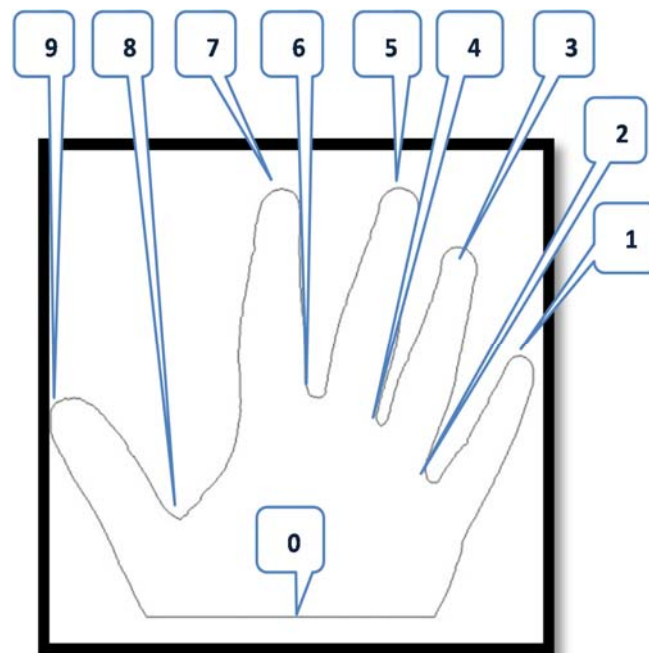


*Ejemplo 8:*

En el vector de contornos se guardan coordenadas de todo el borde de la mano, en el vector de puntos de inflexión se guardan posiciones del vector de contornos.

En *Figura 35* se muestra el orden como se hallan los 10 primeros puntos de inflexión (recordando que los vectores empiezan desde la posición 0).

Para recorrer el contorno no es necesario recorrer todo el contorno se recorre



*Figura 35.* Puntos de Inflexión de la palma izquierda.

de 10 en 10 posiciones.

El primer punto de inflexión es la primera coordenada de la *Figura 36* los valores de los siguientes coordenadas previos al punto de inflexión #1 se van incrementando es decir siempre son mayores al anterior, y cuando el siguiente valor es menor al anterior este reconoce como punto de inflexión, una vez encontrado el tope de la curva es decir la máxima extensión del dedo. Al

encontrar el punto más alto los valores del contorno empiezan a decrecer; cuando el valor anterior deja de ser menor al nuevo se registra como punto de inflexión, que viene a ser la base del dedo. Y esta lógica se repite encontrando las 5 extensiones de los dedos así como también las 4 comisuras entre los dedos más la coordenada cero.

En la Tabla 4 se puede ver un ejemplo de los 10 puntos de inflexión.

Tabla 4.

*Ejemplo - Arreglo de posiciones donde se encuentran los puntos de inflexión*

Pos.	Valor
Begin	0
1	451
2	611
3	881
4	1081
5	1361
6	1601
7	1861
8	2201
9	2421

*(elaboración propia)*

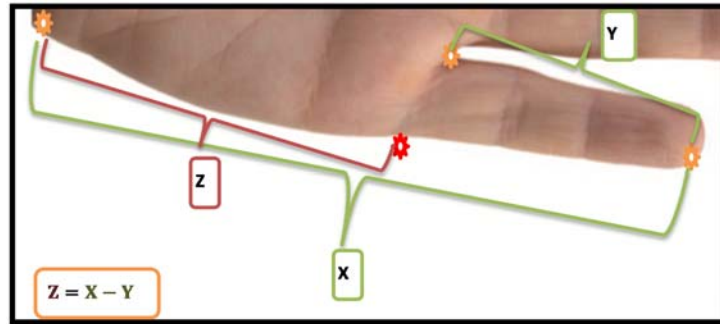
*Ejemplo 9:*

Existen 3 puntos de inflexión denominados puntos de “inflexión falsos” ya que no son generados al cambiar de dirección la curvatura pero si son importantes para determinar la extensión de los dedos meñique, índice y pulgar. Como se muestra en la *Figura 36*.



*Figura 36.* Puntos de Inflexión Falsos

Para encontrar la posición se toma su valor por formula que se explica en la *Figura 37* donde se halla el punto de inflexión falso. Este mismo procedimiento se utiliza para el dedo índice y pulgar.



*Figura 37.* Representación gráfica del cálculo del primer punto de inflexión.

Esta lógica explicada desde el vector del ejemplo 8, para el dedo meñique seria:

$$VectorInflexion[1] - (VectorInflexion[2] - VectorInflexion[1])$$

$$451 - (611 - 451) = 291$$

Una vez determinada la posición del primer punto de inflexión falso este es ubicado en la posición 1 por lo que los demás puntos corren una posición como se ve en la *Figura 38*.

Pos.	Valor		Pos.	Valor
Begin	0		Begin	0
1	451		1	291
2	611		2	451
3	881		3	611
4	1081		4	881
5	1361		5	1081
6	1601		6	1361
7	1861		7	1601
8	2201		8	1861
9	2421		9	2201
			10	2421

*Figura 38.* Inserción del primer punto de inflexion falso.

Para hallar el punto de inflexión Falso del dedo índice sería:

$$\text{VectorInflexion}[8] + (\text{VectorInflexion}[8] - \text{VectorInflexion}[7])$$

$$1861 + (1861 - 1601) = 2121$$

Este nuevo punto es insertado en la posición 9 como se ve en *Figura 39*.

Pos.	Valor		Pos.	Valor
Begin	0		Begin	0
1	291		1	291
2	451		2	451
3	611		3	611
4	881		4	881
5	1081		5	1081
6	1361		6	1361
7	1601		7	1601
8	1861		8	1861
9	2201		9	2121
10	2421		10	2201
			11	2421

*Figura 39.* Inserción del segundo punto de inflexión falso (dedo índice).

Y para hallar el punto de inflexión Falso del dedo pulgar sería:

$$\text{VectorInflexion}[11] + (\text{VectorInflexion}[11] - \text{VectorInflexion}[10])$$

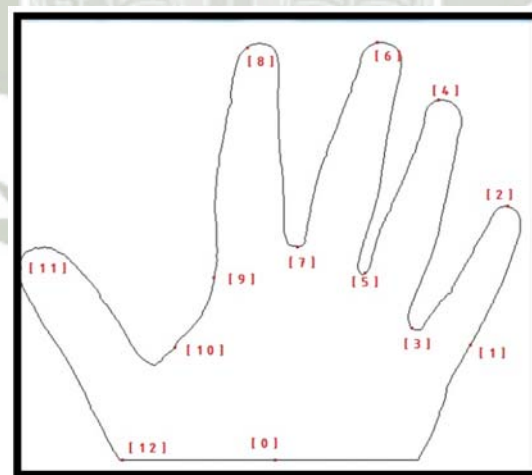
$$2421 + (2421 - 2201) = 2641$$

El tercer y último punto de inflexión falso se agrega al final del vector teniendo ubicados los 12 puntos de inflexión que se muestra en la *Figura 40* siendo esto necesarios para poder calcular los centros de cada dedo.

Pos.	Valor		Pos.	Valor
Begin	0		Begin	0
1	291		1	291
2	451		2	451
3	611		3	611
4	881		4	881
5	1081		5	1081
6	1361		6	1361
7	1601		7	1601
8	1861		8	1861
9	2121		9	2121
10	2201		10	2201
11	2421		11	2421
			12	2641

*Figura 40.* Inserción del tercer punto de inflexión (dedo pulgar).

En la *Figura 41* se muestran ubicados los 12 puntos de inflexión.



*Figura 41.* Representación gráfica de donde se encuentran los puntos de inflexión.

Una vez determinados los 13 puntos de inflexión se procede a calcular los centros de cada dedo. Centros que son calculados según el Algoritmo 5.



**Algoritmo 5.** Ubicación de los centros de cada dedo y de la palma.

ENTRADA:

*VInflex*: vector <Enteros> /\*Array que almacena las coordenadas de los puntos de inflexión\*/*VContor*: vector <point2D> /\*Array donde se guardan las coordenadas del contorno de la mano \*/

SALIDA:

*VCentros*: vector <point2D> /\*Array donde se guardan las coordenadas del contorno de la mano \*/

VARIABLES:

*p1, p2, p3, p4, p5*: point2D /\*Cinco centros\*/*Index*: Entero*Current*: Flotante*R[]*: Array*Flag*: Booleano /\*Identifica si la inflexión va de subida o de bajada\*/**INICIO***Index* = *VInflex*[0]

/\*Bucle que determina el primer centro, que viene a ser el dedo meñique\*/

PARA *p1.x* = 0, *p1.y* = 0 HASTA *Index* < *VInflex*[3] CON INCREMENTO +*Index*CALCULAR *p1.x* = *p1.x* + *VContor*[*index*].*x*CALCULAR *p1.y* = *p1.y* + *VContor*[*index*].*y*

FIN\_PARA

CALCULAR  $p1.x = \frac{p1.x}{VInflex[3] - VInflex[1]}$ CALCULAR  $p1.y = \frac{p1.y}{VInflex[3] - VInflex[1]}$ *VCentros*[0] = *p1*

/\*Bucle que determina el segundo centro, que viene a ser el dedo anular\*/

PARA *p2.x* = 0, *p2.y* = 0 HASTA *Index* < *VInflex*[5] CON INCREMENTO +*Index*CALCULAR *p2.x* = *p2.x* + *VContor*[*index*].*x*CALCULAR *p2.y* = *p2.y* + *VContor*[*index*].*y*

FIN\_PARA

CALCULAR  $p2.x = \frac{p2.x}{VInflex[5] - VInflex[3]}$ CALCULAR  $p2.y = \frac{p2.y}{VInflex[5] - VInflex[3]}$ *VCentros*[1] = *p2*

/\*Bucle que determina el tercer centro, que viene a ser el dedo medio\*/

PARA *p3.x* = 0, *p3.y* = 0 HASTA *Index* < *VInflex*[7] CON INCREMENTO +*Index*CALCULAR *p3.x* = *p3.x* + *VContor*[*index*].*x*CALCULAR *p3.y* = *p3.y* + *VContor*[*index*].*y*

FIN\_PARA

CALCULAR  $p3.x = \frac{p3.x}{VInflex[7] - VInflex[5]}$ CALCULAR  $p3.y = \frac{p3.y}{VInflex[7] - VInflex[5]}$ *VCentros*[2] = *p3*

/\*Bucle que determina el cuarto centro, que viene a ser el dedo índice\*/

PARA *p4.x* = 0, *p4.y* = 0 HASTA *Index* < *VInflex*[9] CON INCREMENTO +*Index*CALCULAR *p4.x* = *p4.x* + *VContor*[*index*].*x*CALCULAR *p4.y* = *p4.y* + *VContor*[*index*].*y*

FIN\_PARA

CALCULAR  $p4.x = \frac{p4.x}{VInflex[9] - VInflex[7]}$ CALCULAR  $p4.y = \frac{p4.y}{VInflex[9] - VInflex[7]}$ *VCentros*[3] = *p4*

/\*Bucle que determina el quinto centro, que viene a ser el dedo Pulgar\*/

PARA *p5.x* = 0, *p5.y* = 0 HASTA *Index* < *VInflex*[12] CON INCREMENTO +*Index*CALCULAR *p5.x* = *p5.x* + *VContor*[*index*].*x*

```
CALCULAR  $p5.y = p5.y + VContor[index].y$   
FIN_PARA  
CALCULAR  $p5.x = p5.x / (VInflex[12] - VInflex[10])$   
CALCULAR  $p5.y = p5.y / (VInflex[12] - VInflex[10])$   
 $VCentros[4] = p5$   
RETORNAR  $VCentros$ 
```

FIN

---



*Ejemplo 10:*

Para hallar los cinco centros se crea cinco bucles donde que son distribuidos de la siguiente manera.

Centro 1.

Recorre todos los contornos comprendidos entre el primer punto de inflexión (excluyendo la posición 0) y el tercer punto de inflexión sin incluir este último. Según el ejemplo visto recorrerá los contornos incluidos entre el rango 291 y 610 como se muestran sombreados en la *Figura 42*.

Pos.	Valor
Begin	0
1	291
2	451
3	611
4	881
5	1081
6	1361
7	1601
8	1861
9	2121
10	2201
11	2421
12	2641

*Figura 42.* Posiciones tomadas en cuenta para hallar el centro del dedo meñique

Sumará su posición del vector x como del vector y. una vez obtenida la suma de posiciones las dividirá respectivamente entre la resta de la tercera y primera posición(es decir el número total de contornos evaluados) para poder hallar así un promedio.

En el caso del ejemplo sería  $611 - 291 = 320$ .

Hallando así el centro del primer dedo (meñique).

### Centro 2.

Recorre todos los contornos comprendidos entre el tercer punto de inflexión y el quinto punto de inflexión sin incluir este último.

Según el ejemplo visto recorrerá los contornos incluidos entre el rango 611 y 1080 como se muestran sombreados en la *Figura 43*.

Pos.	Valor
Begin	0
1	291
2	451
3	611
4	881
5	1081
6	1361
7	1601
8	1861
9	2121
10	2201
11	2421
12	2641

*Figura 43.* Posiciones tomadas en cuenta para hallar el centro del dedo anular.

Centro 3.

Recorre todos los contornos comprendidos entre el quinto punto de inflexión y el séptimo punto de inflexión sin incluir este último.

Según el ejemplo visto recorrerá los contornos incluidos entre el rango 1081 y 1600 como se muestran sombreados en la *Figura 44*.

Sumará su posición del vector x como del vector y. una vez obtenida la suma

Pos.	Valor
Begin	0
1	291
2	451
3	611
4	881
5	1081
6	1361
7	1601
8	1861
9	2121
10	2201
11	2421
12	2641

*Figura 44.* Posiciones tomadas en cuenta para hallar el centro del dedo medio.

de posiciones las dividirá respectivamente entre la resta de la quinta y séptima posición (es decir el número total de contornos evaluados) para poder hallar así un promedio.

En el caso del ejemplo sería  $1601 - 1081 = 520$

Hallando así el centro del tercer dedo (medio).

Centro 4.

Recorre todos los contornos comprendidos entre el séptimo punto de inflexión y el noveno punto de inflexión sin incluir este último.

Según el ejemplo visto recorrerá los contornos incluidos entre el rango 1601 y 2120 como se muestran sombreados en la *Figura 45*.

<i>Pos.</i>	<i>Valor</i>
Begin	0
1	291
2	451
3	611
4	881
5	1081
6	1361
7	1601
8	1861
9	2121
10	2201
11	2421
12	2641

*Figura 45.* Posiciones tomadas en cuenta para hallar el centro del dedo índice.

Sumará su posición del vector x como del vector y. una vez obtenida la suma de posiciones las dividirá respectivamente entre la resta de la séptima y novena posición (es decir el número total de contornos evaluados) para poder hallar así un promedio.

En el caso del ejemplo sería  $2121 - 1601 = 520$

Hallando así el centro del cuarto dedo (índice).

### Centro 5.

Recorre todos los contornos comprendidos entre el décimo punto de inflexión y el doceavo punto de inflexión sin incluir este último.

Según el ejemplo visto recorrerá los contornos incluidos entre el rango 2201 y 2640 como se muestran sombreados en la *Figura 46*.

<i>Pos.</i>	<i>Valor</i>
Begin	0
1	291
2	451
3	611
4	881
5	1081
6	1361
7	1601
8	1861
9	2121
10	2201
11	2421
12	2641

*Figura 46.* Posiciones tomadas en cuenta para hallar el centro del dedo pulgar.

Sumará su posición del vector x como del vector y. una vez obtenida la suma de posiciones las dividirá respectivamente entre la resta de la décima y doceava posición (es decir el número total de contornos evaluados) para poder hallar así un promedio.

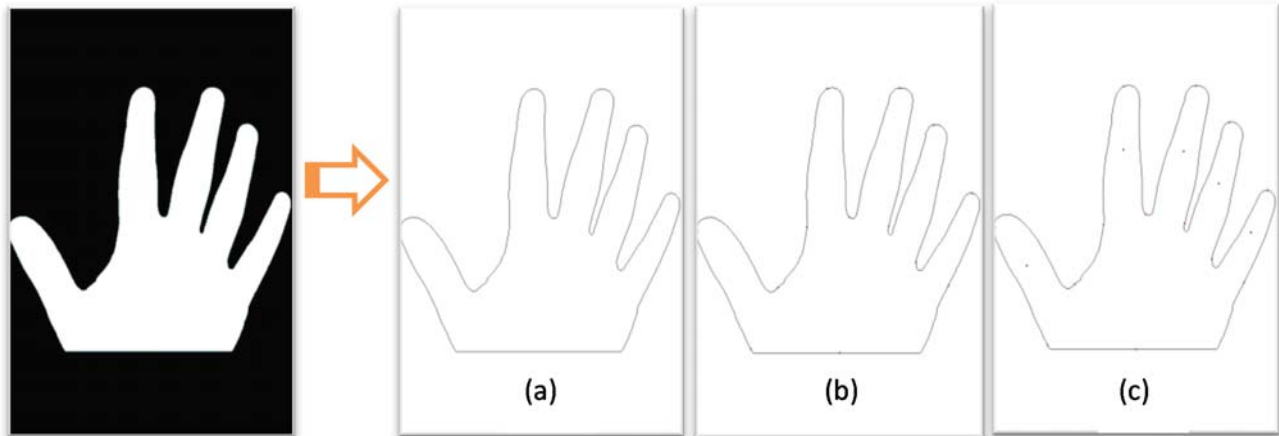
En el caso del ejemplo sería  $2641 - 2201 = 440$

Hallando así el centro del quinto dedo (pulgarcito).

Con los centros ubicados en un vector de nombre centers, y los puntos de inflexión ubicados es posible generar los las distancias que diferencien una mano de otra.

Previo a la generación de distancias se realizan 3 procesos que facilitan la obtención de distancias (Como se muestra en la *Figura 47*):

- Limpieza del fondo negro. *Figura 47 (a)*
- Pintar los 13 puntos de inflexión. *Figura 47 (b)*
- Pintar los centros de los 5 dedos. *Figura 47(c)*



*Figura 47.* Generación gráfica de los puntos de inflexión y los centros.

Para hallar estas distancias utilizamos el siguiente algoritmo.



**Algoritmo 6.** Generación de características

ENTRADA:

*VInflex*: vector <Enteros> /\*Array que almacena las coordenadas de los puntos de inflexión\*/

*VContor*: vector <point2D> /\*Array donde se guardan las coordenadas del contorno de la mano \*/

*VCentros*: vector <point2D> /\*Array donde se guardan las coordenadas del contorno de la mano \*/

SALIDA:

Features: vector&lt;float&gt;

VARIABLES:

*Alto, Ancho*: entero

*i, Cuadrante1, Cuadrante2, CentroX, CentroY*: Entero

*Distancia*: Flotante

*R[]*: Array

*Flag*: Booleano /\*Identifica si la inflexión va de subida o de bajada\*/

INICIO

 LEE *Alto, Ancho*
*CentroX = CentroY = 0* /\*Bucle que recorre los cuatro primeros dedos \*/

 PARA *i = 0* HASTA *i < 4* CON INCREMENTO +*i*

/\*Distancia entre el centro del dedo y el primer punto de inflexión que limita el dedo. Ejemplo #11\*/

$$Distancia = (VCentros[i].x - VContor[VInflex[(i * 2) + 1]].x)^2 + (VCentros[i].y - VContor[VInflex[(i * 2) + 1]].y)^2$$

 INSERTAR ATRAS *Features(Distancia)*

/\*Distancia entre el centro del dedo y el segundo punto de inflexión que limita el dedo. Ejemplo #11 \*/

$$Distancia = (VCentros[i].x - VContor[VInflex[(i * 2) + 2]].x)^2 + (VCentros[i].y - VContor[VInflex[(i * 2) + 2]].y)^2$$

 INSERTAR ATRAS *Features(Distancia)*

/\*Distancia entre el centro del dedo y el tercer punto de inflexión que limita el dedo. Ejemplo #11 \*/

$$Distancia = (VCentros[i].x - VContor[VInflex[(i * 2) + 3]].x)^2 + (VCentros[i].y - VContor[VInflex[(i * 2) + 3]].y)^2$$

 INSERTAR ATRAS *Features(Distancia)*

/\*Determinamos el número de vectores que se va a generar en el primer cuadrante del dedo a evaluar. Ejemplo #12\*/

$$Cuadrante1 = (VInflex[(i * 2) + 2] - VInflex[(i * 2) + 1])/8$$

 PARA *j = 1* HASTA *j < 8* CON INCREMENTO +*j*

$$Distancia = (VCentros[i].x - VContor[VInflex[(i * 2) + 1] + (Cuadrante1 * j)].x)^2 + (VCentros[i].y - VContor[VInflex[(i * 2) + 1] + (Cuadrante1 * j)].y)^2$$

 INSERTAR ATRAS *Features(Distancia)*

FIN\_PARA

/\*Determinamos el número de vectores que se va a generar en el primer cuadrante del dedo a evaluar. Ejemplo #12\*/

$$Cuadrante2 = (VInflex[(i * 2) + 3] - VInflex[(i * 2) + 2])/8$$

 PARA *j = 1* HASTA *j < 8* CON INCREMENTO +*j*

$$Distancia = (VCentros[i].x - VContor[VInflex[(i * 2) + 2] + (Cuadrante2 * j)].x)^2 + (VCentros[i].y - VContor[VInflex[(i * 2) + 2] + (Cuadrante2 * j)].y)^2$$

 INSERTAR ATRAS *Features(Distancia)*

FIN\_PARA

FIN\_PARA

/\*Distancia entre el centro del dedo pulgar y los puntos de inflexión que limita el dedo. \*/

$$Distancia = (VCentros[4].x - VContor[VInflex[(4 * 2) + 2]].x)^2 + (VCentros[4].y - VContor[VInflex[(4 * 2) + 2]].y)^2$$

 INSERTAR ATRAS *Features(Distancia)*

$$Distancia = (VCentros[4].x - VContor[VInflex[(4 * 2) + 3]].x)^2 + (VCentros[4].y - VContor[VInflex[(4 * 2) + 3]].y)^2$$

