

**Ing. PABLO EDUARDO CAICEDO RODRÍGUEZ**



**APREHENSIÓN ESTABLE CON UNA MANO ROBÓTICA USANDO  
INFORMACIÓN VISUAL**

**Universidad del Cauca  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Maestría en Electrónica y Telecomunicaciones**

**Popayán  
2011**



**PABLO EDUARDO CAICEDO RODRÍGUEZ**

**APREHENSIÓN ESTABLE CON UNA MANO ROBÓTICA USANDO  
INFORMACIÓN VISUAL**

**Tesis presentada a la Facultad de Ingeniería  
Electrónica y Telecomunicaciones de la  
Universidad del Cauca para la obtención del  
Título de**

**Magíster en Ingeniería, Área  
Electrónica y Telecomunicaciones**

**Director:  
Jaime Oscar Díaz Ch.  
Magíster**

**Popayán  
2011**



Hoja de Aprobación

---

---

---

Director

\_\_\_\_\_  
Ing. Jaime Oscar Díaz Chávez. Mgr.

Jurado

\_\_\_\_\_  
Ing. Carlos Alberto Gaviria López. PhD

Jurado

\_\_\_\_\_  
Ing. Sandra Esperanza Nope Rodríguez. PhD

Fecha de sustentación: Popayán, 1 de Marzo de 2010



# Agradecimientos

A Dios por darme la oportunidad de un logro más en la vida.

A mis padres por el valor y la fuerza que me inculcaron

A mi esposa por su continuo apoyo y amor.

A Jaime Díaz por su tiempo, dedicación y esmero como director del proyecto

A Marta sin tu ayuda y guía, el final de este camino no habría sido visto nunca.

A María Teresa por introducirme el concepto de curtosis.

A mis amigos que me dieron el valor de la amistad y algunas ideas para terminar este proyecto

A la Universidad Antonio Nariño y a la Corporación Universitaria Autónoma del Cauca por el tiempo y el apoyo para el desarrollo de este proyecto.

A la Universidad del Cauca y el Instituto de Postgrados en Electrónica y Telecomunicaciones por su colaboración administrativa y económica.



# Resumen

Para una persona que no presente discapacidad motriz, sujetar un objeto es una tarea tan usual que seguramente nunca habrá discernido acerca de la complejidad de este proceso, al punto de poder tomar un vaso hasta con los ojos cerrados. Si la misma tarea es realizada por una máquina, aspectos como los puntos de agarre, la cantidad de fuerza a aplicar, la ubicación de los elementos, etc., se convierten en el centro de interés pues se debe evitar que los cuerpos se deslicen.

El desarrollo de este trabajo giró en torno a la identificación de los puntos de aprehensión de diversos objetos, usando un agarre de dos y tres dedos. Para tal fin, se recurrió al uso de diversas técnicas de procesamiento de imágenes digitales, en las cuales se visualizan los elementos a sujetar. Para cumplir con este propósito, las tareas realizadas fueron:

1. Capturar la escena usando una cámara web; la imagen debe tener alto contraste.
2. Segmentar los objetos presentes en la imagen.
3. Seleccionar el elemento cuyo centro de masa esté más cercano al centro de la escena, y establecer sus características físicas (curvatura y vectores normales a la curvatura)
4. Identificar los posibles puntos de agarre, los cuales deben cumplir los criterios de fricción (para dos dedos) y del área de intersección de los conos de fricción (para agarre con tres dedos).

Este proyecto está enmarcado en el proyecto “Prótesis robótica de mano” que fue propuesto por el grupo de investigación en Automática Industrial de la Universidad del Cauca.

**Palabras Clave:** Agarre estable, análisis de imágenes, segmentación, curtosis.



# Abstract

For a human without any handicap, grasp an object is an usual task, so usual that he or she doesn't realize the complexity of this process. When the same task (grasp an object) is done by a computer, certain aspects like grasp points, applied forces, objects location, between other, becomes important, because these are the main topics to avoid an unstable grasp (object slides off from the grasper).

This Project development it's about to identify two and three fingered grasping points of various objects, this is made with different digital image processing techniques that calculate them. To accomplish this objective, the principal task to be made were:

1. Acquire 3D scene using a webcam. This image must have high contrast
2. Segmentation of the object in the scene
3. Select the element whom mass center is closest to the center of the scene and capture this objects's physical characteristics (Curvature and normal vectors)
4. Identify the possible grasping point, which must meet the friction criteria (for two-fingered grasp) and intersection area of friction cones (for three-fingered grasp)

This project is part of the Automatic Industrial investigation group's project "Prótesis robótica de mano" at Cauca's University

**Key Words:** Stable grasp, artificial vision, segmentation, kurtosis.



# Contenido

<b>AGRADECIMIENTOS</b>	<b>7</b>
<b>RESUMEN</b>	<b>9</b>
<b>ABSTRACT</b>	<b>11</b>
<b>CONTENIDO</b>	<b>13</b>
<b>LISTA DE FIGURAS</b>	<b>15</b>
<b>LISTA DE TABLAS</b>	<b>17</b>
<b>CAPÍTULO 1. INTRODUCCIÓN</b>	<b>19</b>
1.1 DESCRIPCIÓN DEL PROBLEMA	20
1.2 OBJETIVOS	22
1.2.1 <i>Objetivo general</i>	22
1.2.2 <i>Objetivos específicos</i>	22
<b>CAPÍTULO 2. MARCO CONCEPTUAL</b>	<b>25</b>
2.1 APREHENSIÓN	25
2.1.1 <i>Definición</i>	25
2.1.2 <i>Clasificación de los agarres</i>	26
2.2 EJECUCIÓN DE AGARRES	26
2.2.1 <i>Ejecución de Agarres con Visión Artificial</i>	26
2.2.2 <i>Ejecución de Agarres con 2 Dedos</i>	27
2.2.3 <i>Ejecución de Agarres con 3 Dedos</i>	29
<b>CAPÍTULO 3. CARACTERIZACIÓN DE LOS ELEMENTOS INVOLUCRADOS EN EL PROBLEMA</b>	<b>31</b>
3.1 SISTEMA DE ADQUISICIÓN	31
3.2 CONDICIONES AMBIENTALES	34
<b>CAPÍTULO 4. ALGORITMO PARA LA SELECCIÓN DE OBJETOS EN UNA ESCENA</b>	<b>35</b>
4.1 PREPROCESAMIENTO	35
4.1.1 <i>Conversión a escala de grises</i>	36
4.1.2 <i>Eliminación de ruido</i>	36
4.2 SEGMENTACIÓN	38
4.3 CÁLCULO DEL CENTRO DE MASA Y LA ORIENTACIÓN	43
4.4 DETERMINACIÓN DEL OBJETO DE INTERÉS EN LA ESCENA	44
<b>CAPÍTULO 5. ALGORITMO PARA IDENTIFICACIÓN DE PUNTOS DE AGARRE A PARTIR DE UNA ESCENA.</b>	<b>47</b>
5.1 DETECCIÓN DE BORDES	47