 INSERTAR ATRAS *Features(Distancia)*

```

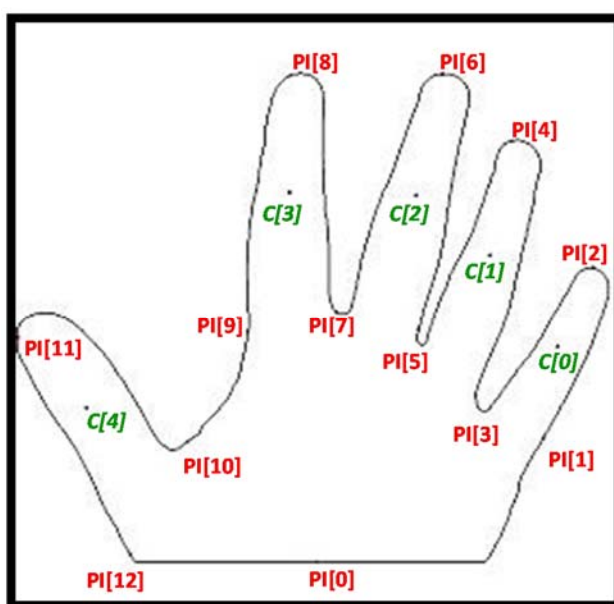
Distancia = (VCentros[4].x - VContor[VInflex[(4 * 2) + 4]].x)2
            + (VCentros[4].y - VContor[VInflex[(4 * 2) + 4]].y)2
INSERTAR ATRAS Features(Distancia)
/*Determinamos el número de vectores que se va a generar en el primer cuadrante del dedo a evaluar. Ejemplo #12*/
Cuadrante1 = (VInflex[(4 * 2) + 3] - VInflex[(4 * 2) + 2])/8
PARA j = 1 HASTA j < 8 CON INCREMENTO +j
    Distancia = (VCentros[4].x - VContor[VInflex[(4 * 2) + 2] + (Cuadrante1 * j)].x)2
                + (VCentros[4].y - VContor[VInflex[(4 * 2) + 2] + (Cuadrante1 * j)].y)2
    INSERTAR ATRAS Features(Distancia)
FIN_PARA
/*Determinamos el número de vectores que se va a generar en el primer cuadrante del dedo a evaluar. Ejemplo #12*/
Cuadrante2 = (VInflex[(i * 2) + 4] - VInflex[(i * 2) + 3])/8
PARA j = 1 HASTA j < 8 CON INCREMENTO +j
    Distancia = (VCentros[4].x - VContor[VInflex[(4 * 2) + 3] + (Cuadrante2 * j)].x)2
                + (VCentros[4].y - VContor[VInflex[(4 * 2) + 3] + (Cuadrante2 * j)].y)2
    INSERTAR ATRAS Features(Distancia)
FIN_PARA
/*Bucle que suma las coordenadas x e y respectivamente de puntos de inflexión que representan la base de los dedos es decir #1,
#3, #5, #7, #9, #10, #12. Ejemplo #13*/
PARA i = 0 HASTA i < VInflex.Tamaño() CON INCREMENTO +i
    SI (i%2 ≠ 0 Y i ≠ 11) O i = 10 O i = 12 ENTONCES
        CentroX = CentroX + VContor[VInflex[i]].x
        CentroY = CentroY + VContor[VInflex[i]].y
    FIN_SI
FIN_PARA
/*Halla el promedio de la sumatoria para determinar el centro de la palma. Ejemplo #13*/
CentroX = CentroX/7
CentroY = CentroY/7
/* Calculamos la distancia entre el centro de la palma y los puntos de inflexión que representan la base de los dedos es decir #1,
#3, #5, #7, #9, #10, #12. Ejemplo #13*/
PARA i = 0 HASTA i < VInflex.Tamaño() CON INCREMENTO +i
    SI (i%2 ≠ 0 Y i ≠ 11) O i = 10 O i = 12 ENTONCES
        Distancia = (CentroX - VContor[VInflex[i]].x)2 + (CentroY - VContor[VInflex[i]].y)2
    INSERTAR ATRAS Features(Distancia)
    FIN_SI
FIN_PARA
/* Calculamos la Ancho del dedo en la base de los cinco dedos. Ejemplo #14*/
INSERTAR ATRAS Features
{(VContor[VInflex[1]].x - VContor[VInflex[3]].x)2 + (VContor[VInflex[1]].y - VContor[VInflex[3]].y)2}
INSERTAR ATRAS Features
{(VContor[VInflex[3]].x - VContor[VInflex[5]].x)2 + (VContor[VInflex[3]].y - VContor[VInflex[5]].y)2}
INSERTAR ATRAS Features
{(VContor[VInflex[5]].x - VContor[VInflex[7]].x)2 + (VContor[VInflex[5]].y - VContor[VInflex[7]].y)2}
INSERTAR ATRAS Features
{(VContor[VInflex[7]].x - VContor[VInflex[9]].x)2 + (VContor[VInflex[7]].y - VContor[VInflex[9]].y)2}
INSERTAR ATRAS Features
{(VContor[VInflex[10]].x - VContor[VInflex[12]].x)2 + (VContor[VInflex[10]].y -
VContor[VInflex[12]].y)2}
RETORNAR Features

```

**FIN**


---

El dedo pulgar que genera características iguales a los otros dedos se encuentra siempre fuera de los bucles de la evaluación debido a sus puntos de inflexión que a diferencia de los dedos anular medio e índice su primer punto de inflexión no es el último punto de inflexión del dedo anterior, como se puede ver en la *Figura 48* donde el punto de inflexión #9 que viene a ser el último del dedo índice no es igual al primero del dedo pulgar como en caso del punto de inflexión #7 que es ultimo del dedo medio y el primero del dedo índice.



*Figura 48.* Ubicación grafica de los puntos de inflexión y centros enumerados.

#### *Ejemplo 11:*

Para hallar las distancias características de la biometría de la mano se necesita:

- Coordenadas de todo el contorno de la mano.
- Coordenadas de los 13 puntos de inflexión.
- Coordenadas de los centros de los cinco dedos.

Las Primeras distancias que se calculan son las distancias que se dan entre los centros de los dedos y los puntos de inflexión que representan a cada dedo respectivamente. Es decir:

- Meñique:

Distancia entre C[0] & PI[1]

Distancia entre C[0] & PI[2]

Distancia entre C[0] & PI[3]



- Anular:
  - Distancia entre C[1] & PI[3]
  - Distancia entre C[1] & PI[4]
  - Distancia entre C[1] & PI[5]
- Medio:
  - Distancia entre C[2] & PI[5]
  - Distancia entre C[2] & PI[6]
  - Distancia entre C[2] & PI[7]
- Índice:
  - Distancia entre C[3] & PI[7]
  - Distancia entre C[3] & PI[8]
  - Distancia entre C[3] & PI[9]
- Pulgar:
  - Distancia entre C[4] & PI[10]
  - Distancia entre C[4] & PI[11]
  - Distancia entre C[4] & PI[12]

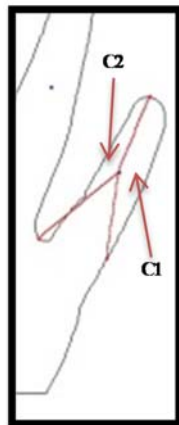
En figura 49 se muestran las distancias generadas al hallar el centro.



Figura 49. Distancias generadas para el dedo meñique

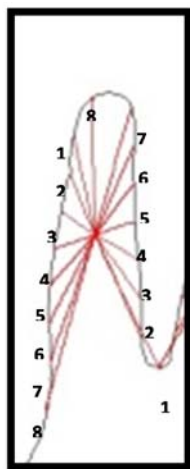
*Ejemplo 12:*

Una vez determinadas las distancias entre los centros y los Puntos de inflexión que representa el dedo, estos generan 2 cuadrantes por dedo, como se muestra en la *Figura 50*.



*Figura 50.* Señalización de los cuadrantes generados al generar las distancias del centro del dedo con sus respectivos puntos de inflexión.

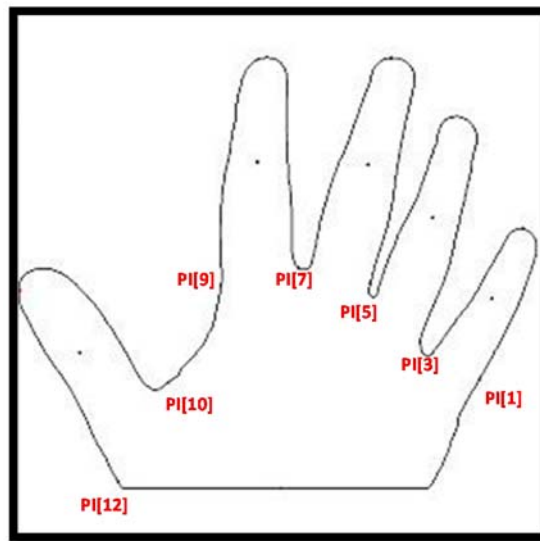
Dentro de cada cuadrante se generan 8 distancias que generan 9 segmentos, como se muestra en la *Figura 51* que muestra las distancias del dedo índice.



*Figura 51.* Representación de las 16 distancias 8 por cada segmento de cada dedo.

*Ejemplo 13:*

Es necesario hallar también el centro de la palma, este se halla sacando un promedio de los puntos de inflexión pero sin contar los puntos de los extremos de los dedos, como se muestra en la *Figura 52*, que muestra los puntos utilizados para hallar el centro de la palma.



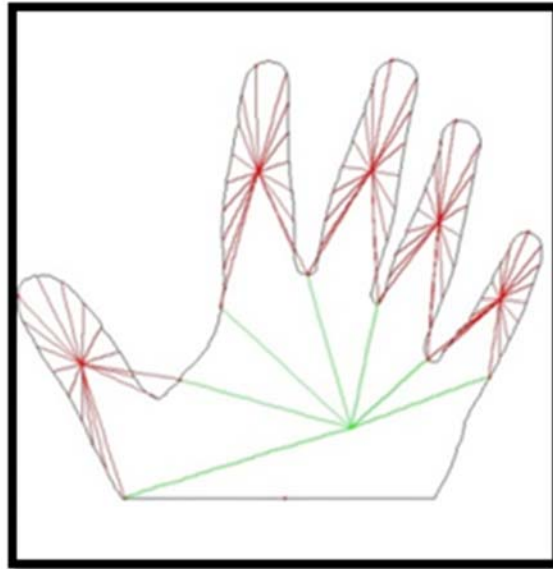
*Figura 52.* Puntos de inflexión necesarios para hallar el centro de la palma de la mano.

Para calcular el centro se genera un bucle que recorre el vector de los puntos de inflexión y dentro de este una condicional que toma solo los puntos anteriormente explicados.

Para el caso de la imagen solo se sumaran las coordenadas [1], [3], [5], [7], [9], [10] y [12]

Una vez obtenida la suma tanto de la coordenada x como la de la coordenada y cada una de estas es dividida entre 7.

Una vez determinada la coordenada del centro de la palma se calcula la distancia entre el centro y los puntos de inflexión [1], [3], [5], [7], [9], [10] y [12], como se muestra en la *Figura 53*, distancias representas por las líneas verdes.

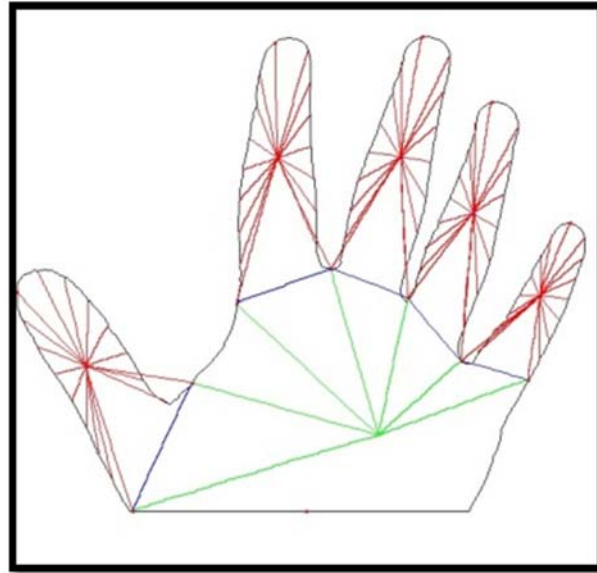


*Figura 53.* Representación gráfica de las distancias generadas a partir de los centros.

*Ejemplo 14:*

Las 5 últimas distancias que se generan son las que representan el ancho de los dedos en la base del dedo que van a ser representados por los mismos puntos de inflexión que determinaron el centro de la palma. Por lo que:

Esta distancia se representa en la *Figura 54* donde se encuentran representadas todas las distancias halladas que son un total de 52(8 por cada dedo=40, 7 distancias centrales y 5 grosores de dedos), siendo las ultimas representadas las pintadas de azul.



*Figura 54.* Representación de todas las distancias (rojo) distancia del centro a los contornos de los dedos (azul) Grosor de los dedos (verde) Centro de la palma.

## 2.5 HERRAMIENTA DE PRUEBA.

Con el propósito de evaluar las variables e hipótesis planteadas como el tiempo de ejecución, segmentación, espectro, y grado de iluminación de las imágenes de prueba se desarrolló dos aplicativos; una aplicación desktop y un batch que permitirá procesar todas las imágenes de prueba con una sola solicitud.

### 2.5.1 Mecanismo Grafico.

En el cual es necesario especificar la ubicación de la imagen, parametrizar los requerimientos, y este mostrara una representación gráfica de las características de la palma escogida. Como se muestra en la *Figura 55.*

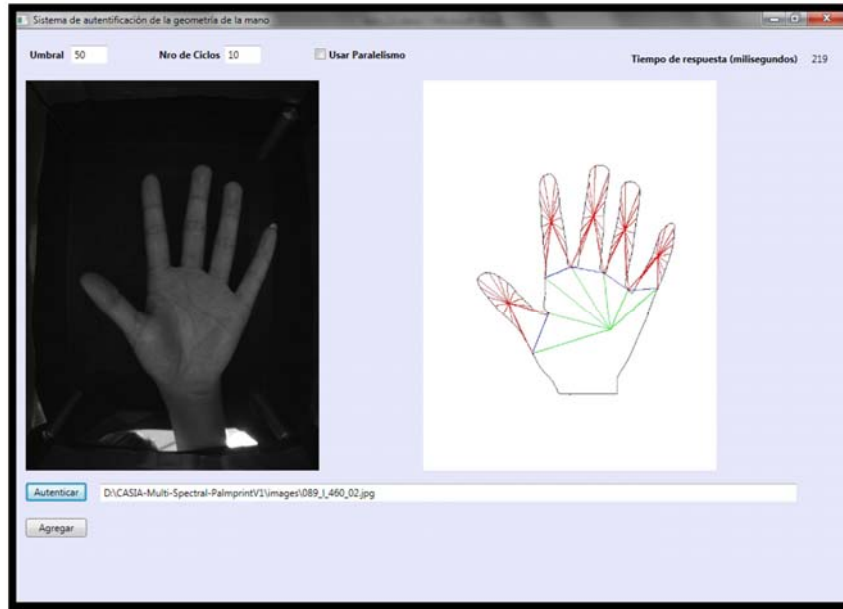


Figura 55. Pantalla con resultados.

La aplicación desktop tiene dos componentes principales:

**a. Motor de autenticación (*ImageProcessing.dll*)**

Esta componente de software tiene programado todo el procesamiento de imágenes y programación paralela empleada en el mecanismo, el lenguaje de programación empleado es C++ con las librerías de procesamiento de imágenes OPENCV y de programación paralela Threading Building Blocks.

Lista de clases:

*ExtractContour*: Esta clase es la encargada de la extracción del borde de la mano, para obtener el perímetro en una estructura de tipo lista enlazada simple.

*HandGeometry*: Esta clase es la clase principal del componente encargado del procesamiento (paralelo y serial) de la imagen.

*ImageRGB*: esta estructura de datos representa una imagen en memoria principal en formato de color RGB en dos dimensiones.

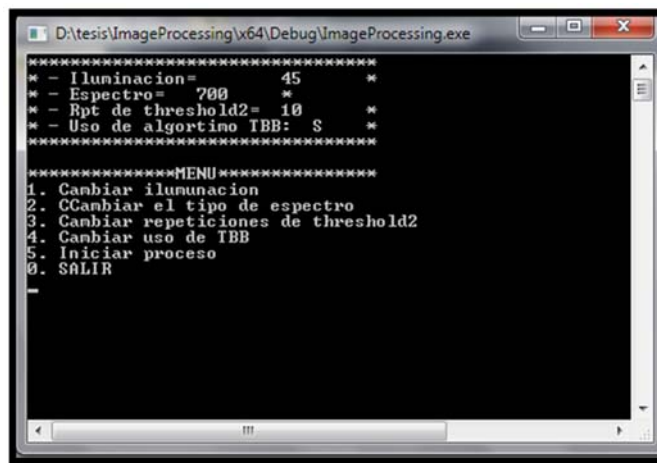
### **b. Vista del mecanismo de sistema desarrollado**

#### **(HandGeometryView.exe)**

Este componente de software representa la parte gráfica del mecanismo de sistema desarrollado, está programado con el lenguaje Visual C# utilizando el framework .NET 4.0 en su versión de 64 bits, la tecnología utilizada es Windows Presentation foundation para el desarrollo de los GUIs (GraphicalUser Interfaces).

### **2.5.2 Batch en Consola.**

Este aplicativo permite coger los parámetros y evaluar un bloque de imágenes con nombres parametrizados retornando el tiempo de ejecución en dicha evaluación. Como se muestra en la *Figura 56*.



```
D:\tesis\ImageProcessing\x64\Debug\ImageProcessing.exe
*****
* - Iluminacion=      45  *
* - Espectro=      700  *
* - Rpt de threshold2= 10  *
* - Uso de algortino TBB: S  *
*****
*****MENU*****
1. Cambiar iluminacion
2. CCambiar el tipo de espectro
3. Cambiar repeticiones de threshold2
4. Cambiar uso de TBB
5. Iniciar proceso
0. SALIR
```

*Figura 56.* Pantalla de batch en Consola de Autenticación

### 2.5.3 Casos de Uso.

Este mecanismo contempla tres casos de uso muy bien definidos los cuales son presentados en la *Figura 57*.

#### ***Cambiar Parámetros.***

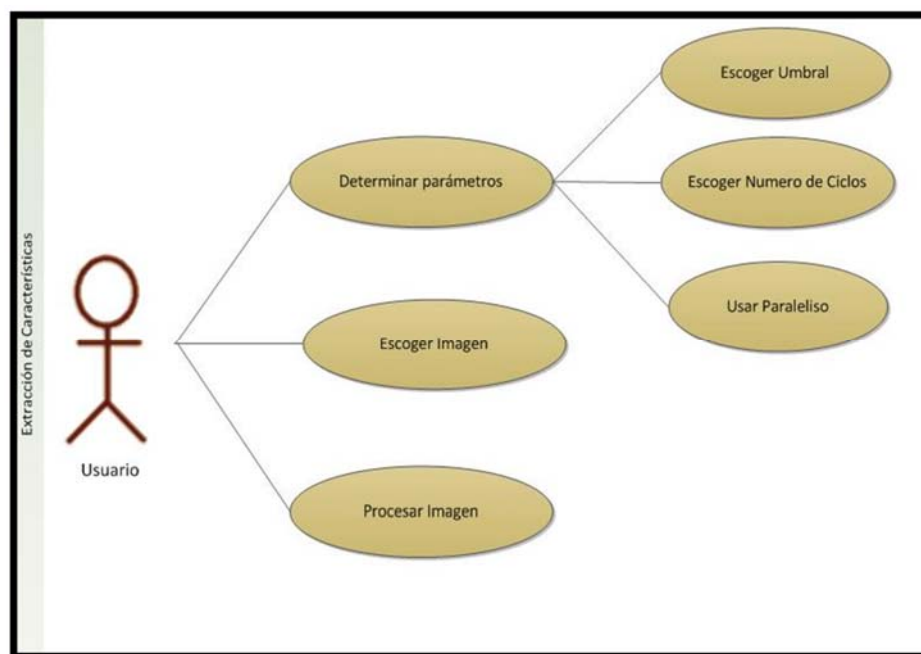
Este caso de uso está orientado a la modificación de parámetros tales como el umbral de iluminación, numero de ciclos de limpieza de la imagen y el uso de programación paralela.

#### ***Escoger Imagen.***

Este caso de uso está orientado a la selección de la imagen de entrada donde se detalla la ubicación y el nombre del archivo donde se encuentra la imagen a procesar.

#### ***Procesar Imagen.***

Es la obtención de la característica de la palma de la mano presente en ella.



*Figura 57.* Diagrama de casos de uso del mecanismo implementado.

#### 2.5.4 Pantallas del sistema.

La vista inicial contiene ocho elementos de datos los cuales funcionan como entradas del usuario o salida del procesamiento, estos elementos se pueden apreciar en la *Figura 58* son:

1. **Umbral.** Este elemento permite modificar el umbral del algoritmo de procesamiento inicial.
2. **Nro. de Ciclos.** Este elemento permite modificar el número de ciclos en la extracción de fondo en el algoritmo principal.
3. **Usar Paralelismo.** Esta opción permite probar el mecanismo de manera serial o paralela.
4. **Tiempo de Respuesta.** Este elemento permite visualizar el número de milisegundos que el proceso principal consume.
5. **Botón Autenticar.** Este elemento permite seleccionar una imagen de la base de datos y da inicio a su procesamiento.
6. **Ruta de archivo de imagen.** Este elemento permite visualizar la ruta del archivo de imagen seleccionado.
7. **Imagen sin procesar.** Este elemento permite visualizar el archivo de imagen tal y como está en la base de datos.
8. **Imagen procesada.** Este elemento permite visualizar el archivo de imagen procesado.



Figura 58. Pantalla inicial del mecanismo.

En la figura se muestra la imagen una vez procesada, donde se ve la representación gráfica de las características de la palma.

### 2.5.5 Distribución de indicadores.

Los resultados pueden ser de dos tipos, (a) *Tiempo de ejecución* y (b) *Algoritmo de extracción de características*. Esto es debido a que las condiciones de operación del algoritmo de extracción de características se mantienen constantes y no dependen del tiempo de ejecución.

En Tabla 5 muestra la relación entre el tipo de resultados y las variables independientes ya definidas.

Tabla 5.

*Relación del Tipo de Resultado y las Variables independientes.*

<b>Tipo de Resultado</b>	<b>Variables Independientes</b>
Tiempo de ejecución	(1) Uso de la librería TBB en el procesamiento de las imágenes de entrada.
	(2) Memoria principal y número de núcleos físicos de la unidad central de procesamiento.
Algoritmo de extracción de características	(3) Segmentación de la imagen de la biometría de la mano.
	(4) Espectro de luz empleado en la toma de imagen.
	(5) Grado de iluminación al momento de la segmentación.

(elaborador propia)

El tiempo de ejecución se ve reflejado en las variables 1 y 2 de la Tabla 5.

Y el algoritmo de extracción de características se ve reflejado en las variables 3, 4 y 5 de la Tabla 5.

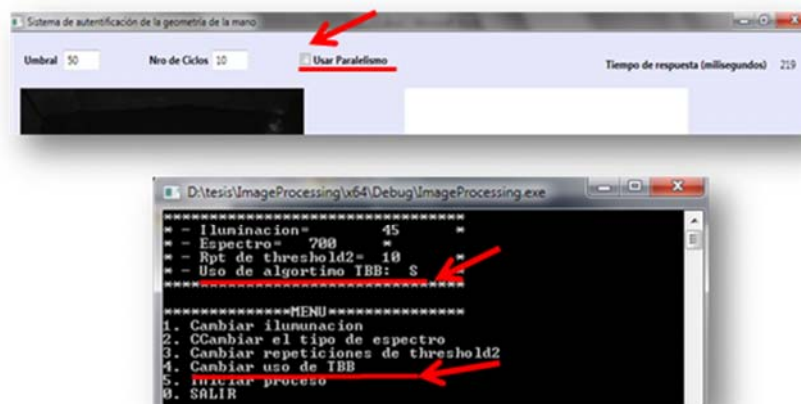
***a. Variable 1(Usó de la librería TBB en el procesamiento de las imágenes de entrada).***

La librería TBB fue usada en los bucles de evaluación pixel por pixel; los cuales forman parte de los tres subprocesos siguientes.