5.2	IDENTIFICACIÓN DE PUNTOS DE APREHENSIÓN	52
5.2.1	<i>Cálculo de la curvatura</i>	52
5.2.2	<i>Identificación de los puntos de aprehensión estable para un agarre con dos dedos.</i>	53
5.2.3	<i>Identificación de los puntos de aprehensión estable para un agarre con tres dedos</i>	55
<b>CAPÍTULO 6.</b>	<b>ANÁLISIS DE RESULTADOS</b>	<b>57</b>
6.1	DESCRIPCIÓN DE LA SOLUCIÓN AL PROBLEMA	57
6.2	DESCRIPCIÓN GENERAL DEL ALGORITMO	58
6.2.1	<i>Algoritmo para seleccionar el objetivo</i>	59
6.2.2	<i>Cálculo de puntos de aprehensión estable</i>	64
6.3	IMPLEMENTACIÓN DEL ALGORITMO.	68
6.4	PROTOCOLO DE PRUEBAS DEL ALGORITMO	69
6.4.1	<i>Estabilidad</i>	69
6.4.2	<i>Desplazamiento</i>	71
6.4.3	<i>Rotación</i>	72
6.4.4	<i>Iluminación</i>	72
6.4.5	<i>Color</i>	75
6.4.6	<i>Tiempo de Procesamiento</i>	76
<b>CAPÍTULO 7.</b>	<b>CONCLUSIONES Y RECOMENDACIONES</b>	<b>81</b>
7.1	CONCLUSIONES	81
7.2	APORTES DE ESTE TRABAJO	83
7.3	TRABAJOS FUTUROS	83
7.3.1	<i>Imágenes</i>	83
7.3.2	<i>Objetos</i>	84
7.3.3	<i>Descripción espacial real de Puntos en una Imagen</i>	84
<b>CAPÍTULO 8.</b>	<b>BIBLIOGRAFÍA</b>	<b>87</b>

# Lista de Figuras

<b>FIGURA 1:</b> POSICIÓN DE LA CÁMARA PARA EVITAR SOLAPAMIENTOS .....	21
<b>FIGURA 2:</b> AGARRE POR UN EXTREMO O "DE PUNTA" .....	21
<b>FIGURA 3:</b> MOVIMIENTO ABDUCCIÓN DE DEDOS. ....	22
<b>FIGURA 4:</b> EL PROBLEMA DE LA APREHENSIÓN. FUENTE [3] .....	25
<b>FIGURA 5:</b> ALGORITMO DE EXTRACCIÓN DE OBJETOS DE UNA IMAGEN. TOMADA DE [2] .....	26
<b>FIGURA 6:</b> INTERPRETACIÓN GEOMÉTRICA DE LA CONDICIÓN NO-DESIZANTE. ADAPTADO DE [7] .....	28
<b>FIGURA 7:</b> DEFINICIÓN DE CERRADURA POR FUERZA. ADAPTADO DE [7] .....	28
<b>FIGURA 8:</b> ASPECTOS BÁSICOS DEL AGARRE CON DOS DEDOS. (A) GEOMETRÍA DEL OBJETO. (B) GEOMETRÍA DE LA MANO. (C) DISTANCIA ENTRE LOS PUNTOS DE AGARRE. ADAPTADO DE [5] .....	29
<b>FIGURA 9:</b> DIAGRAMA PARA APREHENSIÓN CON TRES DEDOS .....	30
<b>FIGURA 10:</b> SENSOR TIPO CCD. FUENTE [22] .....	31
<b>FIGURA 11:</b> SENSOR TIPO CMOS. FUENTE [22] .....	32
<b>FIGURA 12:</b> ILUMINACIÓN SEMICONTROLADA [25] .....	34
<b>FIGURA 13:</b> DIAGRAMA DE FLUJO DEL ALGORITMO PARA SELECCIÓN DE OBJETOS EN UNA ESCENA .....	35
<b>FIGURA 14:</b> DEMOSTRACIÓN DEL EFECTO DEL RUIDO. ....	37
<b>FIGURA 15:</b> MÁSCARA PARA FILTRO PROMEDIO .....	37
<b>FIGURA 16:</b> EJEMPLOS DE IMÁGENES RUIDO. (A) FACILMENTE SEGMENTABLE. (B) DIFÍCILMENTE SEGMENTABLE .....	38
<b>FIGURA 17:</b> ALGORITMO BÁSICO PARA UMBRALIZACIÓN .....	39
<b>FIGURA 18:</b> CLASIFICACIÓN DE DISTRIBUCIONES SEGÚN SU CURTOSIS .....	42
<b>FIGURA 19:</b> PROCESO DE CÁLCULO DEL UMBRAL .....	42
<b>FIGURA 20:</b> DEFINICIÓN DE ÁNGULO DE ORIENTACIÓN .....	44
<b>FIGURA 21:</b> DEFINICIÓN DE VECINDAD-4 .....	45
<b>FIGURA 22:</b> DIAGRAMA DE FLUJO DEL ALGORITMO PARA IDENTIFICACIÓN DE PUNTOS DE AGARRE .....	47
<b>FIGURA 23:</b> IMAGEN BASE PARA CÁLCULO DE BORDES .....	48
<b>FIGURA 24:</b> ALGORITMO DE CANNY .....	50
<b>FIGURA 25:</b> ELEMENTO ESTRUCTURANTE .....	51
<b>FIGURA 26:</b> RESULTADO DE LAS PRUEBAS PARA LA FIGURA 23. APLICANDO: (A) ALGORITMO DE SOBEL, (B) ALGORITMO DE PREWITT, (C) ALGORITMO DE ROBERTS, (D) ALGORITMO DE ROBINSON, (E) ALGORITMO DE CANNY Y (F) ALGORITMO POR MORFOLOGÍA .....	51
<b>FIGURA 27:</b> DEFINICIÓN DE CURVATURA .....	53
<b>FIGURA 28:</b> CÁLCULO DE B Y B' .....	53
<b>FIGURA 29:</b> ALGORITMO PARA CÁLCULO DE PUNTOS DE APREHENSIÓN EN AGARRES CON TRES DEDOS .....	55
<b>FIGURA 30:</b> HIPÓTESIS DE SOLUCIÓN PARA EL PROBLEMA DEL PROYECTO .....	57
<b>FIGURA 31:</b> ESQUEMA GENERAL DE LA SOLUCIÓN .....	58
<b>FIGURA 32:</b> ALGORITMO GENERAL DEL PROYECTO .....	58
<b>FIGURA 33:</b> UTILIZACIÓN FILTRO DE MEDIANA .....	59
<b>FIGURA 34:</b> APLICACIÓN DEL ALGORITMO DE NAGAO-MATSUYAMA .....	60
<b>FIGURA 35:</b> COMPROBACIÓN ALGORITMO SEGMENTACIÓN .....	61
<b>FIGURA 36:</b> RESULTADOS DE SEGMENTACIÓN CON OBJETOS ESTAMPADOS .....	63
<b>FIGURA 37:</b> RESULTADOS DEL ALGORITMO DE SELECCIÓN DE OBJETIVO .....	64

<b>FIGURA 38.</b> RESULTADOS ALGORITMO DETECCIÓN DE BORDES .....	64
<b>FIGURA 39.</b> RESULTADO DEL CÁLCULO DE LOS PUNTOS DE APREHENSIÓN .....	66
<b>FIGURA 40:</b> RESULTADOS ALGORITMO DE DETECCIÓN DE PUNTOS DE APREHENSIÓN ESTABLE PARA UN AGARRE CON 3 DEDOS. (A) IMAGEN ORIGINAL. (B) DIVISIÓN DE CURVATURA EN 3 ZONAS. (C) DIAGRAMADO DE CONOS DE FRICCIÓN Y DETECCIÓN DE ÁREA COMÚN. (D) RESULTADO DEL ALGORITMO .....	67
<b>FIGURA 41:</b> IMÁGENES PARA SUSTENTAR LA UBICACIÓN DE LA CÁMARA .....	68
<b>FIGURA 42.</b> ESTRUCTURA DEL ALGORITMO IMPLEMENTADO.....	68
<b>FIGURA 43:</b> RESULTADO DE PUNTOS DE APREHENSIÓN EN LA REFERENCIA [2]. (A) IMAGEN ORIGINAL. (B) IMAGEN RESULTANTE (FUENTE: [2]). (C) RESULTADO DEL PROCESAMIENTO DE (A) CON ESTE ALGORITMO. ....	70
<b>FIGURA 44:</b> RESULTADO DE PUNTOS DE APREHENSIÓN EN LA REFERENCIA [2]. (A) IMAGEN ORIGINAL. (B) IMAGEN RESULTANTE (FUENTE [2]). (C) RESULTADO DEL PROCESAMIENTO DE (A) CON ESTE ALGORITMO. ....	70
<b>FIGURA 45:</b> RESULTADOS DEL PROCESAMIENTO PARA UNA TUERCA. (A) REFERENCIA [2]. (B) ALGORITMO DISEÑADO .....	71
<b>FIGURA 46:</b> IMÁGENES PARA DEMOSTRAR LA INMUNIDAD AL DESPLAZAMIENTO DEL ALGORITMO. ....	71
<b>FIGURA 47:</b> IMÁGENES PARA COMPROBACIÓN DE LA INMUNIDAD A LA ROTACIÓN. ....	72
<b>FIGURA 48:</b> MONTAJE CON DIFERENTES FUENTES DE ILUMINACIÓN UTILIZADAS PARA LAS PRUEBAS.....	73
<b>FIGURA 49:</b> IMÁGENES CON DIFERENTES ILUMINACIONES.....	74
<b>FIGURA 50:</b> RESULTADOS DEL ALGORITMO PARA LA FIGURA 49 .....	75
<b>FIGURA 51:</b> IMÁGENES PARA COMPROBACIÓN DE COLOR .....	75
<b>FIGURA 52:</b> RESULTADO PARA IMÁGENES DE LA FIGURA 51: IMÁGENES PARA COMPROBACIÓN DE COLOR .....	75
<b>FIGURA 53:</b> IMÁGENES PARA COMPROBAR EL LÍMITE DE LA INMUNIDAD AL COLOR .....	76
<b>FIGURA 54.</b> IMÁGENES PARA PROBAR EL TIEMPO DE PROCESAMIENTO .....	77
<b>FIGURA 55.</b> RESULTADOS DE LAS IMÁGENES .....	78
<b>FIGURA 56:</b> NECESIDAD DE SISTEMA DE ELIMINACIÓN DE RUIDO. (A) SIN SISTEMA DE ELIMINACIÓN. (B) CON SISTEMA DE ELIMINACIÓN. ....	82
<b>FIGURA 57:</b> ILUMINACIÓN RADIAL .....	84

# Lista de Tablas

<b>TABLA 1.</b> COMPARACIÓN ENTRE SENSORES DE ADQUISICIÓN DE IMÁGENES. FUENTE [22].....	32
<b>TABLA 2.</b> FICHA TÉCNICA DE LA CÁMARA WEB .....	33
<b>TABLA 3.</b> CLASIFICACIÓN DISTRIBUCIONES SEGÚN SU CURTOSIS .....	42
<b>TABLA 4.</b> DIVISIÓN HISTOGRAMA PROCESO CÁLCULO UMBRAL ÓPTIMO .....	43
<b>TABLA 5.</b> COMPARACIÓN DE ALGORITMOS PARA LA DETECCIÓN DE BORDES .....	52
<b>TABLA 6.</b> COMPARACIÓN DE ALGORITMOS DE CURTOSIS VS OTSU PARA IMÁGENES DE ALTA LUMINANCIA .....	62
<b>TABLA 7.</b> COMPARACIÓN DE ALGORITMOS DE CURTOSIS VS OTSU FRENTE A IDENTIFICACIÓN DE OBJETOS EN FONDOS COLORIDOS.....	62
<b>TABLA 8.</b> RESUMEN DE RESULTADOS DEL CÁLCULO DE PUNTOS DE APREHENSIÓN PARA UNA IMAGEN.....	65
<b>TABLA 9.</b> CARACTERÍSTICAS DE LAS IMÁGENES DE PRUEBA .....	76
<b>TABLA 10.</b> RESUMEN DE TIEMPOS DE PROCESAMIENTO .....	78



# Capítulo 1. Introducción

Cada día, las aplicaciones del campo de la robótica están más cerca de la gente del común: desde las historietas infantiles hasta las prótesis para discapacitados; sin embargo aún falta camino por recorrer, un aspecto a mejorar es la interacción del robot con su entorno, especialmente en ambientes dinámicos donde los obstáculos que deben ser superados emergen y desaparecen en el tiempo. En el caso de las prótesis médicas en general, usadas para restaurar un cierto grado de la funcionalidad de un órgano amputado, se hace necesario que ésta emule [1],[2] los comportamientos del miembro que va a suplir, para lo cual se requiere: sentido continuo de las condiciones del medio en el que funciona, adecuarse a los cambios en el entorno y responder a órdenes heterogéneas.

En el caso particular de una prótesis de mano, y específicamente la que sirve de base en este proyecto, se pretende que pueda sujetar objetos de manera estable (función principal de una mano humana); de esta forma se espera que si la prótesis puede realizar el cálculo del agarre, su interacción con el paciente mejorará y el movimiento de la mano robótica será más flexible. Para efectuar el cálculo antes mencionado, se requiere establecer tres aspectos:

1. Identificar los puntos donde serán ubicados los dedos.
2. Determinar la orientación de cada uno de los dedos sobre el punto.
3. Establecer la cantidad de fuerza que se debe aplicar para agarrar el objeto.

Con los resultados de este trabajo se da solución al primer punto, siendo los otros dos, materia de trabajos posteriores.

Para modelar el funcionamiento de una prótesis médica que reemplace la mano humana, el desarrollo de algoritmos se encaminan hacia la identificación de los puntos de aprehensión estables; en tal sentido, se parte de la siguiente hipótesis: se puede desarrollar un algoritmo que permita a la prótesis robótica determinar dichos puntos, usando tecnología accesible a las condiciones de países en vía de desarrollo donde los recursos económicos son escasos. De acuerdo con lo anterior, esta investigación constituye un aporte importante en el desarrollo de soluciones que permitan a un usuario de prótesis de mano, en el futuro tomar objetos y trasladarlos o manipularlos. En una aprehensión además de cerrar los dedos de la

prótesis y aplicar la fuerza adecuada (que no rompa el objeto) se debe considerar en qué punto(s) se va a aplicar este esfuerzo.

El desarrollo de algoritmos para agarre es una parte fundamental de la investigación sistémica de la prótesis para reemplazo de una mano humana, que está adelantando el grupo de investigación en Automática Industrial de la Universidad del Cauca; por tal razón, el problema está siendo abordado desde diferentes disciplinas asociadas a la robótica como el modelamiento, el control, las matemáticas, la implementación, el sensado de los agentes externos que interactúan con el robot; de esta manera al considerar los requerimientos de otros investigadores, se garantiza que las prótesis puedan suplir las funciones que por alguna razón el cuerpo ha dejado de cumplir.

## 1.1 Descripción del problema

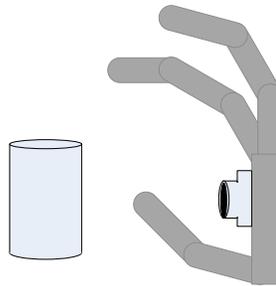
La tarea de agarrar un objeto involucra identificar los puntos de agarre del objeto de forma tal que no se deslice y estimar cuanta fuerza aplicar; esto implica adoptar decisiones a partir de múltiples factores: tamaño, composición, estática y dinámica del objeto; tamaño, composición, estática y dinámica del dedo; número de dedos y de objetos en la escena. Teniendo en cuenta la magnitud y complejidad de factores que deben ser estudiados, es necesario delimitar el alcance de este proyecto.

En primer lugar, es preciso tener en cuenta las características propias de la mano robótica construida por el grupo de investigación en Automática Industrial de la Universidad del Cauca. La prótesis de mano tiene tres dedos, por lo que este trabajo se ocupa en determinar los puntos de aprehensión estable para agarres con dos y tres dedos únicamente.