- Transformación de la imagen a blanco y negro.
- Limpieza de la imagen.
- Extracción del contorno de la mano.

El grado de manipulación de esta variable es de “Presencia-Ausencia” por lo que la misma Base de datos será evaluada como grupo experimental así como de control.

Para la respectiva comparación de eficiencia de su uso se colocó en los aplicativos un flag donde se pudiera determinar si se va a utilizar la librería en el proceso o no, como se muestra en la *Figura 59*.



*Figura 59.* Opciones de uso librería TBB.

**b. Variable 2(Memoria principal y número de núcleos físicos de la unidad central de procesamiento).**

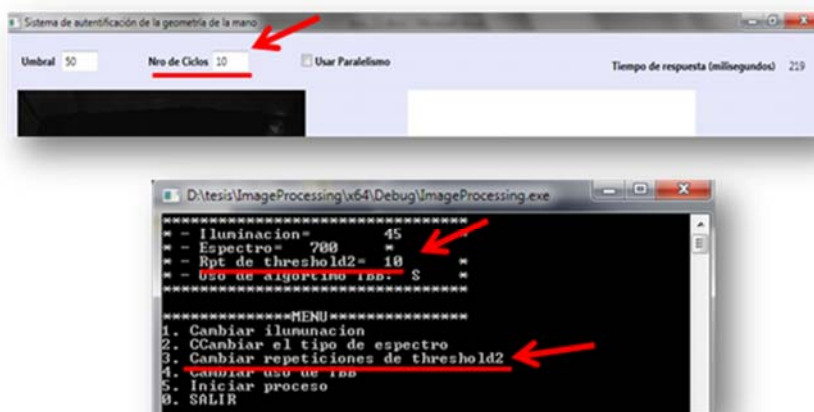
El grado de manipulación de esta variable es la denominada “Más de dos grados” que se ve reflejada en los computadores utilizados para realizar las pruebas.

**c. Variable 3(Segmentación de la imagen de la biometría de la mano).**

El grado de manipulación de esta variable es de “Modalidad de manipulación”.

Para la respectiva comparación de eficiencia de su uso se colocó en los aplicativos una opción para que el número de segmentaciones pueda ser variado, en el caso del mecanismo grafico se utilizó una

caja de texto; y en el batch una opción en el menú principal. Como se muestra en la *Figura 60*.



*Figura 60.* Opción de número de segmentaciones.

**d. Variable 4 (Espectro de luz empleado en la toma de imagen).**

El grado de manipulación de esta variable es de “Más de dos grados”.

La base de datos de las imágenes cuenta con la misma imagen tomada con 6 diferentes espectros. (a) 460, (b) 630, (c) 700, (d) 850, (e) 940 y (f) Blanco.

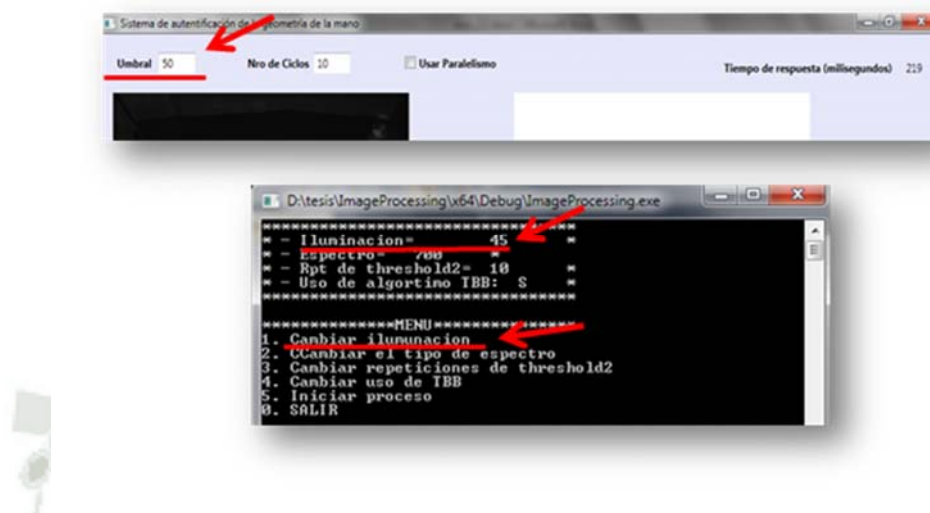
En el caso del mecanismo gráfico el espectro será escogido al momento de escoger la imagen evaluar.

Y en el caso del batch se colocó una opción en el menú.

**e. Variable 5 (Grado de iluminación al momento de la segmentación).**

El grado de manipulación de esta variable es de “Modalidad de manipulación”.

A igual que la variable de segmentación se colocó en los aplicativos una opción para que el grado de iluminación pueda ser variado. Como se muestra en la *Figura 61*.



*Figura 61.* Opción del Grado de Iluminación.

## 2.6 REQUISITOS DE DESARROLLO

El desarrollo de ambos aplicativos se realizaron bajo los siguientes requisitos:

### 2.6.1 Confiabilidad.

Ambos aplicativos emiten siempre los mismos resultados.

### 2.6.2 Validez en Contenido.

Ambos aplicativos tienen parametrizadas las propiedades así como también resultados cuantitativos, obteniendo un dominio específico de lo que se requiere.

### 2.6.3 Validez en Criterio.

El aplicativo en consola es probado en diferentes maquinas con diferentes procesadores evaluado en cada uno su rendimiento.

### 2.6.4 Validez de Constructo.

El aplicativo logra representar y medir las variables de la presente tesis.

### 2.6.5 Objetiva.

Tanto el mecanismo grafico como el batch de consola tienen la misma estandarización de proceso logrando evaluar todas las imágenes de la misma manera y bajo las mismas condiciones.



### CAPITULO III EVALUACION Y DISCUSION DE LOS RESULTADOS

En el presente capitulo se realiza la síntesis de los resultados organizada en dos partes. En la primera, se evalúa las mejores condiciones para el algoritmo de reconocimiento combinando las variables espectro, umbral de iluminación y segmentación. En la segunda parte, se evalúa y compara el tiempo de ejecución del algoritmo con y sin programación paralela en diferentes equipos de diferentes características

#### 3.1. Clasificación de las palmas.

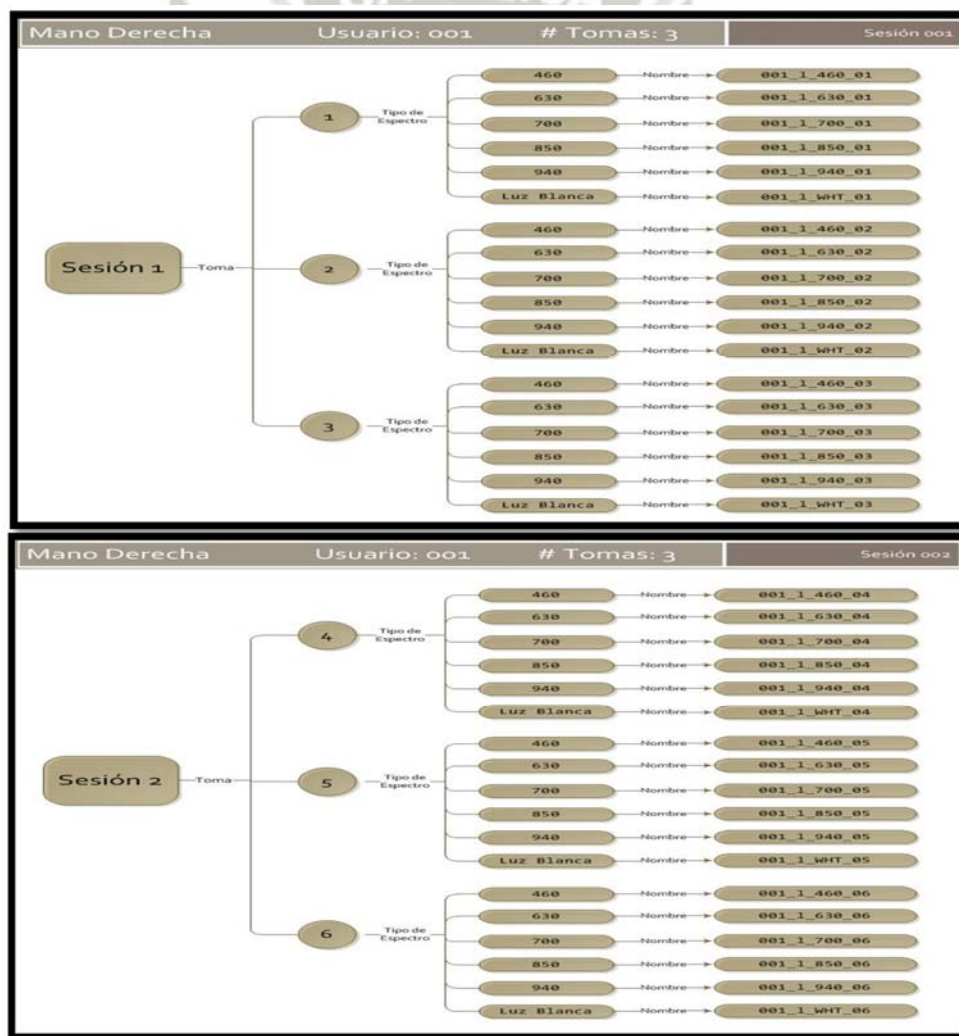


Figura 62. Esquema de las sesiones tomadas para la obtencion de la imagen

De las 7200 imágenes de palmas tomadas de (Center for Biometrics and Security Research, 2010) solo se utilizaron 3600 palmas que vienen a ser de las manos izquierdas de 100 individuos distintos tomadas en 2 sesiones cada sesión con una diferencia de tiempo mayor a un mes. En cada sesión se tomaron 3 imágenes por cada tipo de espectro siendo 6 los tipos de espectros evaluados como se ve en el esquema de la *Figura 62*.

### 3.2. Resultados de la variable de Grado de iluminación Vs. Espectro utilizado.

En el anexo B se muestra la imagen de la palma evaluada con todos los posibles grados de iluminación en un rango de 10 entre cada prueba y los 6 tipos de espectro, donde:

A menor umbral de iluminación mayor cantidad de pixeles blancos.

A mayor umbral de iluminación mayor cantidad de pixeles negros.

El grado de iluminación adecuado depende del espectro utilizado como se ve en la *Tabla 6*.

Tabla 6.

*Umbral Para el Reconocimiento de Manos Uniformes Según Espectro.*

	Blanco Completo	Fondo Uniforme	Mano Uniforme	Picos En el Borde	Negro Completo
<b>460</b>	0-10	40	40	40	400
<b>630</b>	0-10	30	30	50	400
<b>700</b>	0-10	30	30	60	400
<b>850</b>	0-10	40	50	70	500
<b>940</b>	0-10	40	50	70	500
<b>BLANCO</b>	0-10	40	30	60	500

(Elaboración Propia)

Algunos picos en los contornos de la palma como se ve en la *Figura 63*, lo cual generan errores en la extracción de los puntos de inflexión.



*Figura 63.* Picos del contorno de la palma.

### 3.3. Segmentación de la imagen

La segmentación permite homogenizar la imagen, quitando píxeles que se filtran como parte de la palma pero que en realidad son parte de la iluminación de la palma. En el anexo D se muestran imágenes tipo con distintos grados de segmentación.

### 3.4. Combinación de las variables Espectro-Iluminación–Segmentación.

Al combinar Espectro-Iluminación–Segmentación se detectó tres tipos de fallos que generan una fallida extracción de características.

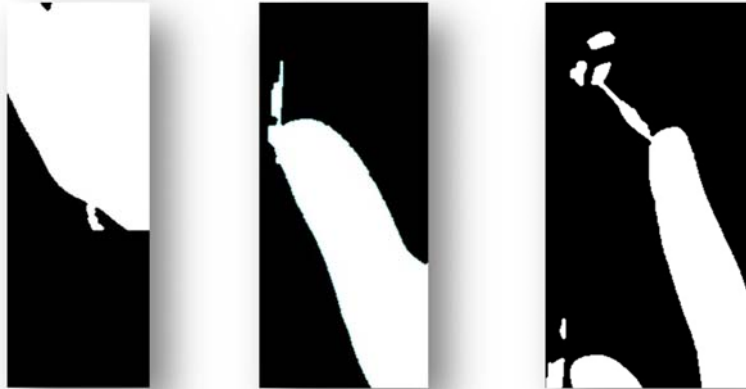
Para hallar los puntos de inflexión se parte de encontrar el punto más alto, para luego encontrar el más bajo y así intercalando hasta volver al punto de inicio, cuando la imagen no es completamente homogénea el sistema encuentra picos que generan distintos tipos de errores.

### 3.4.1. Error Tipo 1.

Existen picos que invierten la busqueda de puntos mas bajos y mas altos.

En las siguientes imágenes de la *Figura 64*.

### 3.4.2. Error Tipo 2.



*Figura 64.* Errores que desvian el verdarero contorno de la palma.

Los picos son confundidos por dedos como se muestra en las siguientes imágenes de la *Figura 66*, Obteniendo más dedos de los realmente existentes, o en caso contrario como en la

*Figura 65* que reconoce como dedo a una protuberancia invirtiendo la búsqueda de puntos de inflexión por lo q no reconoce menos dedos de los que debería.



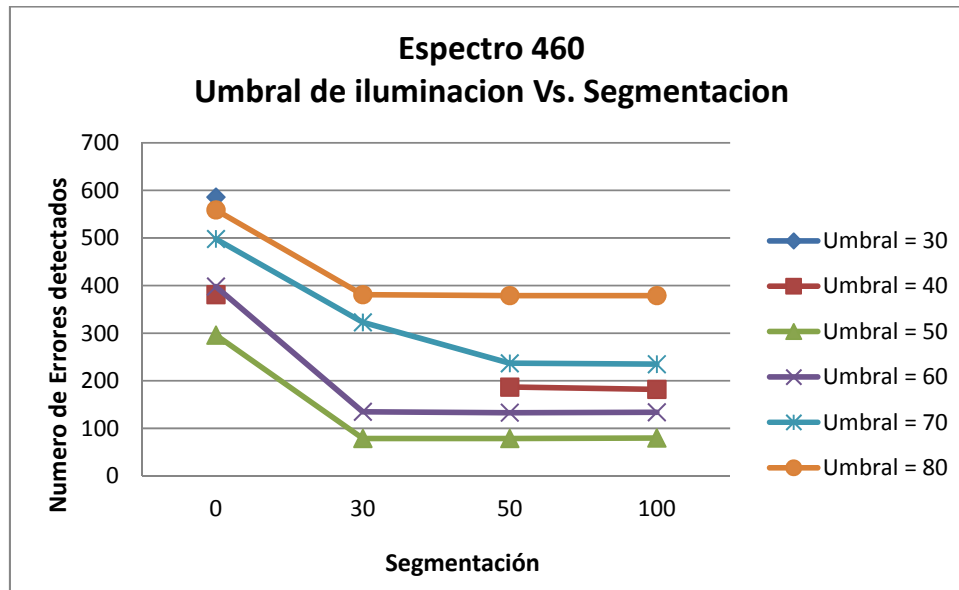
*Figura 65.* Existen Picos que no permiten generar los puntos de inflexión completos.



*Figura 66.* Picos que son confundidos por dedos.

Una vez identificados los tipos de errores que se pueden presentar al momento de evaluar la palma, se evaluaron las 3600 palmas es decir en todos los espectros con distintos umbrales de iluminación y distintos números de segmentación para encontrar los valores donde se genera la menor cantidad de errores ya descrito. Los resultados de esta evaluación se resumen en el anexo D. Donde en un primer momento se evaluaron todos los umbrales de iluminación posibles de reconocimiento en un rango de 10 en 10 y con los valores de segmentación de (a)0, (b) 30 ,(c) 50 y (c) 100 de los cuales los valores con menor margen de error por

cada de espectro se representan en la *Figura 67* para el espectro 460, *Figura 68* para el espectro 630, *Figura 69* para el espectro 700, *Figura 70* para el espectro 850, *Figura 71* para el espectro 950 y la *Figura 72* para el espectro blanco.



*Figura 67.* Gráfico de líneas que representa la cantidad de errores según iluminación - segmentación del Espectro 460

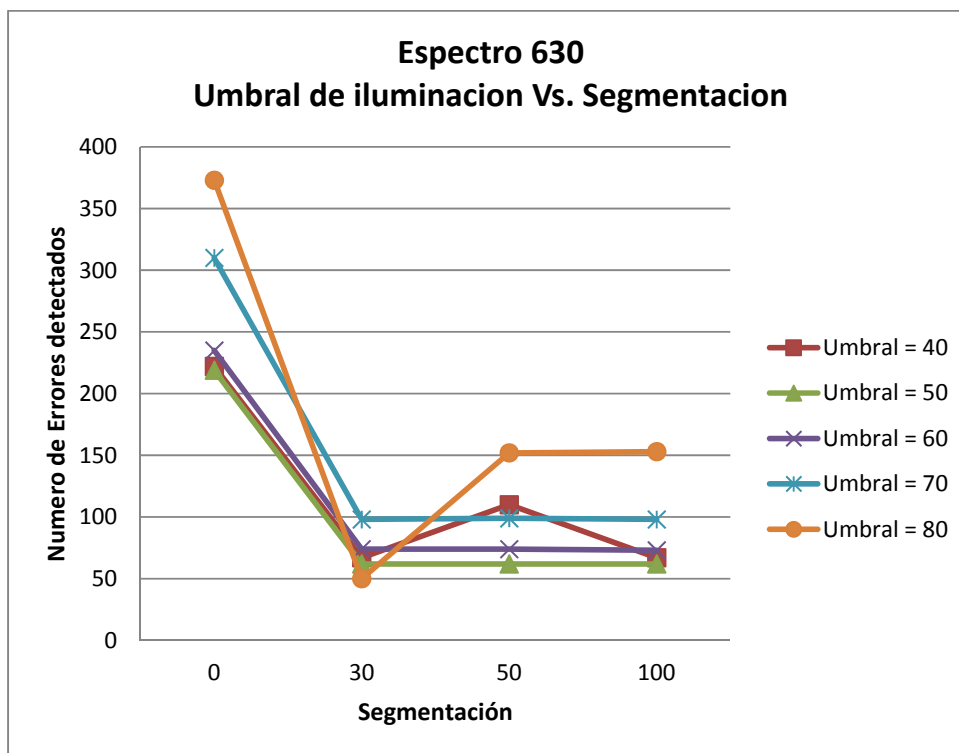


Figura 68. Gráfico de líneas que representa la cantidad de errores según iluminación - segmentación del Espectro 630.

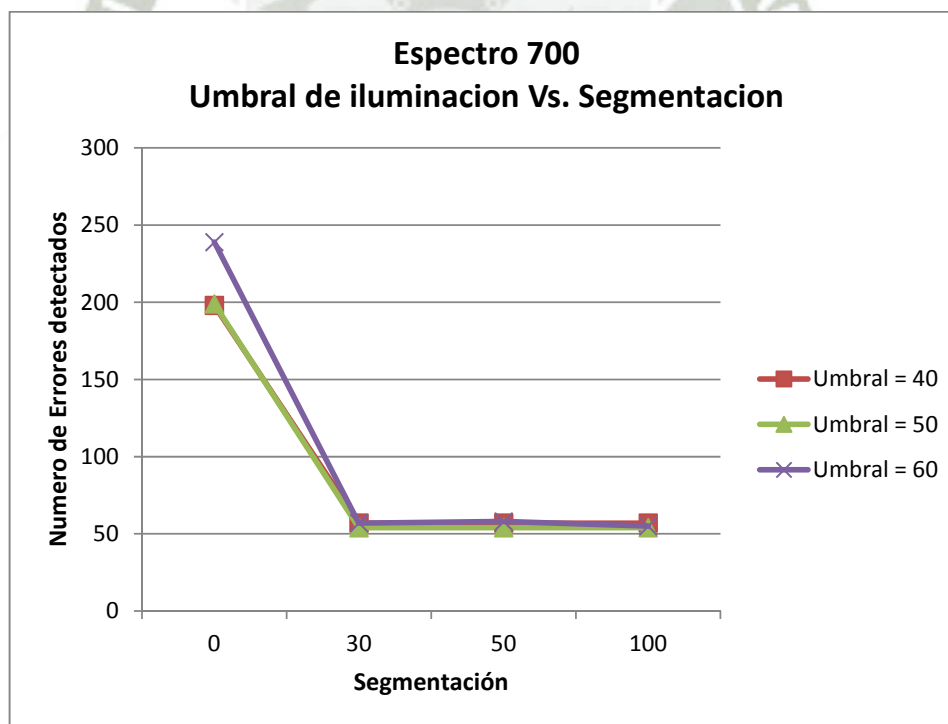


Figura 69. Gráfico de líneas que representa la cantidad de errores según iluminación - segmentación del Espectro 700.

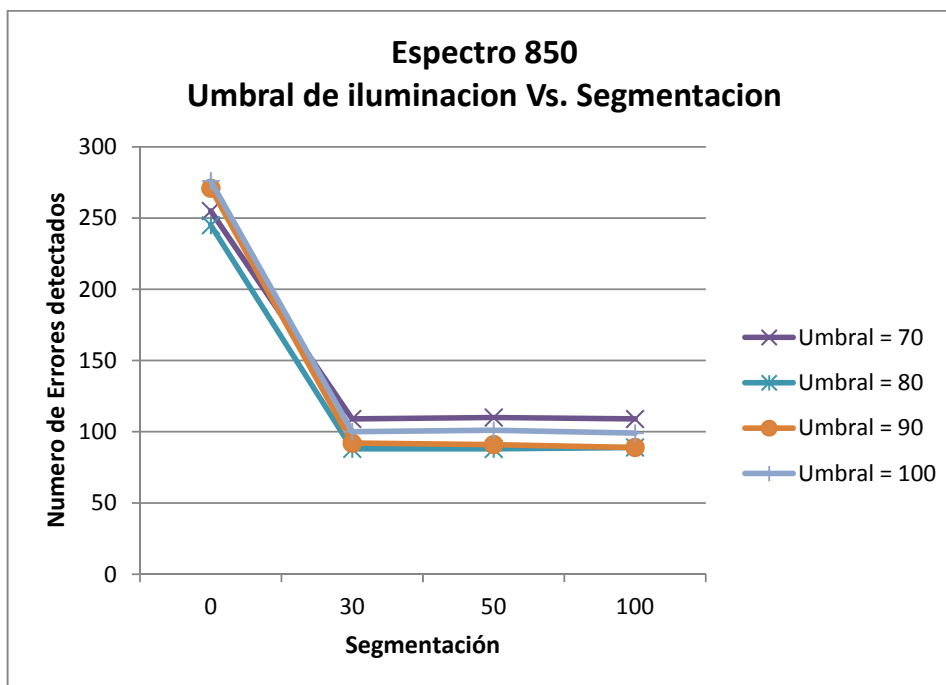


Figura 70. Gráfico de líneas que representa la cantidad de errores según iluminación - segmentación del Espectro 850.