Otras consideraciones que se tienen presentes para el desarrollo de este proyecto son las siguientes.

En cuanto a las imágenes:

- Se usa una sola cámara. Aunque la presencia de dos cámaras permitiría una reconstrucción 3D, la estructura mecánica sería compleja. Por otro lado, esta solución consigue maximizar el espacio físico en la prótesis y permite la ubicación de otros componentes (ver **Figura 1**).



**Figura 1:** Posición de la cámara para evitar solapamientos

- Para la iluminación se asume un ambiente común. En la literatura del procesamiento de imágenes este tipo de iluminación se conoce como tipo semicontrolada. Aunque eventualmente se pudieran presentar iluminaciones complejas, como a contra luz o difusas, poco comunes, se asume que los algoritmos desarrollados tengan un comportamiento aceptable. En este sentido se harán las pruebas necesarias para comprobar su funcionamiento.
- Se asume que los objetos presentes en la escena son claramente diferenciables del fondo.

En cuanto a los objetos:

- La información del borde del objeto es suficiente para calcular los puntos de agarre estable. En términos generales esto quiere decir que los objetos son poco profundos, desde el punto de vista de la cámara. Esta situación es habitual, si se tiene en cuenta que normalmente no se toman los objetos por un extremo o "de punta".

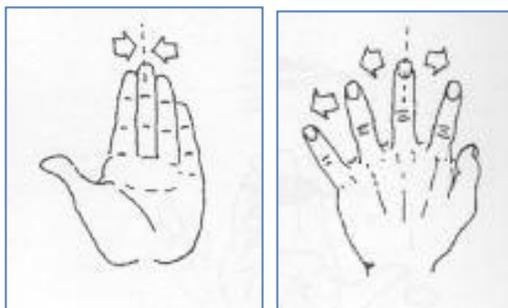


**Figura 2:** Agarre por un extremo o "de punta"

- Poseen una masa uniformemente distribuida, dado que esta es una condición necesaria para que el centro geométrico (calculable a través de procesamiento de imágenes) coincida con el centro de masa. Esto implica que el número de objetos que se procesarán correctamente está limitado a una clase muy específica, aunque una descripción completa en tres dimensiones podrá aliviar en parte esta situación<sup>1</sup>.

<sup>1</sup> Esta descripción será objetivo de próximas investigaciones.

Aunque este proyecto resuelve el problema de un agarre de tres dedos, debe tenerse en cuenta que la mano robótica que le sirve de base, desarrollada por el Grupo de Investigación en Automática Industrial, carece de la habilidad para realizar la abducción de los dedos (ver Figura 3). Sin embargo, este desarrollo seguramente será útil en el futuro.



**Figura 3:** Movimiento Abducción de Dedos.

Fuente: <http://www.estrucplan.com.ar/producciones/entrega.asp?identrega=65>

## 1.2 Objetivos

### 1.2.1 Objetivo general

Determinar los puntos de agarre estable de un objeto a partir de la información obtenida de una cámara situada en una prótesis de mano existente.

### 1.2.2 Objetivos específicos

- Determinar las características del objeto a tomar a partir de la imagen obtenida por la cámara.
- Desarrollar un algoritmo que permita seleccionar el objeto a tomar deseado a partir de la imagen de la escena.
- Desarrollar un algoritmo para calcular los puntos de agarre partir de una imagen.

El presente trabajo se encuentra desarrollado de la siguiente manera: en el capítulo 2 se encuentra una breve descripción de los conceptos teóricos y metodológicos

del trabajo. En el capítulo 3 se hace una descripción de las características de los elementos involucrados dentro del desarrollo del proyecto. En el capítulo 4 se describe como se realiza la selección de los objetos dentro de la escena. En el capítulo 5 se describe el algoritmo que resuelve el problema de la identificación de los puntos de aprehensión estable para dos y tres dedos. El capítulo 6 hace una descripción y análisis de los resultados obtenidos en los diferentes algoritmos. Por último se harán las conclusiones y las recomendaciones en el capítulo 7.



## Capítulo 2. Marco conceptual

En el presente apartado se hará una descripción de los principales conceptos teóricos involucrados en la resolución del problema de este proyecto. Como se describió en el apartado anterior el problema básico es la detección de puntos para una aprehensión estable basándose en la información proveniente de una cámara web, es por ello que este capítulo hará una descripción de conceptos como: aprehensión, aprehensión estable ejecución de agarres con dos y tres dedos mediante el uso de visión artificial.

### 2.1 Aprehensión

#### 2.1.1 Definición

De acuerdo a [3], un agarre o aprehensión es *la aplicación de fuerzas efectivas y funcionales, a través de la mano, a un objeto para realizar una tarea, dadas un número determinado de restricciones.*

En la Figura 4 se puede observar el problema general de la aprehensión (artificial o natural) desde una perspectiva sistémica; como entradas al sistema se tienen el objeto y la tarea a llevar a cabo y la salida es un comportamiento prensil [3]; éste se mide por medio de las posturas y las fuerzas a través del tiempo, difiere según las características del objeto y la tarea.

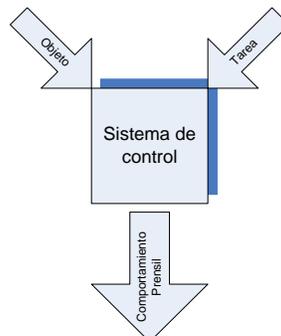


Figura 4: El problema de la aprehensión. Fuente [3]

### 2.1.2 Clasificación de los agarres

Existen diversas formas de agrupación, atendiendo múltiples características:

- Según la forma en que se realiza: agarre por clausura de forma y agarre por clausura de fuerza.
- Considerando si se conoce o no el objeto a inmovilizar.
- De acuerdo con el número de dedos del sistema de aprehensión [2].

Para mayor claridad, según [4], [5] y [6] la cerradura por fuerza se realiza cuando los puntos de contacto ejercen fuerzas y torques para contrarrestar cualquier perturbación externa al objeto. Cerradura por forma se presenta cuando el agarre inmoviliza por completo el objeto por medio de un conjunto de puntos de contacto.

Para el desarrollo del proyecto se consideró el cálculo de puntos de aprehensión con dos y tres dedos de un objeto desconocido por medio del uso de procesamiento de imágenes.

## 2.2 Ejecución de Agarres

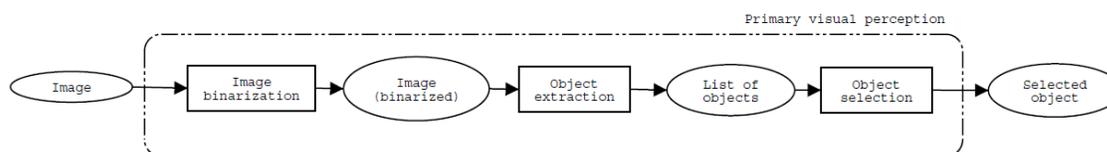
Para realizar el cálculo de puntos de aprehensión se tuvieron en cuenta tres aspectos, que son detallados a continuación:

- Uso del procesamiento de imágenes para obtener los puntos de aprehensión.
- Agarre estable con dos dedos.
- Agarre estable con tres dedos.

Cabe anotar que los cálculos son realizados en dos dimensiones, es decir teniendo en cuenta la condición de la profundidad explicada en el apartado 1.1.

### 2.2.1 Ejecución de Agarres con Visión Artificial

Según las referencias [2],[7]-[18] para ejecutar un agarre con visión artificial se deben seguir una serie de pasos. El primer paso es extraer el objeto al cuál se le hará la operación, y para ello se siguen los pasos descritos en la Figura 5:



**Figura 5:** Algoritmo de extracción de objetos de una imagen. Tomada de [2]

En la Figura 5, los rectángulos representan operaciones de procesamiento de imágenes y las elipses simbolizan los resultados de dichas operaciones; la elipse rotulada *image* es la imagen de entrada del algoritmo.

Una vez se ha hecho selección del objeto, este debe ser caracterizado. Caracterizar implica extracción de información acerca del objeto. La información básica requerida es la curvatura del objeto. El resto de información depende del número de dedos que se utilizan para realizar el agarre. Ahora se explicará cómo se realiza una aprehensión con dos o tres dedos, tema específico de este trabajo.

### 2.2.2 Ejecución de Agarres con 2 Dedos

Es importante hacer claridad entre los conceptos de equilibrio y estabilidad de un agarre. Según Nguyen [4], un agarre está en equilibrio cuando la sumatoria de fuerzas y la sumatoria de torques sobre un objeto son iguales a cero. La estabilidad de un agarre implica que el objeto aprehendido siempre vuelve a estar en equilibrio aunque haya sido perturbado por una fuerza externa [4].

Según Morales [7], la estabilidad de un agarre se divide en:

- Estabilidad de contacto: todos los aspectos que involucran la unión del dedo con el objeto (contacto).
- Estabilidad de objeto: conjunto de fuerzas y torques necesarios para contrarrestar las fuerzas externas y sus torques ejercidos, que afectan al objeto.

Para lograr estabilidad de contacto es indispensable tener en cuenta su tipo. En los contactos sin fricción, no se pueden generar las fuerzas necesarias para evitar el deslizamiento del dedo cuando éste aplica la fuerza sobre el objeto, debido a que este tipo de contacto solo permite la aplicación de fuerzas normales al objeto [4]. En cuanto a los contactos con fricción, permiten la aplicación de fuerzas normales y tangenciales que evitan el deslizamiento del contacto[4].

De acuerdo a lo anterior, para un contacto con fricción, la condición no-deslizante es un aspecto muy importante para lograr la estabilidad de contacto [7], ésta proviene de la ley de Coulomb [19]: “La fuerza de fricción tangencial durante un deslizamiento es directamente opuesta al movimiento, y la magnitud es proporcional a la fuerza normal”. La magnitud proporcional se conoce como coeficiente de fricción estática y depende de los materiales de los contactos.

Entonces un objeto no se desliza si:

$$\overline{f^t} \leq \mu \overline{f^N} \tag{Ec. 1}$$

Haciendo un despeje sencillo se tiene que:

$$\arctan\left(\frac{\overline{f^t}}{\overline{f^N}}\right) \leq \arctan(\mu) \tag{Ec. 2}$$

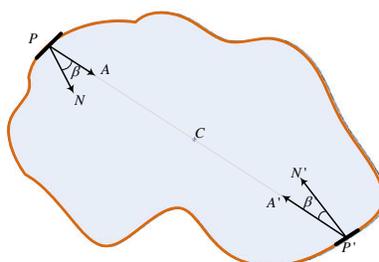
La Ec. 2 define la condición para lograr la estabilidad de contacto.

Desde el punto de vista geométrico, esta ecuación describe un cono, que se conoce como cono de fricción (ver Figura 6).



**Figura 6.** Interpretación geométrica de la condición no-deslizante. Adaptado de [7]

Como lo plantea [7] una aprehensión a través de dos contactos con fricción se logra cuando la línea de agarre (línea que une los dos puntos de aprehensión) está dentro de los dos conos de fricción.



**Figura 7.** Definición de cerradura por fuerza. Adaptado de [7]

En la Figura 7, se observan los elementos mencionados en la definición dada anteriormente de aprehensión con dos contactos con fricción. Los contactos  $P$  y  $P'$  tienen vectores normales  $N$  y  $N'$ . Los vectores  $A$  y  $A'$  son los vectores hacia cada uno de los contactos en el sentido de la línea de agarre, desde ahora llamados antiparalelos. Los ángulos  $\beta$  y  $\beta'$  son los ángulos entre el vector normal y el vector antiparalelo respectivo.

Entonces, la cerradura por fuerza se crea cuando se cumple la siguiente condición:

$$\begin{aligned} \beta &\leq \arctan(\mu) \\ \beta' &\leq \arctan(\mu) \end{aligned} \quad \text{Ec. 3}$$

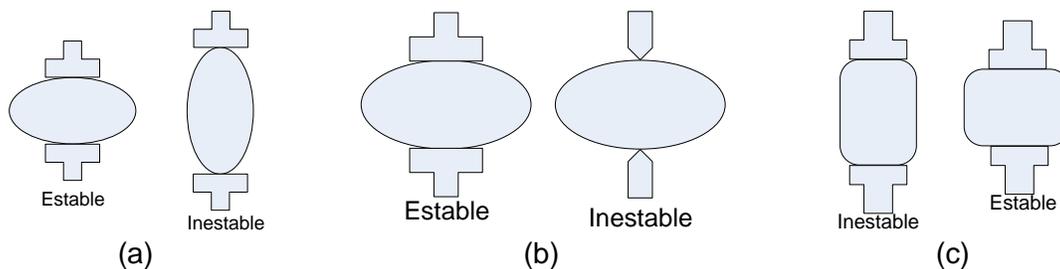
En [2] se plantea que minimizando la distancia al centro de masa desde los contactos aumenta la calidad del agarre, es decir, éste soporta perturbaciones<sup>2</sup> de mayor magnitud. Con los anteriores postulados se logra una cerradura por fuerza.

Pero según Montana en la referencia [5], existen otros criterios que permiten el aumento de la calidad del agarre, los cuales son:

<sup>2</sup> Una perturbación es una fuerza que trata de sacar del equilibrio a un objeto.

1. La geometría local del objeto
2. La geometría local de los dedos
3. La distancia entre los puntos de agarre

En la Figura 8 se representan 6 agarres que cumplen la condición ilustrada en la Figura 7. Los agarres de la Figura 8a corresponde a agarres en el cual la geometría local<sup>3</sup> de los objetos es diferente; en la Figura 8b la geometría de la mano es diferente y la Figura 8c contrasta como la distancia entre los puntos de aprehensión afecta la calidad del agarre [5]. Según la Figura 8a un agarre es tiene mejor calidad, si la geometría local del objeto es más amplia. En la Figura 8b se puede comprobar que en un agarre si el área de contacto entre el dedo y el objeto es proporcional a la calidad del agarre. Por último un agarre es de mayor calidad si la distancia entre puntos de aprehensión opuestos es menor [5].



**Figura 8.** Aspectos básicos del agarre con dos dedos. (a) Geometría del objeto. (b) Geometría de la mano. (c) Distancia entre los puntos de agarre. Adaptado de [5]

Durante este trabajo, no existía la posibilidad de hacer cambios ni a los objetos, ni a los dedos de la prótesis, por lo cual no existe la posibilidad de hacer un refinamiento de los puntos de aprehensión debido a la geometría del objeto ni a por geometría de la mano entonces el refinamiento se realizó por medio de una minimización de la distancia entre los puntos de aprehensión calculados.

### 2.2.3 Ejecución de Agarres con 3 Dedos

Según Ponce y Faverjon [18], comprobado por Recatalá [2] y Morales [16], una cerradura por fuerza se logra si se cumplen dos condiciones:

- $N_1$ ,  $N_2$  y  $N_3$  deben cubrir positivamente a  $\mathbb{R}^2$ . Donde  $N_1$ ,  $N_2$  y  $N_3$  son los vectores normales a los posibles puntos de aprehensión estable. Esto sucede cuando cualquier vector de  $\mathbb{R}^2$  puede ser descrito por una combinación positiva del conjunto de vectores iniciales. Otra forma de ver esta condición es que los vectores normales no se encuentren ubicados en el mismo cuadrante del plano  $\mathbb{R}^2$ .
- La intersección de los tres conos de fricción debe ser diferente de vacío. El ángulo del cono de fricción, está definido por la Ec. 3

<sup>3</sup> La geometría local de un objeto es la medida de la curvatura del objeto en el punto de contacto.