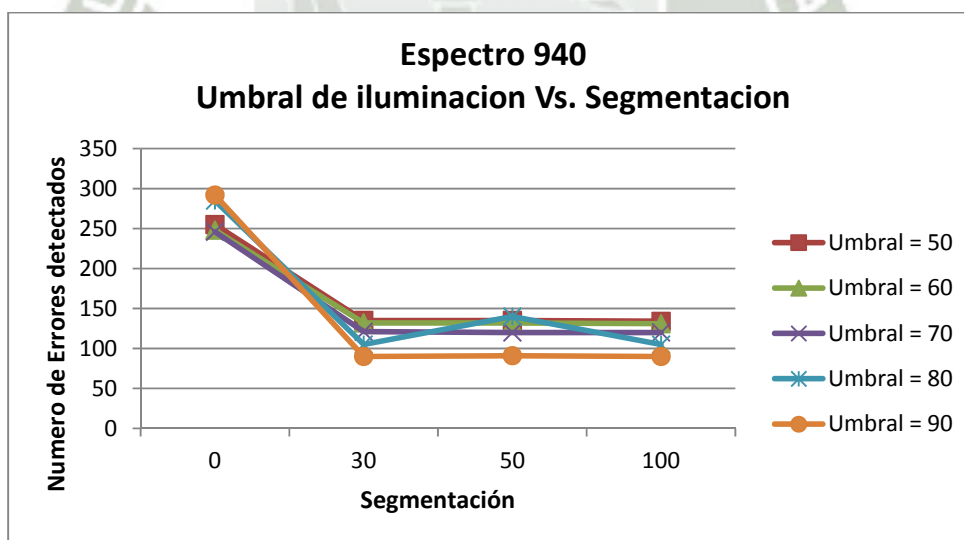


Figura 71. Gráfico de líneas que representa la cantidad de errores según iluminación - segmentación del Espectro 940.

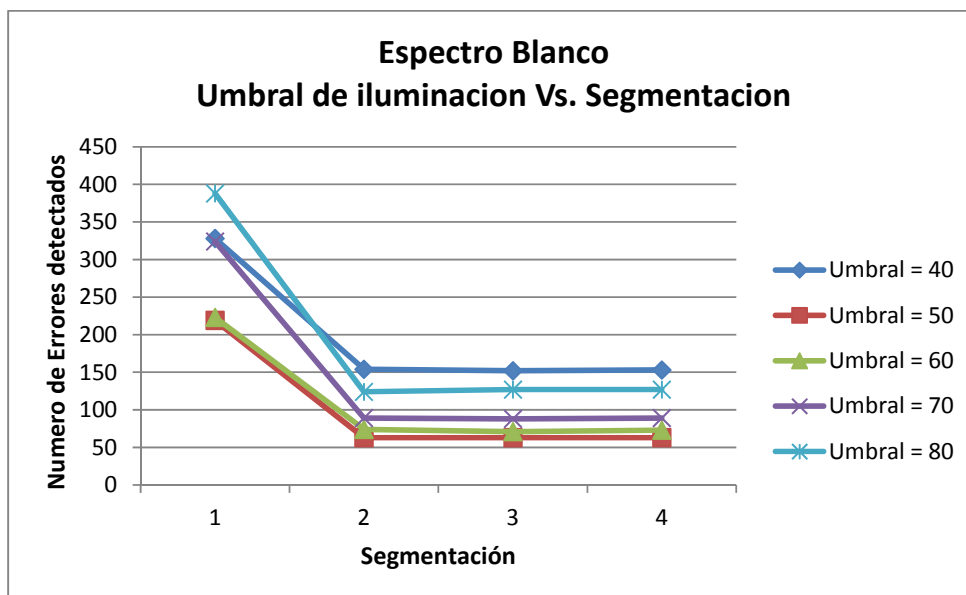


Figura 72. Gráfico de líneas que representa la cantidad de errores según iluminación - segmentación del Espectro Blanco.

Donde se puede observar que:

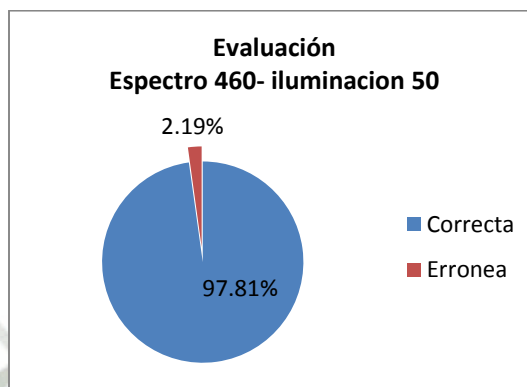
- De los 6 espectros evaluados el espectro que genera menos errores es el espectro de valor 700.
- La iluminación que se debe utilizar para una mejor performance depende del espectro.
- El 1.5% es el porcentaje mínimo de errores que se generan al reconocer una palma.

Una vez hallado estos valores se evaluaron sus vecinos más cercanos para hallar una mejor precisión. Los resultados de esta evaluación se describen en el Anexo D.

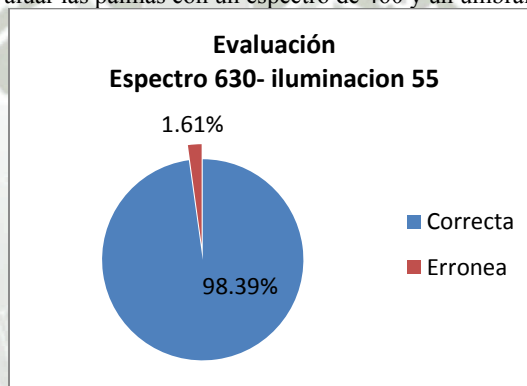
Dónde:

- Se confirma que el espectro 700 sigue siendo el más óptimo, con un umbral de iluminación en el rango de 45 a 55 con el 1.5% de margen de

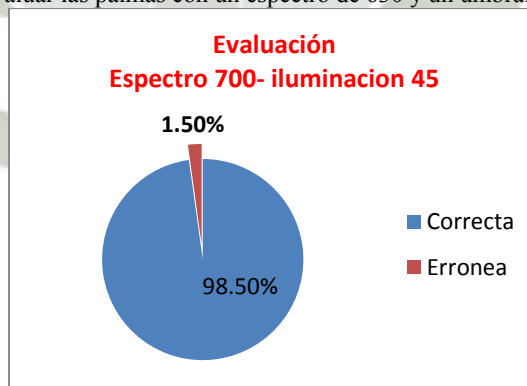
error, Al ser comparado con los demás espectros como se representan en la *Figura 73* para el espectro 460, *Figura 74* para el espectro 630, *Figura 75* *Figura 76* y *Figura 77* para el espectro 700, *Figura 78* para el espectro 850, *Figura 79* para el espectro 950 y la *Figura 80* para el espectro blanco.



*Figura 73.* Margen de error al evaluar las palmas con un espectro de 460 y un umbral de iluminación de 50.



*Figura 74.* Margen de error al evaluar las palmas con un espectro de 630 y un umbral de iluminación de 55.



*Figura 75.* Margen de error al evaluar las palmas con un espectro de 700 y un umbral de iluminación de 45.

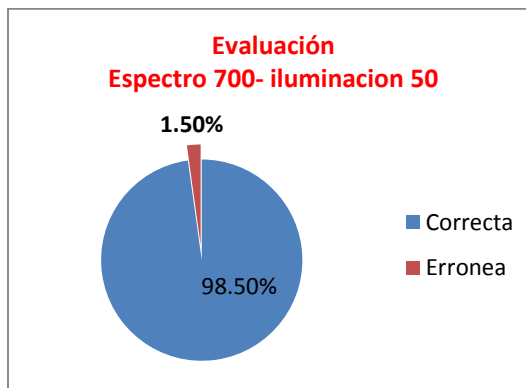


Figura 76. Margen de error al evaluar las palmas con un espectro de 700 y un umbral de iluminación de 50.

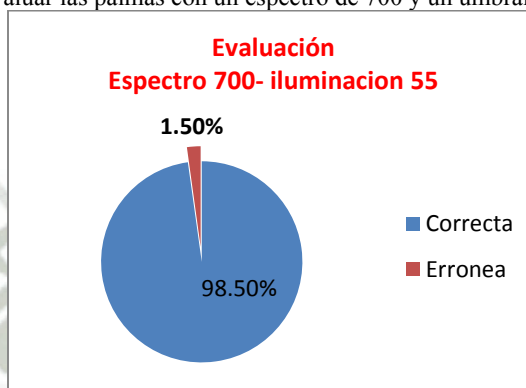


Figura 77. Margen de error al evaluar las palmas con un espectro de 700 y un umbral de iluminación de 55.

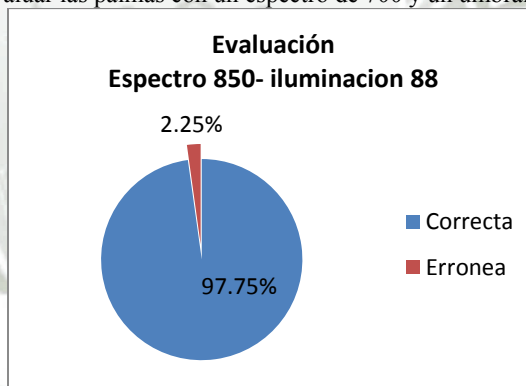


Figura 78. Margen de error al evaluar las palmas con un espectro de 850 y un umbral de iluminación de 50.

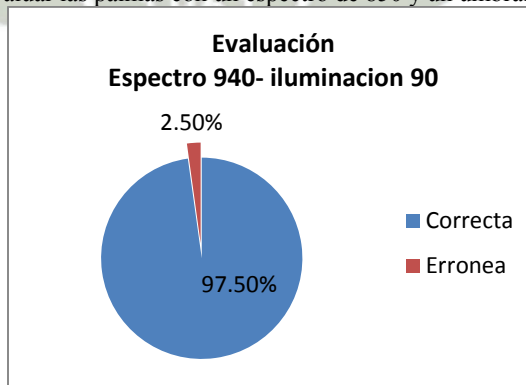
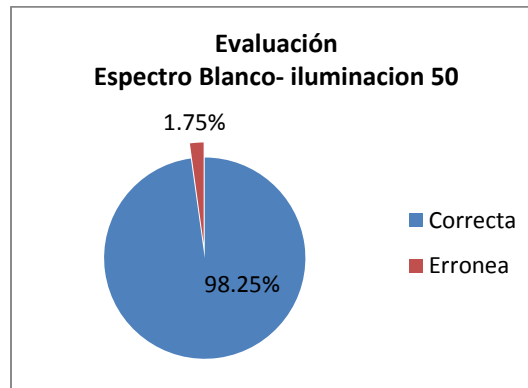
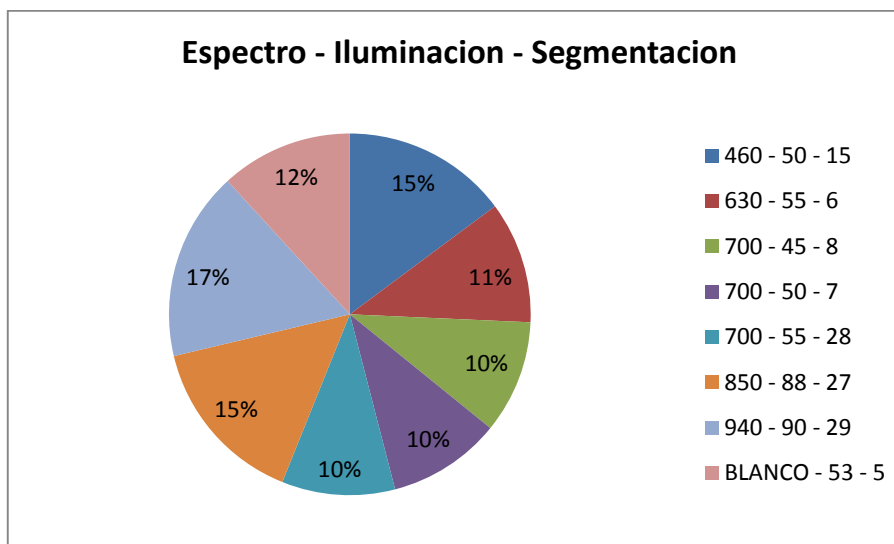


Figura 79. Margen de error al evaluar las palmas con un espectro de 940 y un umbral de iluminación de 90.



*Figura 80.* Margen de error al evaluar las palmas con un espectro de Blanco y un umbral de iluminación de 50.

- Una vez obtenida la iluminación óptima por cada espectro se procede a determinar la segmentación de la imagen. La evaluación anterior determinó que el número de segmentaciones tiene un tope y se puede determinar que el tope máximo de segmentaciones es 30. Se hace una segunda evaluación en base a los datos encontrados, para encontrar una mejor precisión, de donde se determina que el número de segmentaciones depende del tipo de espectro escogido, y este va desde 7 hasta 29 como se ve en la gráfica de la *Figura 81*. Los resultados de estas evaluaciones se encuentran detallados en el Anexo E.



*Figura 81.* Porcentaje de Error en Comparación a Todas las Evaluaciones Espectro Iluminación y Segmentación.

### 3.5. Tiempo de ejecución.

Una vez halladas las condiciones necesarias para un óptimo y eficaz reconocimiento se evalúa el tiempo que se reduce utilizando la librería TBB para lo cual se lleva a cabo pruebas en diferentes equipos.

Para esta evaluación, las condiciones de operación como son umbral de iluminación, segmentación de la imagen y espectro escogido se mantuvieron constantes en el análisis de las imágenes, para poder tomar resultados confiables con respecto al tiempo de evaluación.

#### 3.5.1. Equipos Utilizados.

Los equipos de cómputo utilizados para evaluar las tres mil seiscientas imágenes de palmas de manos izquierdas fueron 9, sus especificaciones se encuentran en la Tabla 7

Tabla 7.

*Lista de Equipos Utilizados en las Pruebas*

Computador	Modelo Procesador	Número de Núcleos	Frecuencia de Reloj	Memoria
1	Intel Core TM i7-3770	7	3.40Ghz	8Gb
2	Intel Core TM I7 950	7	3.07Ghz	6Gb
3	Intel Core TM I7 Q720	7	1.60Ghz	4Gb
4	Intel Core TM i5	5	2.77Ghz	4Gb
5	Intel Core TM 2Quad Q9000	4	2.00Ghz	4Gb
6	Intel Core TM i3 2120	3	3.30Ghz	4Gb
7	Intel Core TM i3 2370	3	2.40Ghz	4Gb
8	Intel Core TM i3 M330	3	2.13Ghz	4Gb
9	AMD Turion X2 Ultra Dual Core	2	2.30Ghz	4Gb

(Elaboración Propia)

### 3.5.2. Resultados en tiempo de ejecución.

Por cada equipo se tomaron 8 muestras 4 de ellas como grupo de control y las cuatro últimas como grupo experimental los tiempos obtenidos se muestran en anexo F.

En la *Figura 82* se observa una reducción considerable en el tiempo al usar las librerías TBB.

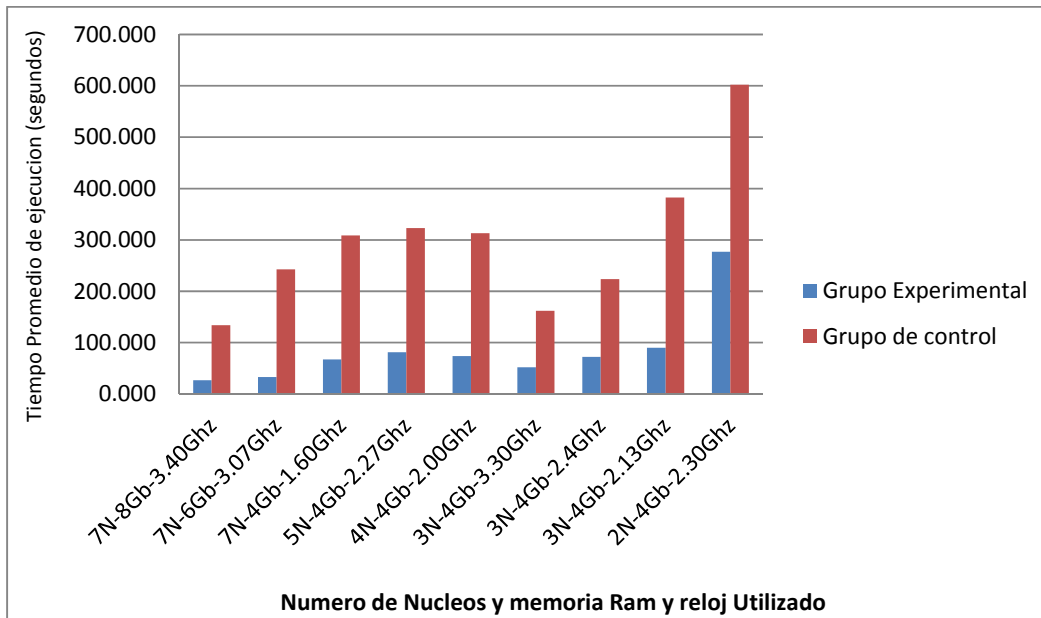


Figura 82. Comparación en el Tiempo Total Utilizado en la Simulación.



## CONCLUSIONES

1. El uso de una implementación basada en bloques paralelos permite una optimización en el tiempo consumido por el algoritmo principal que puede ir desde 2.17 veces más rápido hasta 7 veces dependiendo del número de procesadores pudiendo mejorar su eficiencia a medida que se van aumentando la cantidad de procesadores.
2. Las condiciones más óptimas de las variables Espectro – Iluminación – Segmentación son:
  - Espectro: 700
  - Umbral de iluminación: 50
  - Numero de Segmentaciones: 7
3. La calidad de la imagen de entrada es determinante en el procesamiento, ya que el perímetro depende de ésta.
4. La palma de la mano no debe de tocar los bordes de la imagen ya que la identificación del perímetro se realiza utilizando el fondo que envuelve la palma.
5. El rendimiento de los resultados se ve fuertemente influenciado por la frecuencia del reloj que tiene cada procesador.
6. Todos los espectros han presentado un margen de error menor al 5%, por lo que es posible utilizar cualquiera de estos espectros para el reconocimiento, pero tomando en cuenta las condiciones más óptimas para el reconocimiento de la palma.

## RECOMENDACIONES

1. Se debe realizar un estudio utilizando más tipos de procesadores, como súper computadoras con más de cien núcleos.
2. Existen alternativas a la utilización de bloques paralelos por CPU, que son bloques paralelos en GPU o procesamiento basado en clusters de computadoras, el presente modelo podría ser desarrollado bajo esos enfoques.
3. El estudio actual hacer un análisis del perímetro de la mano a demostrando mayor eficiencia en el tiempo de ejecución, por lo que si se utilizaran imágenes en tres dimensiones se podría realizar un estudio a la superficie de la palma de la mano.
4. Esta aplicación como tal sirve como base para la creación de un sistema real, capaz de ser implementado en una institución.

## TRABAJOS FUTUROS

1. Proponer una mejora al algoritmo actual para el soporte del procesamiento de cualquier posición de la mano de entrada.
2. Proponer una mejora al algoritmo actual para el soporte del procesamiento de la mano derecha o izquierda.
3. Proponer una mejora al algoritmo actual para el soporte del tamaño de las uñas identificando y procesando solo los segmentos que son de color de piel humana.



## REFERENCIAS BIBLIOGRAFICAS

- (Diciembre 2012). *Estudio sobre las tecnologías biométricas aplicadas a la seguridad*. España: INTECO, Observatorio de la Seguridad de la Información.
- Anil K. Jain & Arun Ross , Sharath Pankanti. (March 22-24, 1999.). A Prototype Hand Geometry-based Verification System. *Appeared in Proc. of 2nd Int'l Conference on Audio- and Video-based Biometric Person Authentication (AVBPA)*, (págs. D.C., pp.166-171). Washington.
- Caroline Pantofaru, M. H. (2005). *A Comparison of Image Segmentation Algorithms*. Pennsylvania: The Robotics Institute Carnegie Mellon University Pittsburgh.
- Center for Biometrics and Security Research. (08 de 2010). *CASIA Multi-Spectral Palmprint Database*. Obtenido de <http://biometrics.idealtest.org/>
- Dorin Comaniciu, P. M. (2002). *Mean shift: A robust approach toward feature space analysis*. IEEE Transactions on Pattern Analysis and Machine Intelligence.
- ed, w. C. (2006). *Unidad Aritmetica Logica*. Obtenido de <http://es.wikipedia.org:>  
[http://es.wikipedia.org/wiki/Unidad\\_aritm%C3%A9tica\\_l%C3%B3gica](http://es.wikipedia.org/wiki/Unidad_aritm%C3%A9tica_l%C3%B3gica)
- EMMANUELLE, H. S. (2007). *“FUNDAMENTOS EN LA DIGITALIZACIÓN Y PROCESAMIENTO DE UNA IMAGEN”*. UNIVERSIDAD VERACRUZANA.
- Etxeberria, J. A. (210). Color tracking - RGB. En J. A. Etxeberria, *Algoritmo de reconocimiento de forma y color para una plataforma robótica* (pág. 15).
- Fraga, L. G. (7 de mayo de 2001). *Curso: Procesamiento Digital de Imágenes*.
- Iturrate, E. (1998). *innovanet- CURSO BASICO DE TELEDETECCION CON ENVI*. Obtenido de innovanet.com: <http://www.innovanet.com.ar/gis/TELEDETE/TELEDETE/pant2.htm>
- Karlovec, M. (12 October, 2010). Conference on Data Mining and Data Warehouses (SiKDD 2010). *HAND GEOMETRY PERSONAL AUTHENTICATION USING MACHINE LEARNING*. Ljubljana, Slovenia: SiKDD 2010.
- Martin, M. (21 de mayo de 2002). *Técnicas Clásicas de Segmentación de Imagen*.