Los elementos descritos en proceso anterior se observan de forma gráfica en la Figura 9. Cabe destacar que  $\frac{\rho_1}{2} = \frac{\rho_2}{2} = \frac{\rho_3}{2} = \beta = \beta'$ , donde  $\beta, \beta'$  se definieron en la Ec. 3

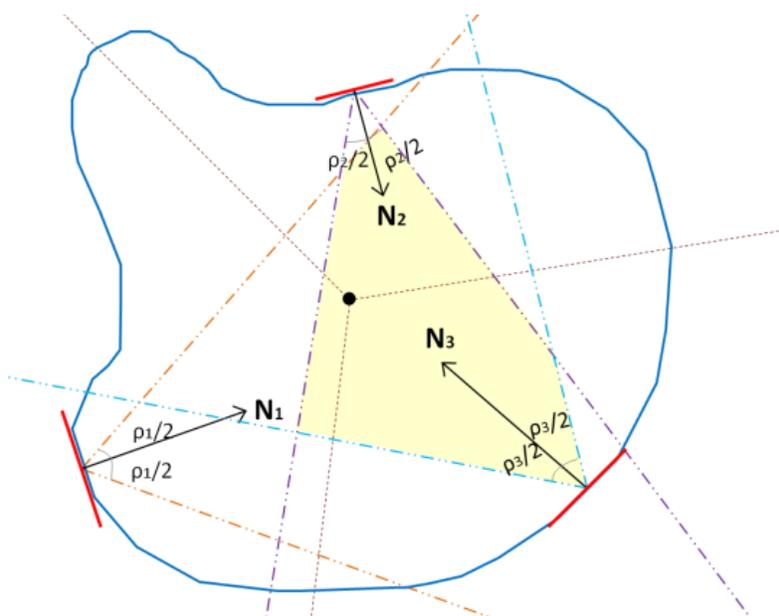


Figura 9. Diagrama para aprehensión con tres dedos

# Capítulo 3. Caracterización de los elementos involucrados en el problema

## 3.1 Sistema de Adquisición

En el proceso de adquisición se obtiene una representación gráfica de la realidad [20]. En la actualidad dicho proceso es realizado por dos tecnologías, la CCD y la CMOS principalmente [22]; en ambas, los sensores acumulan carga eléctrica, según la intensidad lumínica. Esta acumulación sirve como una función de muestreo[22].

En la Figura 10 se observan los bloques funcionales de un sensor tipo CCD. La principal característica de este sensor es que los “paquetes” de carga son enviados secuencialmente a una estructura de salida común donde se hace la conversión de carga a voltaje, después de la cual se almacena y se envía fuera del integrado [22].

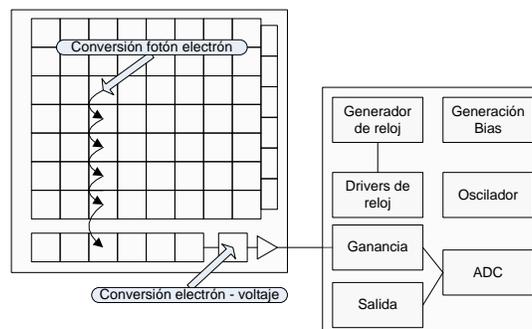


Figura 10. Sensor tipo CCD. Fuente [22]

En la Figura 11 se muestra la configuración básica de un sensor tipo CMOS. Es de notar que la conversión carga - voltaje se hace en cada píxel por separado [22].

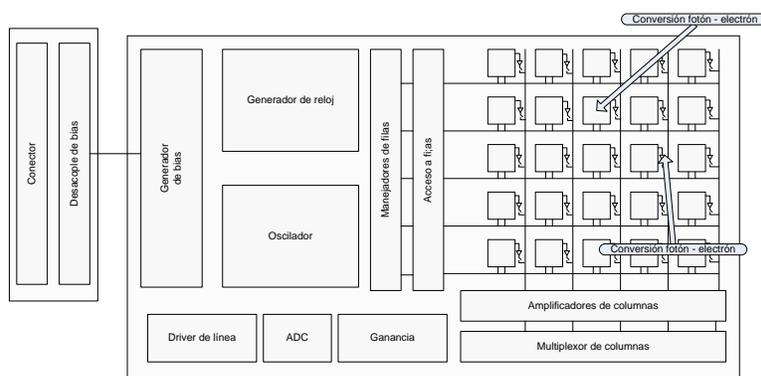


Figura 11. Sensor tipo CMOS. Fuente [22]

En la Tabla 1 se presenta una comparación entre los dos tipos de sensores. Los aspectos a comparar son nueve [22]:

- Respuesta. Cantidad de señal que envía el sensor en función de la energía óptica.
- Rango dinámico. Amplitud de la señal del pixel entre saturación a su nivel umbral mínimo.
- Uniformidad. Similitud de respuesta de dos pixeles con idéntica iluminación.
- Obturación. La capacidad del sensor para detener y volver a comenzar la exposición arbitrariamente.
- Velocidad. Velocidad de respuesta del sensor
- Ventanas. Habilidad del sensor de leer solamente una porción de los pixeles presentes en la escena.
- Antiblooming. Habilidad del sensor para drenar las sobrexposiciones de un pixel, sin comprometer el resto de pixeles.
- Bias y reloj. Tipo de bias y de reloj que necesitan los sensores.
- Confiabilidad. Inmunidad del sensor a daños

Tabla 1. Comparación entre sensores de adquisición de imágenes. Fuente [22]

Atributo	Sensor CCD	Sensor CMOS
Respuesta	Los amplificadores anexos al integrado de sensado permiten una ganancia con un coste de potencia significativo, problema que se está resolviendo por medio de nuevas técnicas de amplificación	Tienen una ventaja marginal ya que los elementos de amplificación son más fáciles de colocar en el integrado.
Rango Dinámico	Posee una clara ventaja, por su eliminación de ruido debido a los sustratos del sensor, tolerancia a las variaciones de capacitancia de bus, amplificadores con geometrías similares adaptadas para mínimo ruido.	Mejora de óptica, mejora de resolución electrónica externa adaptada.
Uniformidad	Con sus amplificadores realimentados tiene un gran ventaja pues son más uniformes tanto en claridad como en oscuridad	Debido a sus amplificadores en lazo abierto los sensores CMOS tienen el peor desempeño tanto en el escenario de uniformidad en claridad y oscuridad

Obturación	Los sensores CCD tienen una obturación electrónica lo que da una ventaja sobre los CMOS. Además debido a su electrónica externa no existe un gran compromiso entre el número de sensores y la obturación.	En los sensores tipo CMOS hay un compromiso entre la obturación y la cantidad de sensores. Para obtener una mejor obturación se necesita adicionar transistores, lo cual hace que la cantidad de sensores se disminuya notablemente.
Velocidad		La ventaja de estos sensores es clara de estos sensores pues toda su funcionalidad está inmersa en un solo integrado lo que hace que los retardos por inductancias y capacitancias de propagación sean menores
Ventanas		Otro acierto de los sensores CMOS. Habilidad única en estos que permite mayores tasa de cuadros por segundo para pequeñas regiones de interés
Antiblooming	CCD necesita circuitería adicional para cumplir con esto	Estos sensores tienen una inmunidad a este fenómeno
Bias y reloj	Típicamente los sensores CCD requieren BIAS un poco más altos. La circuitería de reloj ha sido simplificada en el transcurso del tiempo.	Ventaja para los CMOS pues solo requieren una señal de BIAS y nivel de reloj.
Confiabilidad	Mejor desempeño de imágenes. Más adaptables a diferentes escenarios. Al no ser sensores integrados el tiempo de desarrollo es mucho menor.	Todas las funciones en un solo integrado, lo cual hace un poco más inmune a los daños por puntos de soldadura. Mayor integración lo que hace sensores más pequeños

### Selección del Dispositivo de captura de imágenes

En términos de calidad de imagen, la mejor opción es un sensor tipo CCD por su bajo ruido y su alta uniformidad de imagen; sin embargo es importante anotar que se buscó un hardware de bajo costo, tamaño reducido y fácil adquisición en el mercado, razón por la cual se optó por un sensor tipo CMOS.

La cámara web seleccionada fue un modelo **GENIUS VideoCAM GE111**, este dispositivo genera una señal típica de un sensor CMOS (ruido y baja uniformidad de luminancia). La ficha técnica se muestra en la **Tabla 2**.

**Tabla 2.** Ficha técnica de la cámara web

Característica	Observación
Ancho	8.5 cm
Profundidad	9.3 cm
Alto	7.6 cm
Peso	35 g
Tipo de imagen	RGB – Color
Formato de video digital	AVI
Imagen fija	JPEG 640x480
Tipo de sensor	CMOS
Ajuste de foco	Manual
Interfaz con el PC	USB
Requisitos de sistema	Windows 98SE/2000/ME/XP - Pentium II - 400 MHz - RAM 64 MB - DD 25 MB

Fuente: [23]

La principal limitación con esta elección es un aumento en el tiempo de procesamiento del algoritmo de detección de puntos de aprehensión, ya que se hace necesaria una fase de eliminación de ruido.

### Calibración del Dispositivo

Según [24], “El proceso de calibración de una cámara consiste en estimar los parámetros intrínsecos y extrínsecos de la misma”; así, la calibración es un proceso necesario para hacer la reconstrucción 3D de un objeto o cuando se realiza sensorización por medio del procesamiento de imágenes.

Por el alcance del trabajo (determinar puntos de aprehensión estable) no se requiere emplear técnicas de calibración, éstas deberán usarse en futuros proyectos cuando se estudie la definición de trayectorias de desplazamiento para los dedos de la prótesis.

## 3.2 Condiciones ambientales

La iluminación es un factor importante en la adquisición de imágenes, ya que afecta el grado de complejidad de los algoritmos de procesamiento de imágenes [25].

Para el desarrollo del proyecto se utilizó iluminación semicontrolada; consiste en combinar la iluminación existente en un cuarto y un haz altamente direccional [25]. (ver Figura 12).

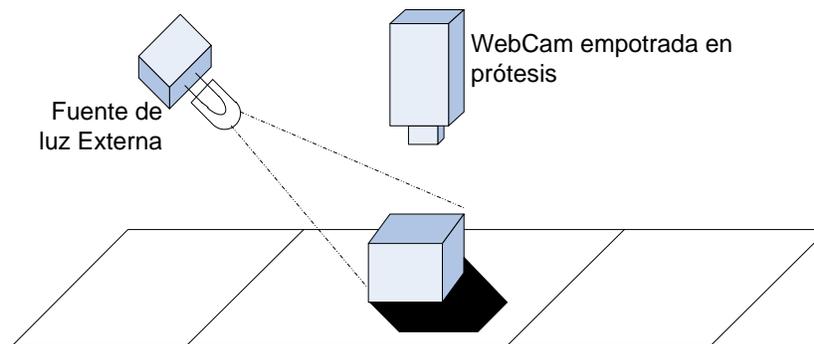


Figura 12. Iluminación Semicontrolada [25]

La selección se soporta en el compromiso entre simplicidad de montaje y sencillez en el procesamiento. El principal inconveniente en este tipo de iluminación es la presencia de sombras, fenómeno que se puede evitar pero requiere de un montaje complejo y costoso, como por ejemplo la iluminación difusa o la iluminación en contraluz.

## Capítulo 4. Algoritmo para la selección de objetos en una escena

A continuación se describen las técnicas de procesamiento de imágenes digitales utilizadas en la identificación y selección del objeto de interés en una escena. En la Figura 13 se muestra el correspondiente diagrama de flujo.

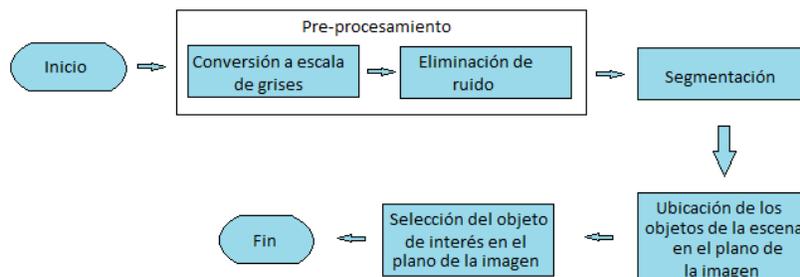


Figura 13. Diagrama de flujo del algoritmo para selección de objetos en una escena

### 4.1 Preprocesamiento

De acuerdo con el diagrama de flujo presentado en la Figura 13, la etapa de pre-procesamiento recibe como señal de entrada la imagen capturada por la webcam. Dicha imagen requiere de una conversión a escala de grises en la cual se elimina el color de manera que queda reducida a información de intensidad lumínica en cada píxel.

La eliminación de ruido corresponde a la eliminación de puntos espúreos, producidos por errores en el sensor, y además se hace una normalización de la intensidad lumínica

Como resultado, la imagen que aporta la webcam es convertida a una imagen en escala de grises, normalizada y libre de ruido.

### 4.1.1 Conversión a escala de grises

Como se mencionó anteriormente, la captura de imágenes fue hecha en color, y a fin de limitar la cantidad de información y mejorar el tiempo de procesamiento, se hace una conversión a escala de grises. Con esta operación se pierde la información de color, lo cual no es relevante para el algoritmo ya el procesamiento está basado en la componente de intensidad de la imagen (condición para los objetos según apartado 1.1).

Para adquirir los datos de la intensidad de la imagen, se usa la ecuación de obtención de la componente de intensidad a fin de realizar la conversión a escala de grises (**Ec. 4**):

$$I = 0.2989 * R + 0.5870 * G + 0.1140 * B \quad \text{Ec. 4}$$

En la ecuación 1, R, G y B son las componentes roja, verde y azul de la imagen original.

### 4.1.2 Eliminación de ruido

El ruido se define como degradación en la calidad de la imagen que se presenta por diferentes causas como por ejemplo, la naturaleza del sensor, interferencias en la transmisión de la imagen, compresión, etc. [28]. Esta pérdida de información debe ser reparada y para ello se usan algoritmos de restauración que dependen de la naturaleza de la pérdida y las causas de su origen.

Los tipos de ruido son variados y provienen principalmente del sistema de adquisición: calidad de los sensores utilizados, desempeño de éstos ante las diferentes situaciones medioambientales, procesos de cuantización, transmisión, iluminación, etc. [28].

Para la eliminación del ruido se utilizan algoritmos heurísticos que manipulan la imagen de manera tal que el sistema de visión adquiera mejor las características de la imagen. Pero ¿Qué es mejorar una imagen?, dar una respuesta a esta pregunta es un asunto complicado, debido a la subjetividad que significa describir la calidad de una imagen [28].

En la Figura 14(b) se puede ver el efecto del denominado ruido *sal y pimienta* (se observan puntos distribuidos al azar). Posiblemente este tipo de ruido se considere importante para una operación como por ejemplo el cálculo del gradiente; sin embargo, si lo que se requiere es la extracción del borde del brazo, las sombras son indeseables y representan un problema mayor; aunque no necesariamente las sombras son una fuente de ruido, existen aplicaciones donde las sombras son utilizadas para la extracción de información. En términos generales, se puede decir que el ruido es función de la operación que se va a realizar.