- Martínez, D. D. (2010). *Universidad Nacional Del NordEste - Republica Argentina*. Obtenido de  
Procesos y Administración del Procesador:  
<http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/SO2.htm>
- P. Felzenszwalb, D. H. (2004). *"Efficient Graph-Based Image Segmentation"*. Intl Journal of  
Computer Vision.
- Pankanti, A. K. (1999). *A Prototype Hand Geometry-based Verification System*. Washington:  
Appeared in Proc. of 2nd Int'l Conference on Audio- and Video-based Biometric Person  
Authentication (AVBPA).
- R. Gayathri & P. Ramamoorthy. (2012). A Fingerprint and Palmprint Recognition Approach Based on  
Multiple Feature Extraction. *European Journal of Scientific Research*, ISSN 1450-216X Vol.76  
No.4 (2012), pp.514-526.
- Sevilla, U. d. (s.f.). *Biblioteca de Ingeniería de la Universidad de Sevilla*. Obtenido de SEGMENTACIÓN  
DE IMÁGENES EN COLOR BASADA EN EL CRECIMIENTO DE REGIONES:  
[http://bibing.us.es/proyectos/abreproy/11875/fichero/Proyecto+Fin+de+Carrera%252F3.  
Espacios+de+color.pdf](http://bibing.us.es/proyectos/abreproy/11875/fichero/Proyecto+Fin+de+Carrera%252F3.Espacios+de+color.pdf)
- Tanenbaum, A. S. (1993). *Sistemas Operativos Modernos*. En A. S. Tanenbaum, *Sistemas Operativos  
Modernos*. México: Prentice Hall ,Hispanoamericana.
- Vandana Roy and C. V. Jawahar. (December 2005). *Feature Selection for Hand-Geometry based  
Person Authentication*. Coimbatore: Proceedings of the Thirteenth International  
Conference on Advanced Computing and Communications.
- wikipedia. (2012). *wikipedia - Jet Propulsion Laboratory*. Obtenido de <http://en.wikipedia.org>:  
[http://en.wikipedia.org/wiki/Jet\\_Propulsion\\_Laboratory](http://en.wikipedia.org/wiki/Jet_Propulsion_Laboratory)
- wikipedia. (2012). *wikipedia- Espacio de color Lab*. Obtenido de  
[http://es.wikipedia.org/wiki/Espacio\\_de\\_color\\_Lab](http://es.wikipedia.org/wiki/Espacio_de_color_Lab)

## ANEXO A

### PLANTEAMIENTO DE LA INVESTIGACION

#### 1.1 Planteamiento del Problema.

##### a. Caracterización del problema.

Hoy en día el proceso de autenticación de personas es de vital importancia para asegurar el correcto acceso a los recursos de una institución.

Las técnicas de autenticación basadas en claves y usuarios, no garantizan la autenticidad de la identidad de la persona, que trate de acceder a estos recursos, además existen otras técnicas basadas en tarjetas de acceso pero éstas también sufren del mismo inconveniente. Es así que surgen técnicas de autenticación biométricas, las cuales permiten asegurar la identidad de la persona que desea acceder a los recursos previamente mencionados, ya que se utiliza una medición de alguna característica biológica propia de la persona para realizar este proceso, estas técnicas sufren de un alto consumo de recursos computacionales, puesto que las características biológicas a procesar deben ser almacenadas primero en un medio de ingreso digital (Fotografías, videos, grabaciones de voz, grabaciones mediante infrarrojos, etcétera), y estos medios contienen mucha información que debe ser pre-procesada a fin de poder obtener la información más relevante a utilizar.

#### 1.2 Objetivos de la Investigación.

##### a. General.

La presente tesis formula la propuesta de utilizar técnicas de computación paralelas basadas en la utilización de bloques de construcción de hebras de software en el la autenticación de personas basada en la geometría de mano

con el fin de mejorar el tiempo que toma la preparación de la imagen y la obtención de sus características.

**b. Específicos.**

- Desarrollar un componente de software para la autenticación de personas basándose en las mediciones de la geometría de su mano que permita demostrar que el uso de la programación paralela por bloques es mejor frente a una programación convencional (técnicas seriales).
- Determinar las condiciones físicas y virtuales, óptimas en la autenticación de personas basándose en las mediciones de la geometría de su mano.

**1.3 Preguntas de Investigación.**

Las preguntas de la investigación a resolver son las siguientes:

- ¿Cuánto es el tiempo que se reduce al utilizar técnicas de computación paralelas basadas en la utilización de bloques de construcción de hebras de software en el pre-proceso y proceso de la autenticación de personas basada en la geometría de mano?
- ¿Cuánta diferencia de tiempo existe al probar el mecanismo con diferente número de procesadores?

**1.4 Área Científica a la que corresponde el Problema.**

**a. *Inteligencia Artificial.***

- Reconocimiento de patrones.

## 1.5 Solución propuesta.

### a. *Justificación e importancia.*

Las técnicas de autenticación biométricas basadas en la geometría de la mano emplean un alto nivel de procesamiento de datos; al ser el tipo dato una imagen digital en dos dimensiones. Donde es importante tener en cuenta la posición de la mano y el nivel de expansión de las partes de ésta (dedos y palma) las cuales juegan un papel importante en este proceso.

Los algoritmos de procesamiento de imágenes digitales en dos dimensiones procesan las unidades mínimas que componen una imagen, llamadas pixeles, estos pixeles forman una matriz bidimensional y su descripción está basada en su posición espacial y las componentes de color que estos tengan.

En una imagen con resolución 1024x1024 se configura un número de 1,048,576 pixeles (siendo este un número pequeño en comparación con las resoluciones de las cámaras actuales), y cada una de estas componentes se debe procesar con el fin de obtener la información relevante en nuestro proceso de autenticación de personas, estos procesos son costosos en términos computacionales, pero se puede optimizar su desarrollo aplicando técnicas de computación de procesos paralelos utilizando los múltiples núcleos de las CPU actuales.

#### - *Conveniencia.*

Es posible lograr mayor eficacia en el proceso de autenticación basadas en técnicas biométricas, siendo esta una tendencia, debido a que estas aseguran con un grado de certeza la identidad de la persona que está siendo autenticada. (Justificación Comercial).

- *Relevancia Social.*

Todas las aplicaciones que requieran de autenticación por medio de técnicas biométricas, pueden implementar programación paralela para reducir su tiempo de autenticación. (Justificación Comercial).

Existen propuestas de software para realizar el proceso de autenticación de personas basadas en la geometría de la mano, pero estas técnicas utilizan algoritmos seriales que son muy lentos en el proceso de análisis de las imágenes de entrada. Hoy en día las investigaciones en el campo de computación paralela y análisis de imágenes digitales han llegado a convertirse en tecnologías robustas, esta propuesta pretende desarrollar un mecanismo de software basada en estas tecnologías que permita la autenticación de muchas personas en tiempo real, permitiendo la utilización de hardware básico para la implantación de un futuro sistema de autenticación biométrica. (Justificación Técnica).

- *Implicación Práctica.*

Las investigaciones en el campo de computación paralela aplicadas a la detección de patrones en imágenes dimensionales han crecido mucho en los últimos años existiendo una literatura muy amplia que sirve como base para investigaciones aplicadas. (Justificación Científica).

**b. Descripción de la Solución.**

Se propone el desarrollo de un mecanismo que recibirá como entrada imágenes de manos humanas a las cuales las procesará y extraerá la información pertinente para realizar un proceso de autenticación con

usuarios previamente registrados. La investigación está basada en la mejora de tiempos de ejecución de algoritmos de procesamiento de imágenes utilizando técnicas de construcción de bloques de hebras, específicamente la plataforma Threading building blocks de INTEL, se realizarán las pruebas respectivas comparando tiempos de consumo de algoritmos secuenciales contra los implementados.

**c. Alcances y Limitaciones.**

- Alcances.
  - ✓ La presente tesis propone el desarrollo de un mecanismo que permita preparar la imagen recibida para la obtención de características (fase 2) así como también la extracción de estas en la autenticación de personas (fase 3) en base a imágenes digitales de la geometría de sus manos, para lo cual se propone la utilización de técnicas de procesamiento digital de imágenes y técnicas de computación paralela.
  - ✓ El presente trabajo permitirá la comparación de tiempos entre el uso de tecnología paralela y el uso convencional.
- **Limitaciones.**
  - ✓ Al ser un mecanismo la investigación se basará únicamente en comprobar el modelo propuesto, y no podría ser implantado en un sistema real sin haberse realizado las modificaciones necesarias para convertirlo en un software comercial.
  - ✓ Las imágenes que se utilizaran son solamente las provenientes de una base de datos de un laboratorio totalmente probadas.

- ***Riesgos que se deben afrontar.***

Se han identificado un conjunto de amenazas y vulnerabilidades intrínsecamente ligados a la utilización de tecnologías de reconocimiento basadas en biometría. De las cuales destacan:

- ✓ *Idoneidad de la implantación.* Es necesario un análisis previo para llevar a cabo la implantación de sistemas apropiados.
- ✓ *Alteración de rasgos.* En ocasiones, voluntaria o involuntariamente los rasgos biométricos de las personas pueden verse alterados.
- ✓ *Aceptación cultural.* La idiosincrasia cultural de cada país puede suponer un inconveniente para la implantación de determinados sistemas de reconocimiento.
- ✓ *Desconocimiento.* En ocasiones las entidades que podrían estar interesadas desconocen la calidad o el abanico de productos y utilidades disponibles.
- ✓ *Percepción negativa.* Percepción de ausencia de privacidad por parte de los usuarios.

- ***Viabilidad de la Investigación.***

La investigación es viable, pues se dispone de los recursos necesarios para llevar a cabo. Se cuenta con la base de datos certificada "CASIA Palm print Image Database", obtenida del Centro de Investigación y Seguridad Biométrica del Laboratorio Nacional de Reconocimiento de Patrones como muestra. Así mismo se contara con el diseño de un cluster que permita le evaluación de las

imágenes en bloques para obtener los tiempos exactos al utilizar programación paralela y sin programación paralela.

## FUNDAMENTOS TEÓRICOS

### 2.1 Estado del arte.

Esta tesis está basada en los siguientes trabajos de investigación:

- a. **Peg-Free Hand Geometry Recognition Using Hierarchical Geometry and Shape Matching, A. L.N.Wong, P. Shi (2002) - Proceedings of the IAPR Conference on Machine Vision Applications.**

El objetivo del presente documento es demostrar que es posible la autenticación de personas en base a la geometría de la mano sin el uso de tarugos reconociendo los puntos de inflexión de la mano, teniendo una aceptación del 89%.

- b. **A Fingerprint and Palmprint Recognition Approach Based on Multiple Feature Extraction, R. Gayathri, P. Ramamoorthy (2012) - European Journal of Scientific Research.**

En este artículo de investigación los autores tratan de presentar una técnica biométrica híbrida que sea capaz de lograr un porcentaje de aceptabilidad alto en autenticar correctamente a una persona, cuidando aspectos importantes como el tiempo en el que el sistema demora en autenticar a una persona. El modelo propuesto realiza pre-procesamiento de imágenes, extracción de características, fusión de características y la búsqueda del individuo de acuerdo a este patrón generado.

- c. **Hand geometry personal authentication using machine learning, Mario Karlovčec (2011) - Faculty of Organization and Informatics, University of Zagreb, Croatia.**

En esta investigación el autor propone un sistema de autenticación basado en la geometría de la mano humana, se propone un desempeño óptimo en el tiempo de ejecución del sistema así como un alto índice de aciertos en la verificación de las identidades de los sujetos que son autenticados, la investigación utiliza técnicas de máquinas de aprendizaje en particular el algoritmo de los K vecinos más próximos, la investigación muestra buenos resultados al utilizar esta técnica sobre diversas técnicas basadas en distancias simples.

- d. **Feature selection for hand-geometry based person authentication, Vandana Roy and C. V. Jawahar (2008), Centre for Visual Information Technology Gachibowli, Hyderabad, India.**

Los autores de esta investigación se basan en que la utilización de la técnica biométrica de la geometría de la mano es de fácil utilización ya que no se necesita de potentes escáneres, como en el caso del reconocimiento basado en iris, o de ambientes con poco ruido, como es el caso del reconocimiento por voz, o de imágenes con una gran resolución, como es el caso de autenticación facial, para realizar una autenticación con altos índices de aceptabilidad en sistemas comerciales, sin embargo ellos postulan que no se utilizan las mejores características para realizar una buena discriminación en la búsqueda de la identidad a autenticar, ellos proponen utilizar una técnica de

análisis discriminante no paramétrico para poder obtener un espacio de características realmente separable.

**e. Hand geometry based verification, Richard L. Zunkel (2008), Recognition Systems Inc Campbell, CA.**

El autor de este artículo se basa en que las dimensiones físicas de la mano humana contienen información que es capaz de autenticar la identidad de un individuo, esta información es conocida como la geometría de la mano. Se hace un estudio de los diversos sistemas comerciales que utilizan esta técnica de autenticación de personas, sobre la aplicabilidad y lo más importante el desempeño en cuanto al tiempo de procesamiento utilizado.

**f. Mecanismos de autenticación biométrica mediante tarjeta inteligente, RAÚL SÁNCHEZ REILLO (2010), tesis de doctorado Universidad Politécnica De Madrid.**

El objetivo fundamental de esta tesis de doctorado es la creación de nuevos Mecanismos de Autenticación del titular de una Tarjeta inteligente. La motivación de dicho trabajo ha sido la continua proliferación de nuevas aplicaciones donde es necesaria una autenticación del usuario que las utiliza y, por otro lado, las nuevas posibilidades que se abren dentro de la tecnología de dicho tipo de tarjetas de identificación. El sistema de autenticación, está basado en la Identificación Biométrica, realizando la comprobación de la identidad del titular dentro de la propia tarjeta. Para ello se han estudiado distintas técnicas biométricas de forma que se puedan extraer conclusiones que lleven a la definición de un modelo para el nuevo Sistema de

Autenticación. Las técnicas escogidas para ser analizadas fueron: la de Reconocimiento de Locutores, es decir, basada en la voz; la Geometría del Contorno de la Mano; y el análisis del Patrón de Iris Ocular.

**g. A prototype hand geometry based verification system , Anil K. Jain & Arun Ross (2007), Michigan State University**

En esta investigación los autores se basan en la literatura existente para proponer la creación de un sistema que permita lograr la autenticación de identidades de usuarios en un sistema basándose en la geometría de la mano, basándose en alineadores físicos e imágenes de entrada en dos dimensiones en escala de grises.

## **2.2 Bases teóricas de la investigación.**

Las bases teóricas de la investigación se centraran en:

- Procesamiento de imágenes de dos dimensiones en formato RGB
- Programación paralela basada en bloques de construcción de hebras.
- Técnicas de clasificación de datos biométricos.

## **METODOLOGÍAS A EMPLEAR**

### **3.1 Tipo y nivel de la investigación.**

#### **a. Tipo de Investigación.**

Esta tesis, es una **investigación aplicada**, ya que se tiene como finalidad primordial el desarrollo de un mecanismo para la autenticación de personas basada en la geometría de la mano aplicando técnicas de procesamiento digital de imágenes y técnicas de computación paralela.

- **Tipo de diseño.** Experimental. Se utiliza la base de datos certificada CASIA Multi-SpectralPalprintImageDatabase V1.0 entregada por el centro de búsqueda de seguridad en biometría del *laboratorio*

*Nacional del instituto del reconocimiento de patrones de automatización, de la academia de ciencias de China. Para las pruebas respectivas.*

- **Prolongación.** Transversal o sincrónico.
- **Énfasis en la naturaleza de los datos.** Cualitativa

**b. Nivel de la investigación.**

Teniendo en consideración la naturaleza, así como el estado del conocimiento acerca del presente tema de investigación, se han determinado los siguientes alcances:

**Alcance Descriptivo.**

Considera la especificación de las propiedades, características y los diferentes perfiles para metodología de autenticación eficaz y eficiente.

**Alcance Correlacionado.**

Abarca la evaluación y cuantificación del grado de vinculación entre las variables determinadas.

**Alcance Explicativo.**

Involucra y explica los resultados obtenidos, así como su variación frente a diversos cambios de parámetros tales como iluminación, posición, espectros y otros.

### 3.2 Diseño de la investigación.

#### a. *Formulación de la hipótesis:*

##### - *Hipótesis Descriptiva.*

El uso de una configuración de bloques de hilos de software reduce considerablemente el tiempo de reconocimiento de la biometría de la mano. Respecto de una configuración tradicional, como la programación en serie.

##### - *Hipótesis Correlaciones.*

A mayor cantidad de memoria principal y número de núcleos físicos de la unidad central de procesamiento tiempo de reconocimiento de la biometría de la mano es menor.

A mayor segmentación de la imagen de la biometría de la mano el margen de error es menor.

A mayor segmentación de la imagen de la biometría de la mano la pérdida de detalles es mayor afectando la autenticación individual.

##### - *Hipótesis De La Diferencia Entre Grupos.*

El resultado de la toma de muestra puede verse afectado por el tipo de espectro de luz empleado.

Al momento de la primera segmentación el grado de iluminación óptimo es del 50%.

#### b. *Diseño de las variables:*

El diseño de la presente investigación tiene como base el conjunto de variables Independientes y dependientes, provenientes de la formulación de las hipótesis.

**h1:** Al usar Configuración de Bloques de hilos de software (TBB) para el procesamiento de las imágenes de entrada se *reduce* el tiempo de reconocimiento de la biometría de la mano.

- *VI (Variable Independiente)*. Uso de la librería TBB en el procesamiento de las imágenes de entrada.
- *VD (Variable Dependiente)*. Tiempo de reconocimiento de la biometría de la mano

**h2:** A mayor cantidad de memoria principal y número de núcleos físicos de la unidad central de procesamiento menor tiempo de reconocimiento de la biometría de la mano.

- *VI (Variable Independiente)*. Memoria principal y número de núcleos físicos de la unidad central de procesamiento.
- *VD (Variable Dependiente)*. Tiempo de reconocimiento de la biometría de la mano.

**h3:** A mayor segmentación de la imagen de la biometría de la mano menor margen de error.

**h4:** A mayor segmentación de la imagen de la biometría de la mano mayor pérdida de detalles.

- *VI (Variable Independiente)*. Segmentación de la imagen de la biometría de la mano.
- *VD (Variable Dependiente)*. Margen de error.
- *VD (Variable Dependiente)*. Detalles de la biometría de mano.

**h4.** El resultado de la toma de muestra puede verse afectado por el tipo de espectro de luz empleado.

- *VI (Variable Independiente)*. El espectro de luz empleado en la toma de imagen.

**h5.** Al momento de la primera segmentación el grado de iluminación óptimo es del 50%.

- *VI (Variable Independiente)*. Grado de iluminación en la primera segmentación.

La Tabla 1 contiene un resumen de las variables citadas. Donde La evaluación de cada una de las variables independientes se va medir mediante un aplicativo, el cual se desarrolla en el transcurso de la presente investigación.

Tabla 1

*Relación De Las Variables Independientes Con Las Variables Dependientes*

<b>Variables Independientes</b>	<b>Variables Dependientes</b>
Uso de la librería TBB en el procesamiento de las imágenes de entrada.	Tiempo de reconocimiento de la biometría de la mano
Memoria principal y número de núcleos físicos de la unidad central de procesamiento.	Tiempo de reconocimiento de la biometría de la mano
Segmentación de la imagen de la biometría de la mano.	Margen de error. Detalles de la biometría de mano.
Espectro de luz empleado en la toma de imagen.	
Grado de iluminación al momento de la segmentación.	

(elaborador propia)

**c. *Tipologías del diseño de la investigación.***

La investigación es de clase “Experimentos Puros” definida en la tipología de Campbell y Stanley (1966).

Las imágenes de las palmas de las manos se pueden evaluar como grupo de control y a la vez grupo experimental logrando tener puntos de comparación y equivalencia entre los grupos.

**d. *Contexto de la investigación.***

La investigación se toma en un contexto de “Experimento De Campos”.

Se evalúa la mano izquierda de 100 individuos tomada en 2 sesiones con una distancia de más de un mes entre cada sesión.

**3.3 Universo o Población y Muestra.**

**a. *Unidad de análisis.***

El presente proyecto formula una propuesta para el desarrollo de un mecanismo que utilice técnicas de computación paralelas basadas en la utilización de bloques de construcción de hebras de software y procesamiento de imágenes digitales, para permitir la autenticación de múltiples personas que desean validar su acceso.

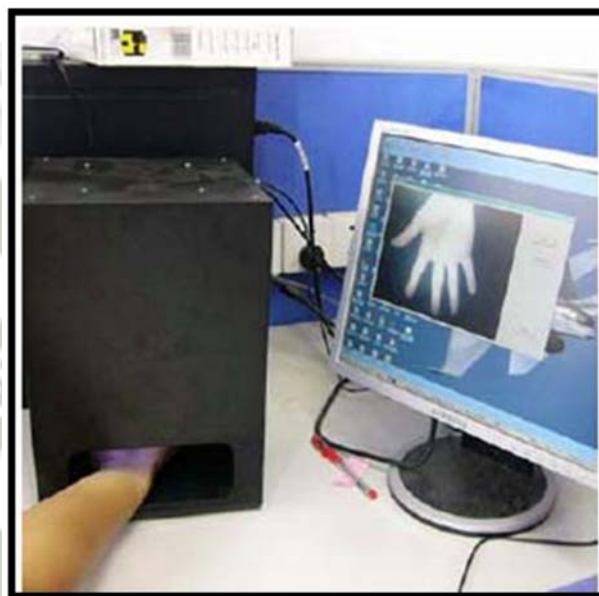
Por lo que se considera como unidad de análisis a:

Muestra de manos izquierdas de personas de ambos sexos que serán evaluados con y sin técnicas de computación paralelas.

**b. *Población.***

Las imágenes utilizadas en este mecanismo son tomadas de la base de datos “CASIA Palm print Image Database”, obtenida del Centro de Investigación y Seguridad Biométrica del Laboratorio Nacional de Reconocimiento de Patrones Del Instituto de Automatización de la

Academia de Ciencias De China, esta base de datos contiene 7200 imágenes palmares capturadas de 100 sujetos. Para cada sujeto, se recogen imágenes palmares de las manos izquierda y derecha. Todas las imágenes de impresiones palmares son de archivos JPEG, recogidas con un dispositivo de reconocimiento palmar, tal como se muestra en la Figura 1.



*Figura 1.* Modelo de Obtención de imágenes de CASIA

En el dispositivo, no hay clavijas para restringir las posturas y las posiciones de las manos. Los sujetos solo están obligados a poner su mano en el dispositivo y esta imagen es tomada sobre un fondo de color uniforme. El dispositivo suministra una iluminación distribuida y captura imágenes de la huella palmar utilizando una cámara CMOS fija en la parte inferior del dispositivo. Seis imágenes palmares en la base de datos se muestran en la figura 2.

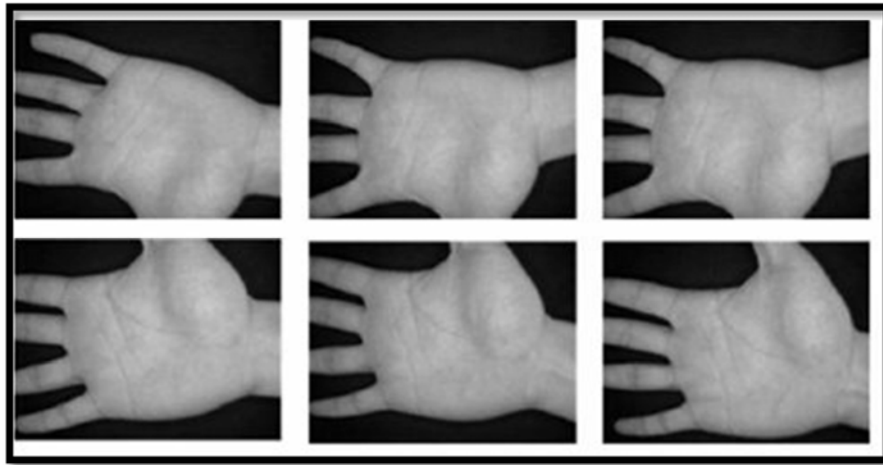


Figura 2. Ejemplos de palmas

Las siguientes seis imágenes de la Figura 3 son tomadas por el mismo dispositivo pero muestran la misma imagen tomada en los seis diferentes espectros.

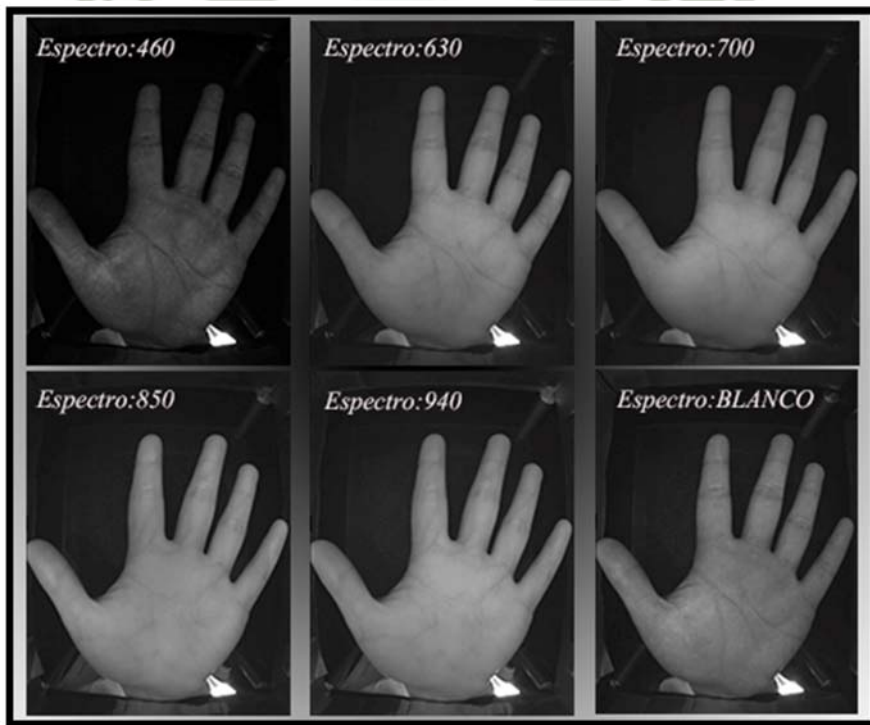


Figura 3. Muestra de los diferentes espectros de la base de datos Casia

**c. Tipo de muestra.**

Este tipo de estudio donde se hace una asociación entre variables y resultados y cuyos resultados servirán de base para tomar decisiones, se da por medio de una muestra probabilística, diseñada de tal manera que los datos lleguen a ser generalizados a la población.

**d. Criterios de inclusión.**

- Todas las muestras son obtenidas de la misma base de datos
- La población la constituye un total de 3600 imágenes de manos izquierdas tomadas de CASIA Multi-Spectral la cual contiene 7.200 imágenes capturadas de ambas manos izquierda y derecha, de 100 personas diferentes.

La captura de imágenes por cada mano se realizó en dos sesiones diferentes con intervalo de tiempo de más de un mes entre cada sesión.

En cada sesión, se tomaron tres muestras y cada muestra contiene seis imágenes de la mano que son capturadas al mismo tiempo con seis espectros electromagnéticos diferentes, entre 460 nm y 940nm incluida la luz blanca respectivamente.

**3.4 Técnicas e Instrumentos.**

El recurso de imágenes ha sido tomado de El *Laboratorio Nacional Del Instituto De Reconocimiento De Patrones De Automatización, De La Academia De Ciencias De China.*

En el presente trabajo para el análisis de las imágenes digitales se propone técnicas de computación paralela en vez de las clásicas técnicas seriales.

Para ello se ha diseñado un sistema de vectorización de la palma de la

mano (palmar), con la finalidad de almacenar valores vectoriales en vez de almacenar imágenes de comparación que ocupan un mayor espacio en el disco y con la consecuente ventaja de velocidad debido a que la comparación entre valores numéricos es más rápida que la comparación entre imágenes.

Para el desarrollo del presente trabajo se han empleado aparatos e instrumentos de diferente naturaleza:

Instrumentos de Hardware:

1. Aparatos de iluminación de longitud de onda determinada.
2. Cámara fotográfica digital CCD

Instrumentos de Software:

3. Visual Studio 2010 Profesional C++
4. Visual Studio 2010 Profesional C#.
5. Librerías OpenCV, ImageRGB

### **3.5 Procedimientos de recolección de datos.**

El recurso de imágenes ha sido tomado de El *Laboratorio Nacional Del Instituto De Reconocimiento De Patrones De Automatización, De La Academia De Ciencias De China.*

### **3.6 Plan de análisis de datos**

La manipulación de las variables independientes para analizar las consecuencias en las variables dependientes. En la Tabla 2 Se muestra los grados de manipulación de las variables independientes.

Tabla 2  
*Grados de Manipulación de las Variables Independientes*

<b>Variabes Independientes</b>	<b>Grado de Manipulación</b>	<b>Modalidad</b>
Uso de la librería TBB en el procesamiento de las imágenes de entrada.	Presencia - Ausencia	La misma Base de datos será evaluada como grupo experimental así como de control.
Memoria principal y número de núcleos físicos de la unidad central de procesamiento.	Más de dos grados	La aplicación va a ser probada en equipos con diferentes características
Segmentación de la imagen de la biometría de la mano.	Modalidad de manipulación	Aplicación parametrizada que permite regular el número de segmentaciones.
Espectro de luz empleado en la toma de imagen.	Más de dos grados	La base de datos tiene las mismas imágenes tomadas desde diferentes espectros.
Grado de iluminación al momento de la segmentación.	Modalidad de manipulación	Aplicación parametrizada que permite regular el grado de iluminación.

(elaborador propia)

**PLAN DE TRABAJO.**

NOMBRE DE LA TAREA	DURACION
<b>Formulación del problema</b>	<b>3 días</b>
<b>Determinación de objetivos</b>	<b>3 días</b>
<b>Elaboración del plan de tesis</b>	<b>7 días</b>
<b>Presentación del plan de tesis</b>	<b>7 días</b>
<b>Investigación teórica: cap. i</b>	<b>7 días</b>
<b>Implementación del mecanismo</b>	<b>7 días</b>
<b>Comprobación del nuevo modelo</b>	<b>7 días</b>
<b>Conclusiones y recomendaciones</b>	<b>7 días</b>
<b>Pruebas de evaluación</b>	<b>7 días</b>
<b>Redacción, revisión y depuración</b>	<b>3 días</b>
<b>Impresión y entrega de ejemplares</b>	<b>2 días</b>

## REFERENCIAS BIBLIOGRÁFICAS.

- ✓ A Fingerprint and Palmprint Recognition Approach Based on Multiple Feature Extraction, R. Gayathri, P. Ramamoorthy (2012) - European Journal of Scientific Research.
- ✓ Hand geometry personal authentication using machine learning, Mario Karlovčec (2011) - Faculty of Organization and Informatics, University of Zagreb, Croatia.
- ✓ Feature selection for hand-geometry based person authentication, Vandana Roy and C. V. Jawahar (2008), Centre for Visual Information Technology Gachibowli, Hyderabad, India.
- ✓ Hand geometry based verification, Richard L. Zunkel (2008), Recognition Systems Inc Campbell, CA.e)Mecanismos de autenticación biométrica mediante tarjeta inteligente, RAÚL SÁNCHEZ REILLO (2010), tesis de doctorado Universidad Politécnica De Madrid.
- ✓ A prototype hand geometry based verification system , Anil K. Jain & Arun Ross (2007), Michigan State University
- ✓ Duda, R. O., Hart, P. E., e Stork, D. G. (2001). Pattern Classification. John Wiley & Sons.
- ✓ Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8):882–891.
- ✓ Gonzales, R. e Woods, R. (2002). Digital Image Processing. Prentice–Hall, Upper Saddle River, New Jersey.
- ✓ Sonka, M., Hlavac, V., e Boyle, R. (1999). Image Processing, Analysis and Machine Vision. SPIE.
- ✓ Theodoridis, S. e Koutroumbas, K. (1998). Pattern Recognition. Academic Press.
- ✓ Shirley, M. (1975). Enzyme variation in Eimeria species of the chicken. Parasitology, 71(3):369–376.

- ✓ Welsh, J. e McClelland, M. (1990). Fingerprinting genomes using PCR with arbitrary primers. *Nucleic Acids Research*, 18(24):7213–7218.
- ✓ Williams, R. (1998). Epidemiological aspects of the use of live anticoccidial vaccines for chickens. *International Journal of Parasitology*, 28(7):1089–1098.



## POSIBLE TEMARIO DEL INFORME FINAL

- ✓ Resumen
- ✓ Introducción
- ✓ Justificación
- ✓ Cap. 1. Generalidades
- ✓ Cap. 2. Marco Teórico
- ✓ Cap. 3. Propuesta
- ✓ Cap. 4. Caso de Estudio
- ✓ Cap. 5. Evaluación
- ✓ Conclusiones y recomendaciones
- ✓ Bibliografía
- ✓ Anexos



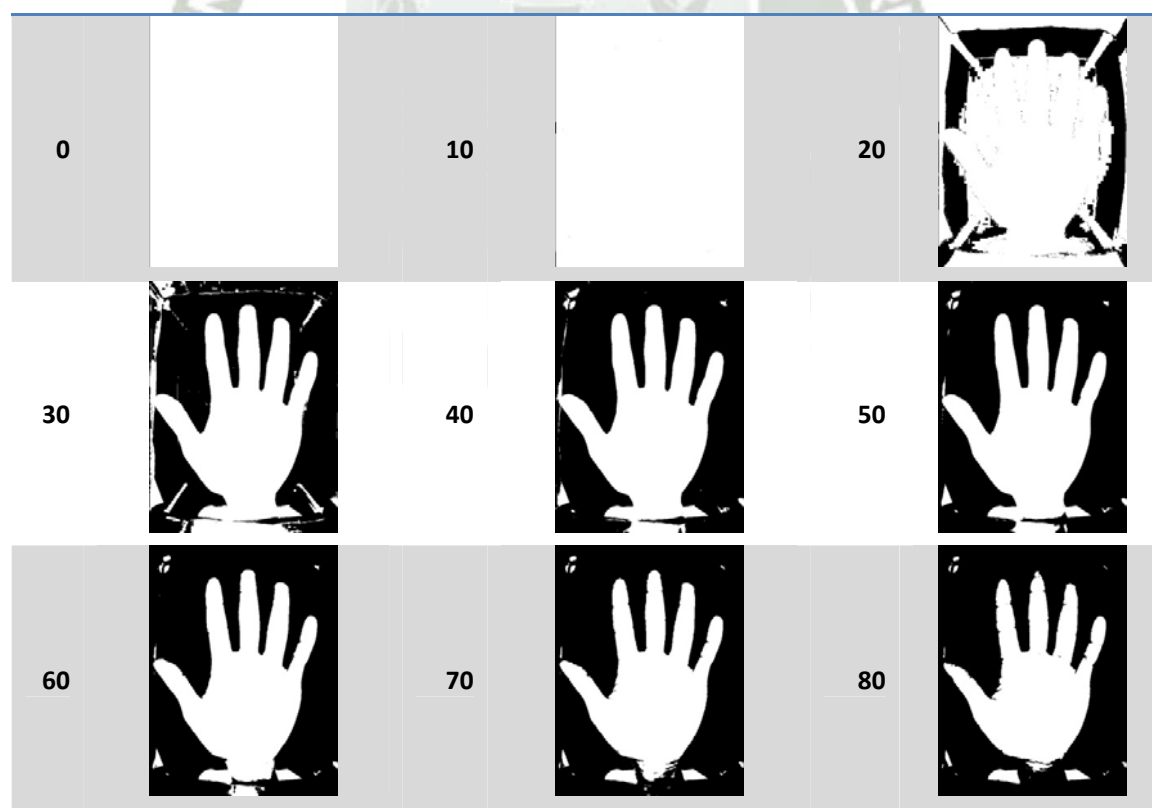
## ANEXO B

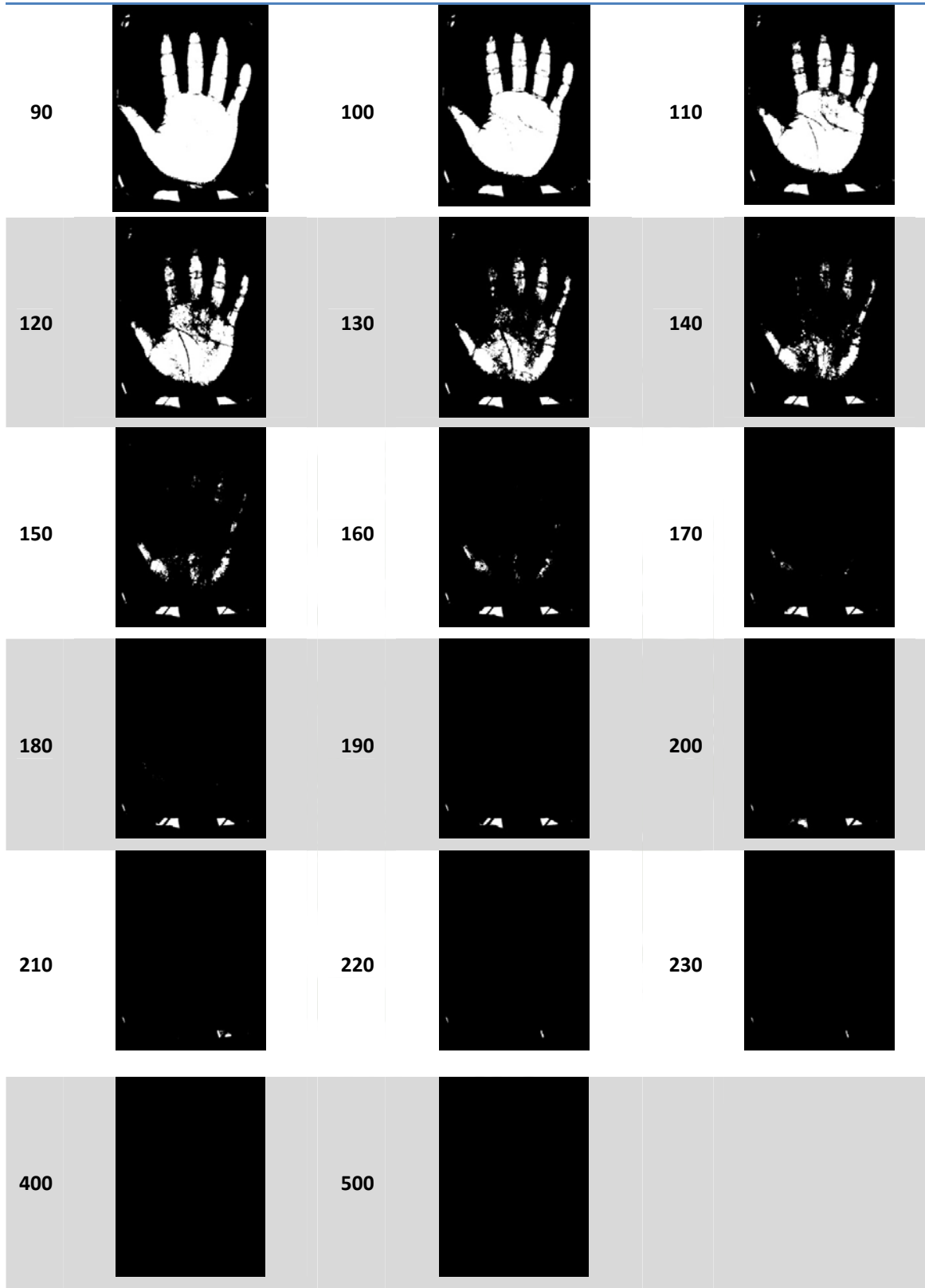
## UMBRAL DE ILUMINACION

Para poder evaluar la palma de la mano es necesario convertir la imagen de escala de grises a una imagen en blanco y negro para lo cual es necesario determinar el umbral correcto de iluminación para separar la palma del fondo de la imagen.

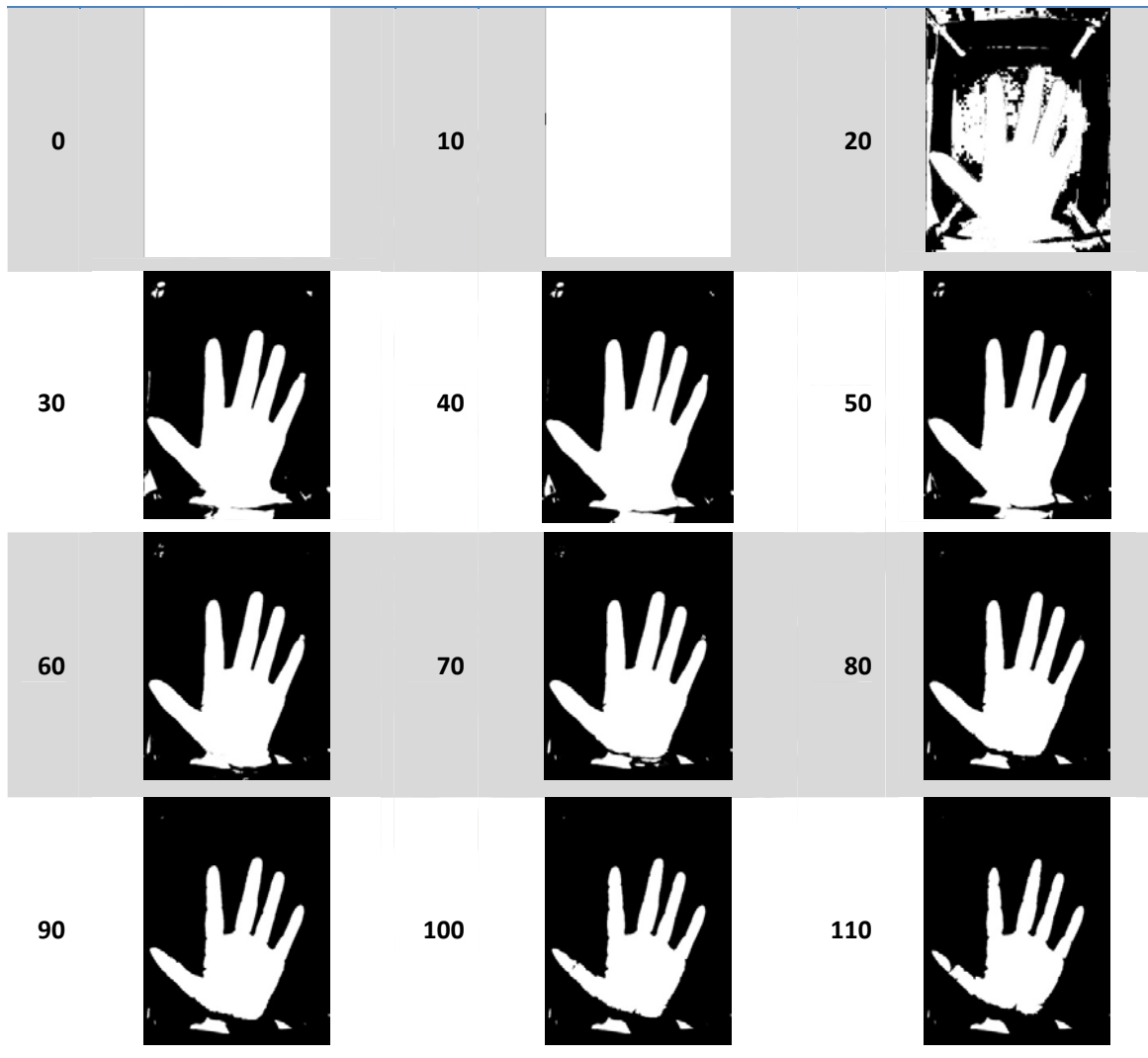
Al recoger la imagen a evaluar esta se ve de la siguiente manera dependiendo del espectro utilizado.

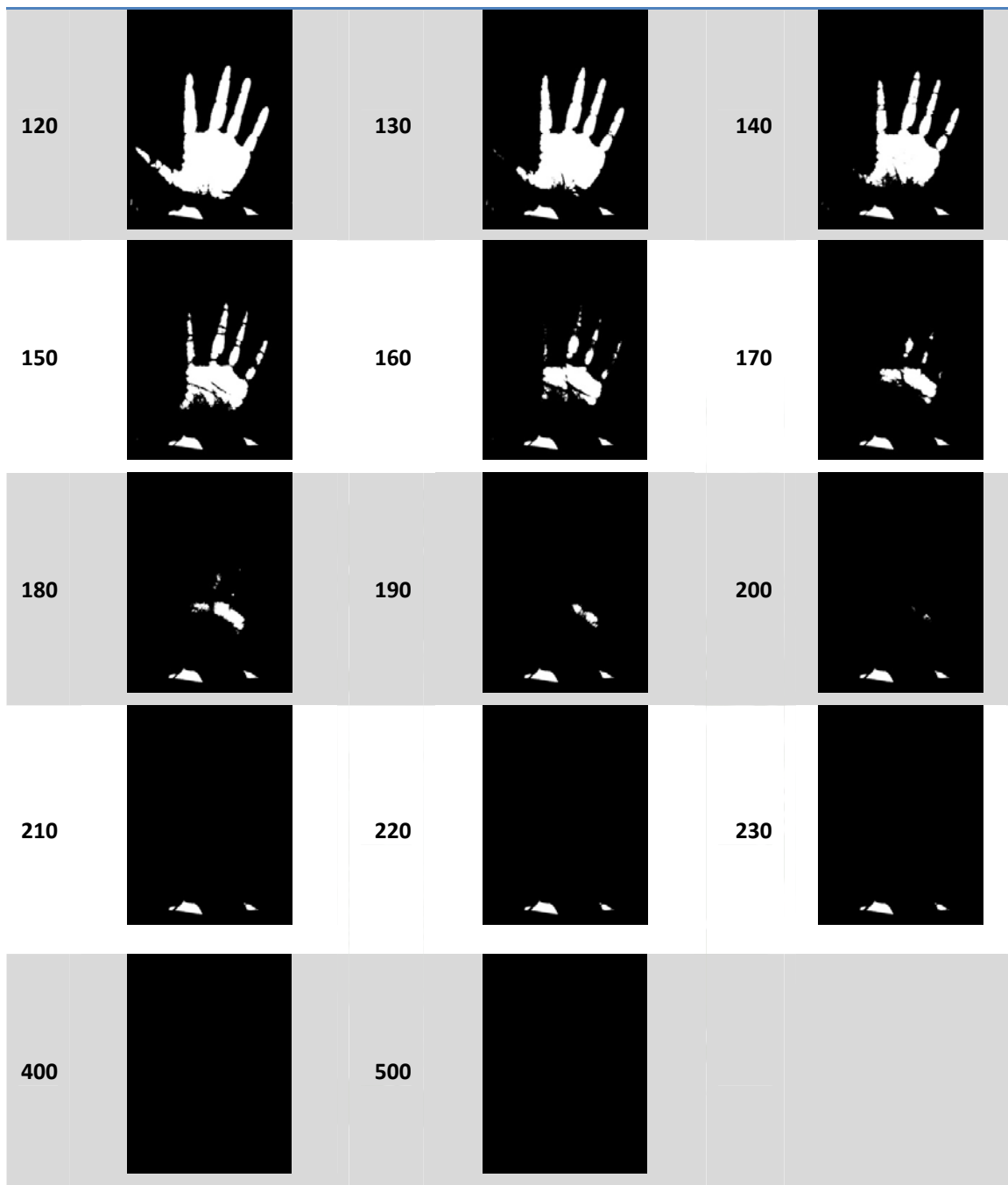
## 1. ESPECTRO 460.



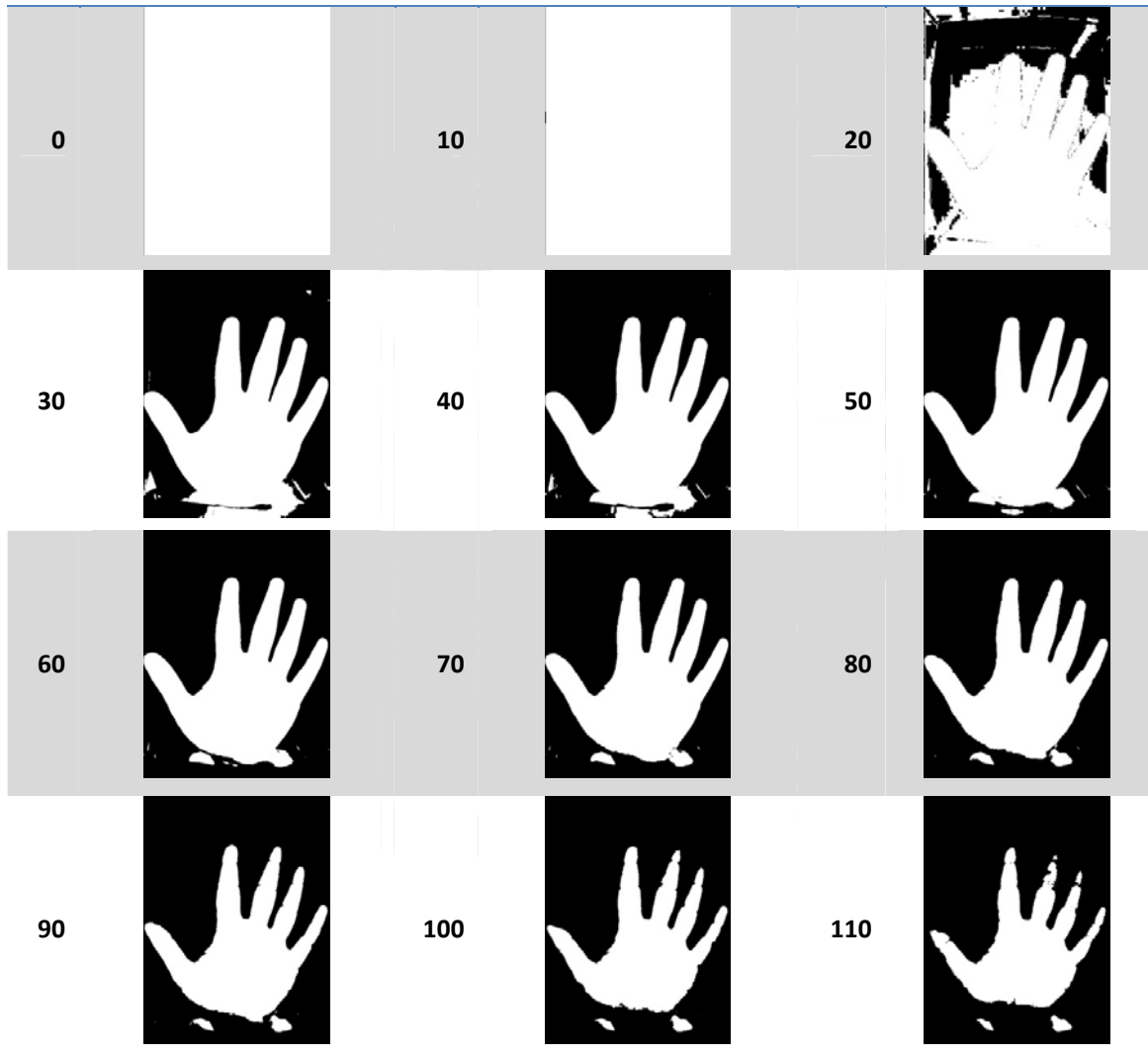


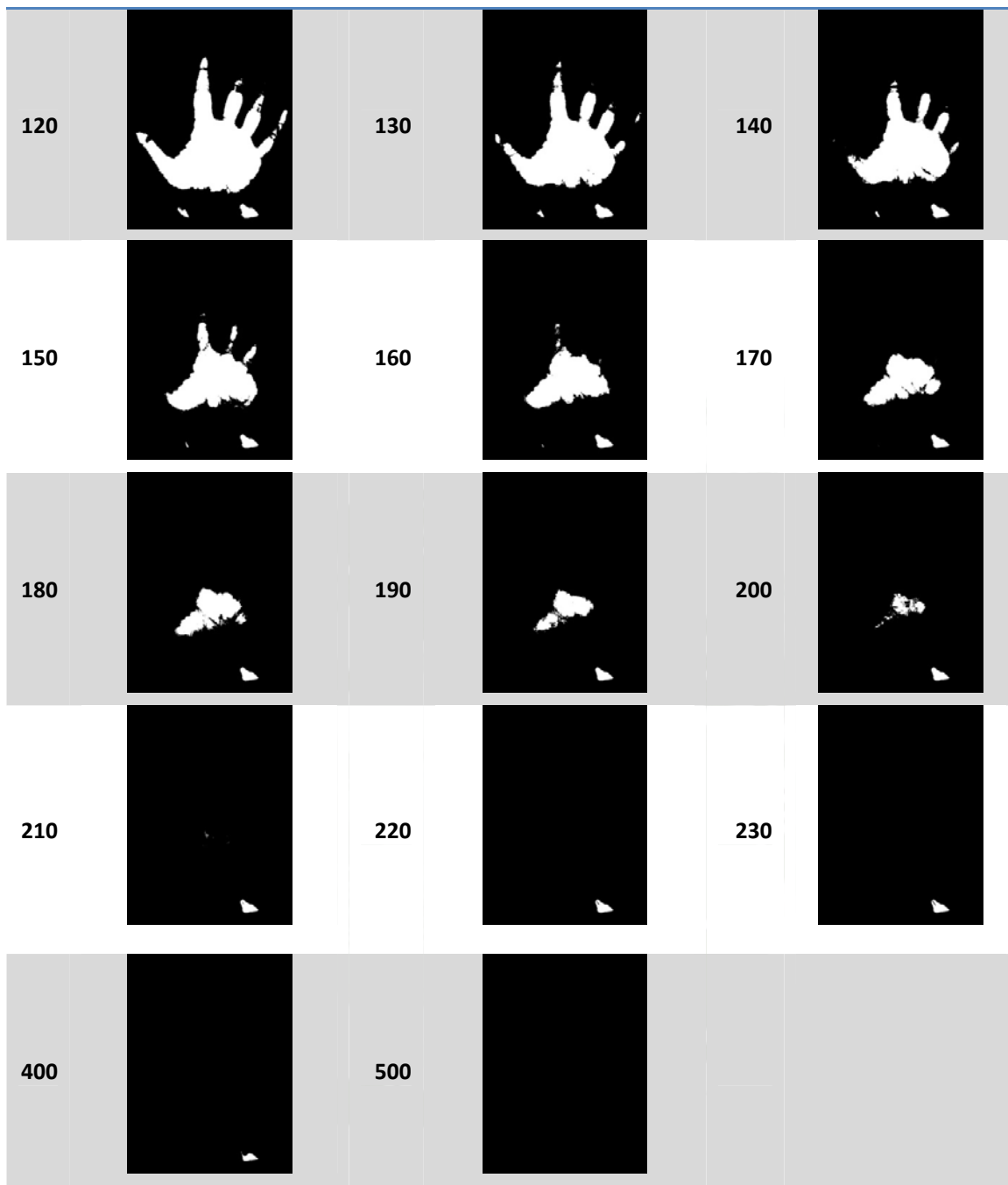
2. ESPECTRO 630.



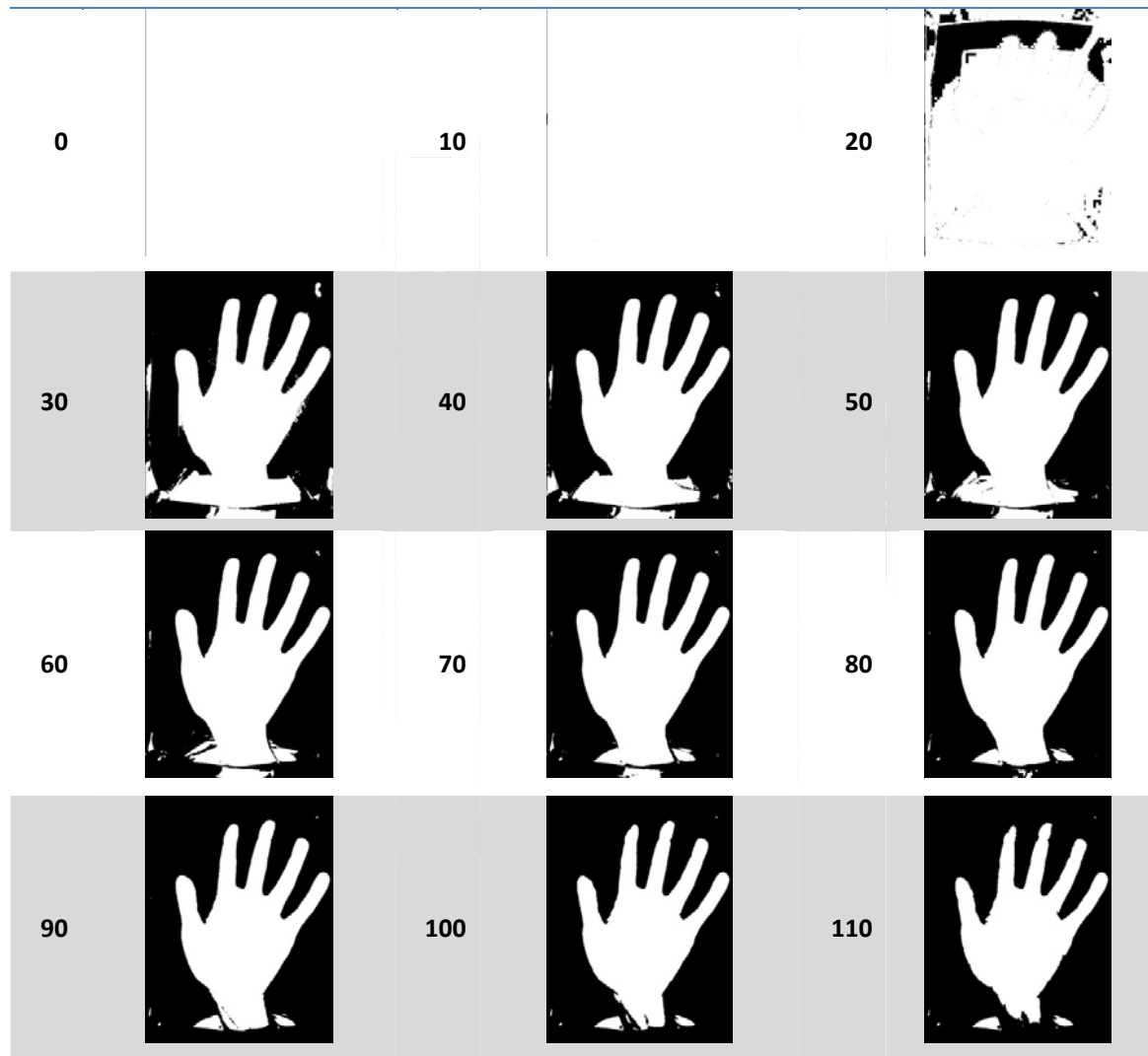


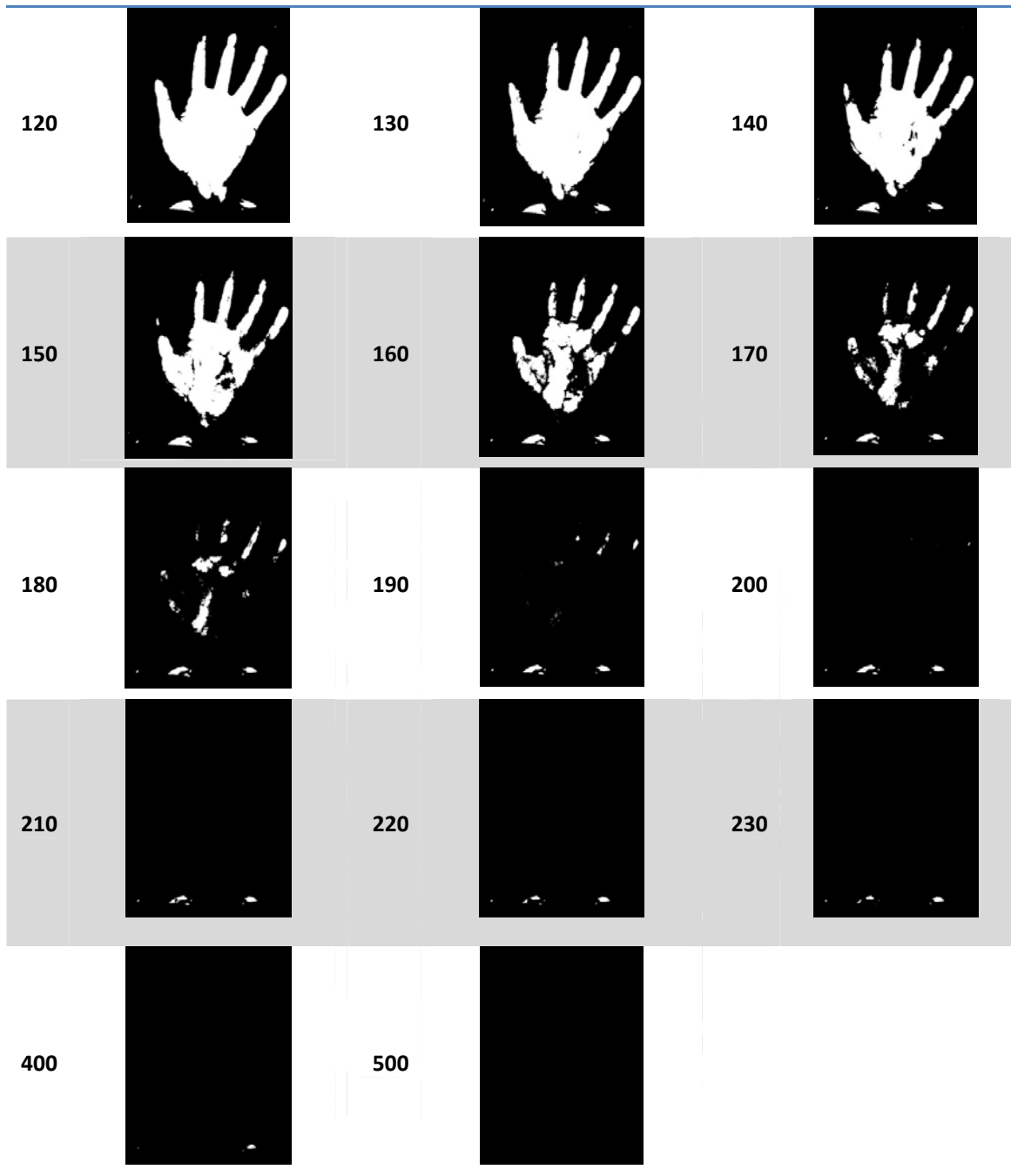
3. ESPECTRO 700.



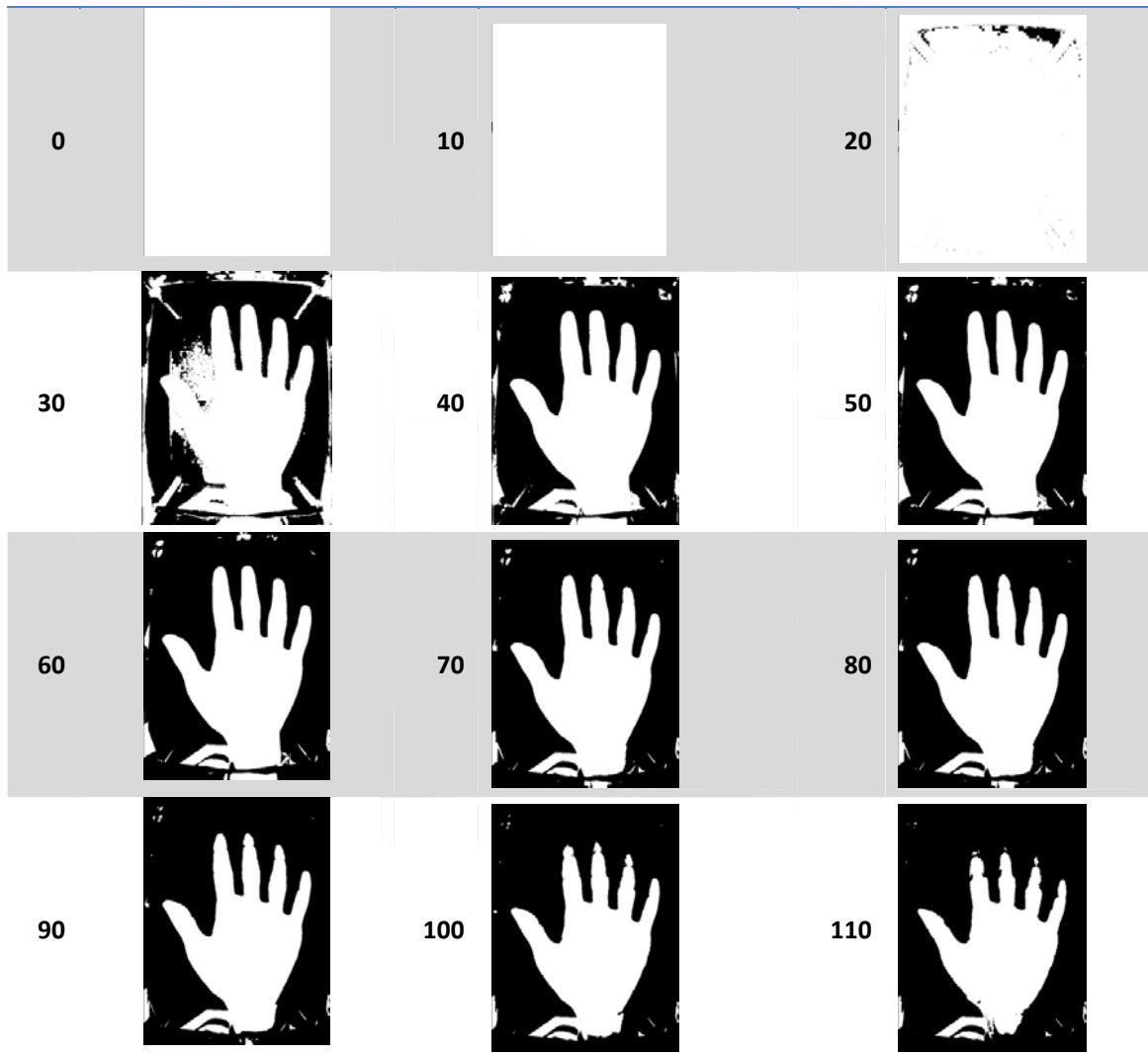


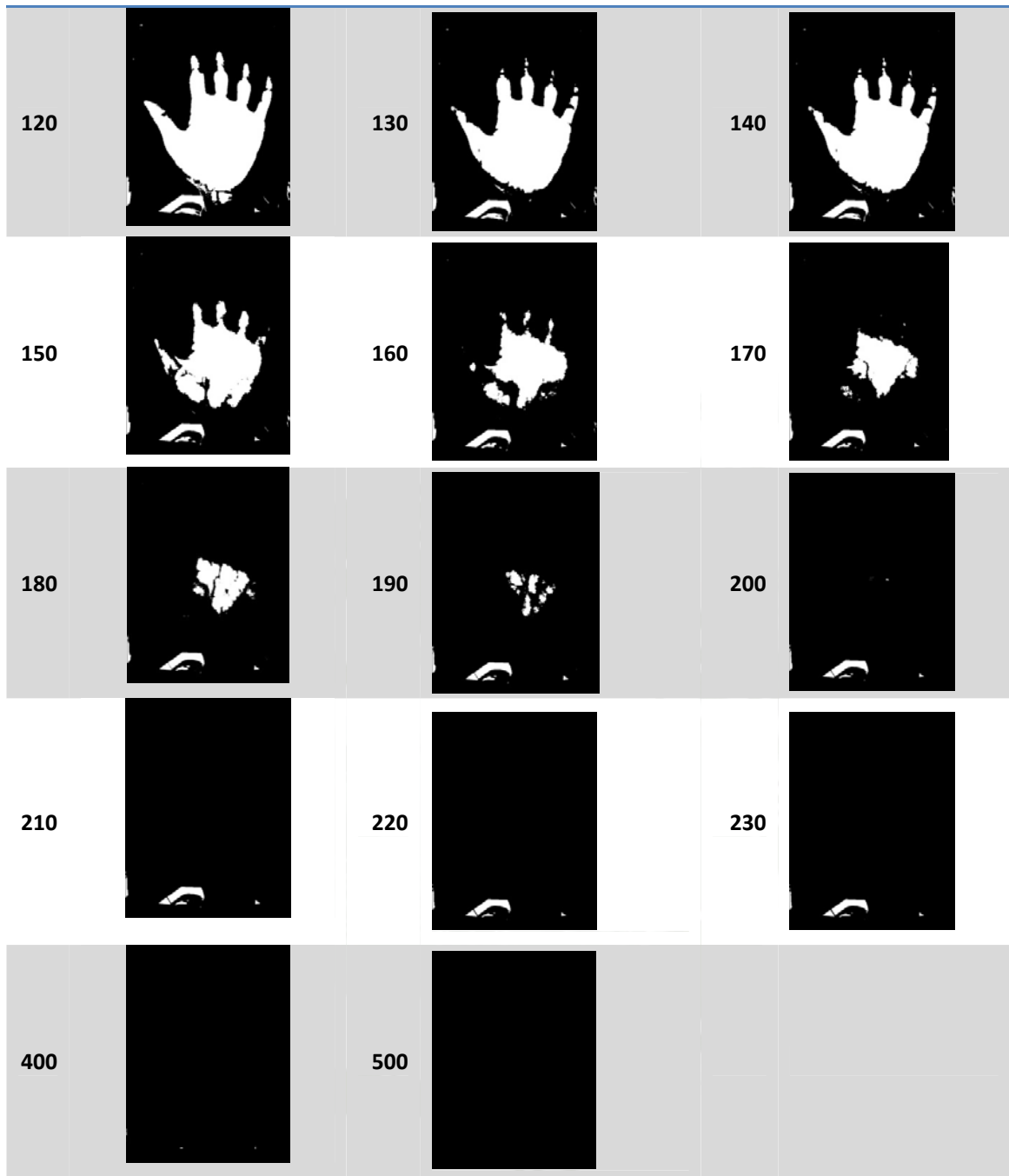
4. ESPECTRO 850.



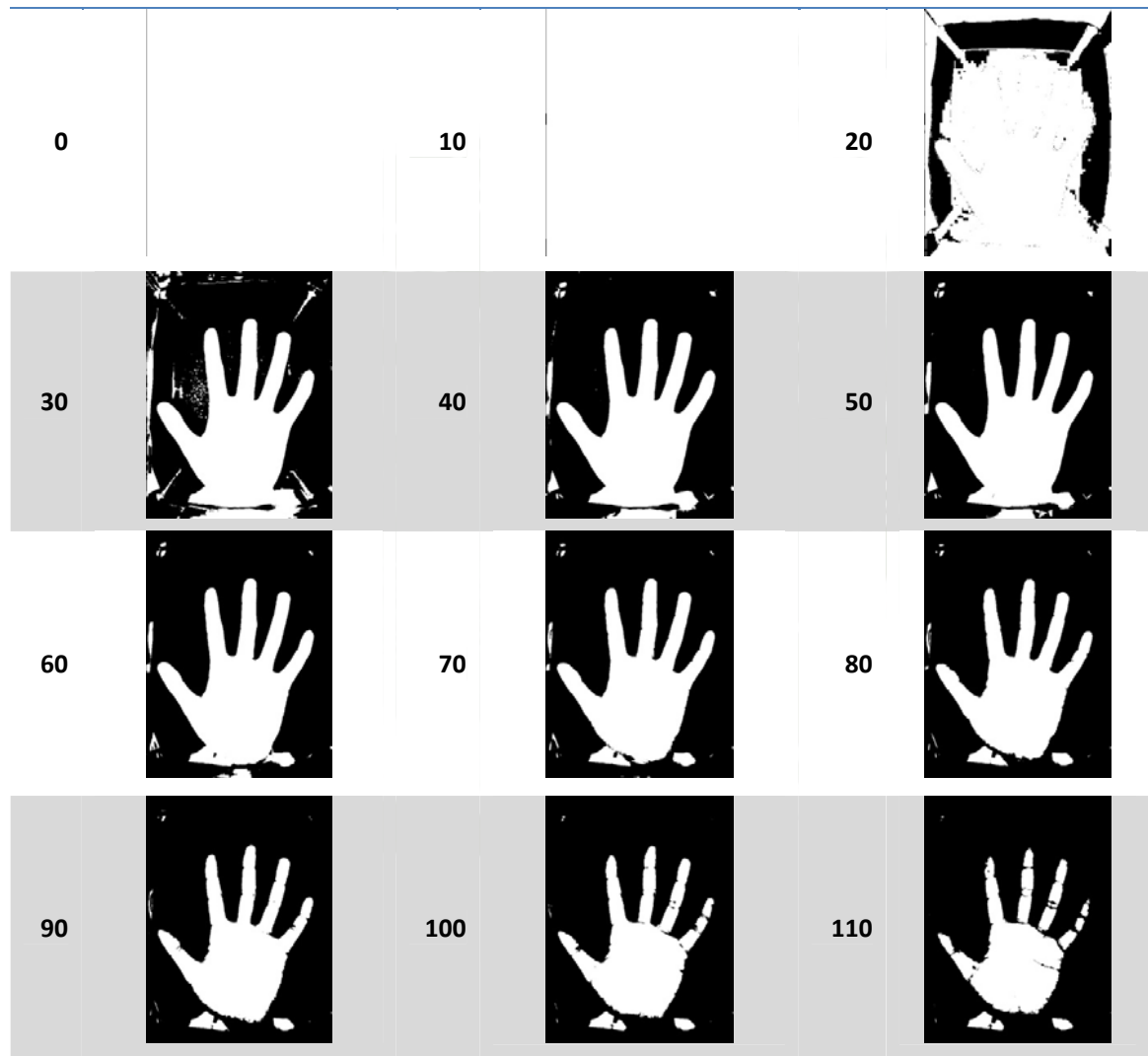


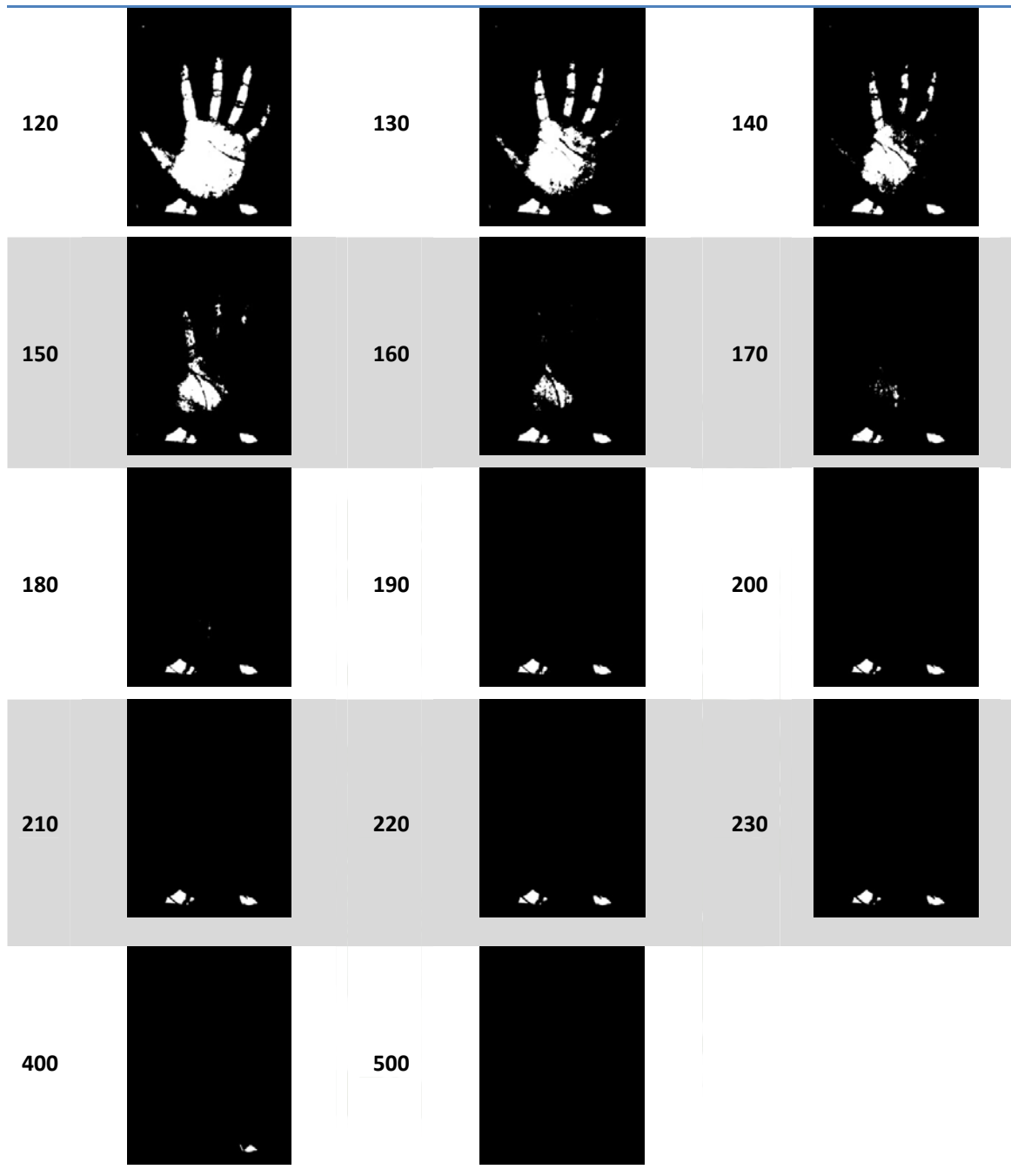
5. ESPECTRO 940.





6. ESPECTRO BLANCO.





### ANEXO C

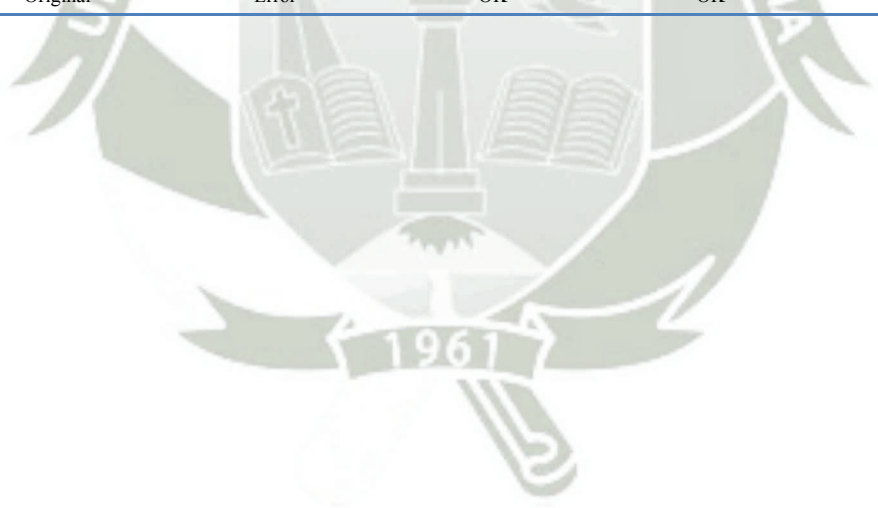
#### SEGMENTACION DE LA IMAGEN

Una vez que la palma está a dos colores es necesario extraer su contorno, por lo que es necesario tener una imagen homogénea.

#### 7. ESPECTRO 460.



Ilum		0	30	50	100	
30						
	Original	Error	Error	Error	Error	
	40					
		Original	Error	Error	OK	OK
		50				
Original			Error	OK	OK	OK



8. ESPECTRO 630.



Ilum		0	30	50	100
30					
	Original	OK	OK	OK	OK
40					
	Original	Error	OK	OK	OK
50					
	Original	OK	OK	OK	OK
60					
	Original	OK	OK	OK	OK

70					
	Original	Error	OK	OK	OK
80					
	Original	Error	OK	OK	OK
90					
	Original	Error	Error	Error	Error
100					
	Original	Error	Error	Error	Error

9. ESPECTRO 700.



Ilum		0	30	50	100
30					
	Original	Error	Error	Error	Error
40					
	Original	Error	Error	Error	Error
50					
	Original	Error	Error	Error	Error
60					
	Original	Error	Error	Error	Error

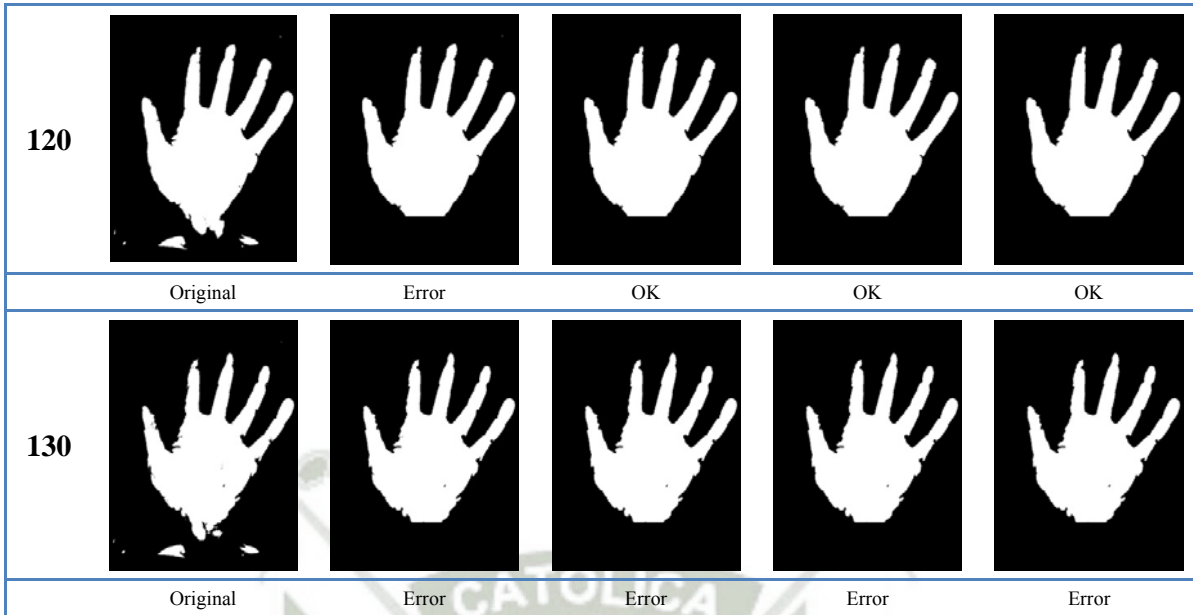
70					
	Original	Error	OK	OK	OK
80					
	Original	Error	Error	Error	Error
90					
	Original	Error	Error	Error	Error

10. ESPECTRO 850.



Ilum.		10	20	30	40
30					
	Original	Error	OK	OK	OK
40					
	Original	Error	OK	OK	OK
50					
	Original	Error	OK	OK	OK
60					
	Original	Error	OK	OK	OK

	Original	Error	OK	OK	OK
70					
	Original	OK	OK	OK	OK
80					
	Original	OK	OK	OK	OK
90					
	Original	Error	OK	OK	OK
100					
	Original	Error	OK	OK	OK
110					
	Original	OK	OK	OK	OK



11. ESPECTRO 940.



Illum .		0	30	50	100
40					
	Original	Error	OK	OK	OK
50					
	Original	Error	OK	OK	OK
60					
	Original	Error	OK	OK	OK
70					
	Original	Error	OK	OK	OK

	Original	Error	OK	OK	OK
80					
	Original	OK	OK	OK	OK
90					
	Original	Error	OK	OK	OK



### 12. ESPECTRO BLANCO.



Ilum.		0	30	50	100	
30						
	Original	Error	OK	OK	OK	
	40					
		Original	OK	OK	OK	OK
		50				
Original			Ok	OK	OK	OK
60						
	Original		Error	OK	OK	OK



## ANEXO D

## TABLAS RESUMEN DE LA EVALUACIÓN DE LA ILUMINACIÓN SEGÚN SU ESPECTRO

Tabla 1.

Cantidad de Errores generados en la combinación Espectro-Iluminación-Segmentación( donde la letra x representa la existencia de palmas irreconocible, y las casillas escritas en rojo muestran la mejor combinación con la menor cantidad de errores por espectro.)

Espectro	Iluminación	Segmentación			
		0	30	50	100
460	30	586	x	x	x
460	40	381	x	187	182
<b>460</b>	<b>50</b>	<b>296</b>	<b>79</b>	<b>79</b>	<b>80</b>
460	60	398	135	133	134
460	70	498	323	237	235
460	80	559	381	379	379
630	30	480	x	x	158
630	40	222	67	110	67
<b>630</b>	<b>50</b>	<b>219</b>	<b>62</b>	<b>62</b>	<b>62</b>
630	60	235	74	74	73
630	70	310	98	99	98
630	80	373	50	152	153
630	90	408	196	192	191
630	100	471	284	290	289
700	30	307	112	110	110
700	40	198	57	57	57
<b>700</b>	<b>50</b>	<b>199</b>	<b>54</b>	<b>54</b>	<b>54</b>
700	60	239	57	58	55
700	70	288	68	68	66
700	80	342	87	88	86
700	90	374	130	130	132
850	30	596	x	567	567
850	40	437	279	280	279
850	50	269	141	140	140
850	60	252	128	130	130
850	70	255	109	110	109
<b>850</b>	<b>80</b>	<b>245</b>	<b>88</b>	<b>88</b>	<b>89</b>
850	90	271	92	91	89
850	100	276	100	101	99
850	110	329	137	135	135
850	120	391	184	187	187
850	130	447	265	269	268
940	40	321	176	177	176
940	50	255	135	135	134
940	60	248	132	132	131
940	70	246	121	120	120
940	80	285	105	140	105
<b>940</b>	<b>90</b>	<b>292</b>	<b>90</b>	<b>91</b>	<b>90</b>

(Continuación.)

(Continuación.)						
BLANCO	30	x	x	x	x	
BLANCO	40	328	154	152	153	
<b>BLANCO</b>	<b>50</b>	<b>219</b>	<b>63</b>	<b>63</b>	<b>63</b>	
BLANCO	60	223	74	71	73	
BLANCO	70	324	89	88	89	
BLANCO	80	388	124	127	127	
BLANCO	90	461	183	186	186	

(Elaboración Propia)

Tabla 2.  
*Precisión del Umbral de Iluminación por Espectro (donde las casillas escritas en rojo muestran la mejor combinación con la menor cantidad de errores por espectro.).*

Espectro	Iluminación	# de errores	
		Seg- 30	Seg- 50
460	45	109	108
<b>460</b>	<b>50</b>	<b>79</b>	<b>79</b>
460	55	107	108
630	45	60	61
630	50	62	62
<b>630</b>	53	65	64
<b>630</b>	<b>55</b>	<b>58</b>	<b>58</b>
630	57	59	59
700	43	60	60
<b>700</b>	<b>45</b>	<b>54</b>	<b>54</b>
<b>700</b>	<b>50</b>	<b>54</b>	<b>54</b>
<b>700</b>	<b>55</b>	<b>54</b>	<b>54</b>
700	57	57	57
850	80	88	88
850	83	84	85
850	85	86	84
<b>850</b>	86	86	86
850	87	82	82
<b>850</b>	<b>88</b>	<b>81</b>	<b>81</b>
850	89	89	90
940	85	94	93
940	89	91	91
<b>940</b>	<b>90</b>	<b>90</b>	<b>91</b>
940	91	101	101
940	95	93	94
BLANCO	45	82	82
BLANCO	45	70	70
<b>BLANCO</b>	<b>50</b>	<b>63</b>	<b>63</b>
BLANCO	53	65	65
BLANCO	55	69	69

(Elaboración Propia)

Tabla 3.  
*Iluminación Óptima por Espectro.*

Espectro	Iluminación	30
460	50	79
630	55	58
700	45	54
700	50	54
700	55	54
850	88	81
940	90	90
BLANCO	50	63

(Elaboración Propia)

### ANEXO E

## TABLAS RESUMEN DE LA EVALUACIÓN DE SEGMENTACIÓN SEGÚN SU ESPECTRO

Tabla 1.  
Precisión del Número de Segmentaciones por Espectro-iluminación.

Espectro	Iluminación	Segmentación	# Error
460	50	30	79
		20	79
		15	79
		14	80
		13	82
		10	80
630	55	30	58
		20	58
		10	58
		7	58
		6	58
		5	59
700	45	30	54
		20	54
		10	54
		9	54
		8	54
		7	55
700	50	5	56
		30	54
		20	54
		10	54
		7	54
		6	55
700	55	5	55
		30	54
		28	54
		27	55
		26	55
		25	54
850	88	20	56
		30	81
		27	81
		26	82
		25	82
		20	82
940	90	30	90
		29	90
		28	91
		25	91
		20	91
		30	91
BLANCO	50	20	63
		10	63
		5	63
		4	65
		3	64
		30	63

(Elaboración propia)

Tabla 2

*Segmentación óptima para la Combinación Espectro- Iluminación.*

Espectro	Iluminación	Segmentación	# Error
460	50	15	79
630	55	6	58
700	45	8	54
700	50	7	54
700	55	28	54
850	88	27	81
940	90	29	90
BLANCO	53	5	63

(Elaboración propia)



## RESULTADO DE TIEMPOS POR EQUIPOS EVALUADOS

Tabla 1.

*Tiempos Obtenidos en el Reconocimiento de la Palma por Equipo de Evaluación.*

Cod. Equipo	Número de Núcleos	Frecuencia de Reloj	Memoria	Grupo	Prueba 1 (seg)	Prueba 2 (seg)	Prueba 3 (seg)	Prueba 4 (seg)	Promedio (seg)
1	7	3.4Ghz	8 Gb	Experimental	26.403	26.125	26.292	28.477	26.824
				Control	134.182	133.633	134.299	134.336	134.113
2	7	3.07Ghz	6Gb	Experimental	33.666	33.427	32.547	32.957	33.149
				Control	243.284	243.574	242.254	242.574	242.922
3	7	1.6Ghz	4Gb	Experimental	68.824	69.583	65.644	65.033	67.271
				Control	311.649	306.825	307.639	309.001	308.779
4	5	2.27Ghz	4Gb	Experimental	78.999	81.909	86.221	77.985	81.279
				Control	318.560	318.303	326.869	328.244	322.994
5	4	2.00Ghz	4Gb	Experimental	77.194	72.479	71.254	75.455	74.096
				Control	313.304	313.057	310.541	315.240	313.036
6	3	3.30Ghz	4Gb	Experimental	51.800	52.310	51.915	51.786	51.953
				Control	162.020	162.073	161.983	161.929	162.001
7	3	2.4Ghz	4Gb	Experimental	71.802	72.961	71.978	72.322	72.266
				Control	224.079	224.140	223.501	223.564	223.821
8	2	2.13Ghz	4Gb	Experimental	89.823	90.132	90.941	89.648	90.136
				Control	384.403	380.444	381.689	384.442	382.745
9	2	2.30Ghz	4Gb	Experimental	291.290	286.941	268.017	262.135	277.096
				Control	607.724	597.841	597.154	606.265	602.246

(elaboración Propia)

Tabla.

*Tiempo promedio y eficiencia por equipo evaluado.*

Cod. Equipo	Reloj	Número de Núcleos	Memoria	Grupo Experimental	Grupo de control	Eficiencia (# de veces más rápido)
1	3.4Ghz	7	7N-8Gb-3.40Ghz	26.824	134.113	5.00
2	3.07Ghz	7	7N-6Gb-3.07Ghz	33.149	242.922	7.32
3	1.6Ghz	7	7N-4Gb-1.60Ghz	67.271	308.779	4.59
4	2.27Ghz	5	5N-4Gb-2.27Ghz	81.279	322.994	3.97
5	2.00Ghz	4	4N-4Gb-2.00Ghz	74.096	313.036	4.22
6	3.30Ghz	3	3N-4Gb-3.30Ghz	51.953	162.001	3.12
7	2.4Ghz	3	3N-4Gb-2.40Ghz	72.266	223.821	3.10
8	2.13Ghz	3	3N-4Gb-2.13Ghz	90.136	382.745	4.25
9	2.30Ghz	2	2N-4Gb-2.30Ghz	277.096	602.246	2.17

(elaboración Propia)