Figura 14. Demostración del efecto del ruido.

Según el origen del ruido, este se puede clasificar en:

- Ruido impulsivo
- Ruido Fotón-electrónico

De los dos anteriores, fue necesario tratar los efectos del ruido impulsivo en la imagen, también conocido como ruido “Sal y Pimienta”, producido por un sensor ruidoso o a una transmisión errada de la imagen [31]. En la **Figura 14**, la parte (a) muestra la imagen original y la parte (b) presenta la misma imagen con una adición del 5% de ruido impulsivo. Para su eliminación se usa principalmente el filtro de mediana [30], donde el pixel de salida es igual a la mediana de los pixeles de la ventana seleccionada [28].

Otra solución, aunque menos efectiva, es el uso de filtros lineales, como por ejemplo el filtro promedio: el pixel de salida es el promedio de los pixeles de la ventana seleccionada (ver **Figura 15**), es decir, se hace una convolución entre la imagen y una máscara promediadora [28].

$$\frac{1}{\sum_{K=0}^N Z_K}$$

Z1	Z2	Z3	Z4	Z5
Z6	Z7	Z8	Z9	Z10
Z11	Z12	Z13	Z14	Z15
Z16	Z17	Z18	Z19	Z20
Z21	Z22	Z23	Z24	Z25

Figura 15. Máscara para filtro promedio

Una situación adversa que surge al utilizar el filtro promedio es la generación de un ensanchamiento de los bordes de la imagen por lo que su detección se ve afectada [28]. Este problema es menor con el filtro de mediana [28], y por esta razón fue utilizado en el proyecto.

Se realizaron varias pruebas para justificar el uso del filtro de mediana; se encontró que para imágenes claramente segmentables como la mostrada en la **Figura 16(a)** no es necesaria, pero para el caso de imágenes cuya segmentación no es tan inmediata como en la **Figura 16(b)**, es esencial.



**Figura 16.** Ejemplos de Imágenes ruidos. (a) Facilmente Segmentable. (b) Dificilmente Segmentable

## 4.2 Segmentación

Proceso en el cual se identifican y separan los objetos de interés que se encuentran en una imagen. Los algoritmos de segmentación se clasifican en: los algoritmos basados en similitudes y los basados en discontinuidades [28].

Dentro del primer grupo se encuentra la umbralización, que permite separar los píxeles que constituyen el fondo de la imagen de los que conforman los objetos dentro de la misma [26]. Otro algoritmo usado es detección de bordes (explicado en la sección 5.1 ya que constituye la definición de una característica del objeto).

En la operación de umbralización, la entrada es una imagen en escala de grises ( $f[x,y]$ ) y el resultado es una imagen binaria ( $g[x,y]$ : compuesta por “unos” y “ceros”). Como se aprecia en la Figura 17, el primer paso es la definición del umbral, tarea fundamental en el algoritmo, que se detallará más adelante; luego se compara el pixel de entrada con el valor del umbral: si el valor de pixel es menor entonces se asigna cero al pixel de salida (en la misma posición), en caso contrario se establecerá un “1” como salida. El procesamiento termina cuando han sido comparados todos los píxeles de la imagen. Cabe recalcar que en una imagen de 8 bits de resolución un “1” equivale a un valor de 255, correspondiente a una intensidad de máximo brillo o “blanco”.

Las principales ventajas de este algoritmo son:

- Una vez se ha elegido el umbral, es sencillo y presenta bajo costo computacional.
- Es muy eficiente para procesar imágenes con suficiente contraste.
- Da múltiples opciones para realizar la segmentación aprovechando características como: color, entropía, forma, histograma, entre otras.

Entre las desventajas se tienen:

- Los algoritmos para el cálculo de umbral pueden resultar complejos, por la gran cantidad de operaciones matemáticas que pueden involucrar.
- Para imágenes con poco contraste puede no ser útil.

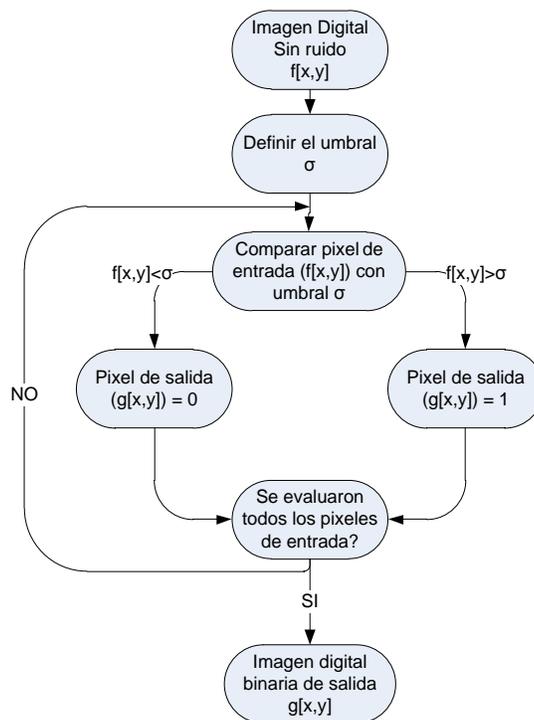


Figura 17. Algoritmo básico para umbralización

Para determinar el valor del umbral  $\sigma$  se usa el histograma de la imagen y algunas otras características estadísticas. En la literatura es posible encontrar diferentes métodos para tal fin [26]:

- Métodos basados en la forma del histograma
- Métodos basados en agrupamientos
- Métodos basados en entropía
- Métodos basados en atributos de los objetos
- Métodos espaciales
- Métodos adaptativos

En el proyecto se usaron dos algoritmos para definir el valor del umbral, el primero es el propuesto por Otsu [34] que es una implementación basada en agrupamientos, y el otro se basa en el momento estadístico curtosis; durante la ejecución de la aplicación es posible elegir cuál de los dos emplear.

El algoritmo de Otsu fue utilizado a fin de considerar un referente de comparación con el algoritmo basado en curtosis, implementado en el desarrollo de este proyecto; el primero (algoritmo de Otsu) está implementado por defecto en MatLab bajo la instrucción *graythresh*.

**Algoritmo de Otsu**

Basándose en la información proporcionada en [34] se tiene: suponiendo una imagen en escala de grises, con  $N$  píxeles y niveles de gris desde 1 hasta  $L$  (255 para este caso, pues se están utilizando imágenes de 8 bits de resolución), el número de píxeles que tienen nivel de gris  $i$  se denotarán como  $N_i$ , la probabilidad de ocurrencia de un nivel de gris en la imagen es:

$$p_i = \frac{N_i}{N} \quad \text{Ec. 5}$$

Para una agrupación en dos clases se tiene: a la clase uno ( $C_1$ ) pertenecen todos los píxeles con un nivel de gris entre 1 y  $t$  ( $[1, \dots, t]$ ) y en la clase dos ( $C_2$ ) se encuentran aquellos con valores entre  $t+1$  y  $L$ . Dada esta información, las distribuciones de probabilidad, se definen como:

$$C_1 : \frac{p_1}{\omega_1(t)}, \dots, \frac{p_t}{\omega_1(t)} \quad \text{Ec. 6}$$

$$C_2 : \frac{p_{t+1}}{\omega_2(t)}, \frac{p_{t+2}}{\omega_2(t)}, \dots, \frac{p_L}{\omega_2(t)} \quad \text{Ec. 7}$$

Los valores esperados para cada una de las clases serán:

$$\mu_1 = \sum_{i=1}^t i \frac{p_i}{\omega_1(t)} \quad \text{Ec. 8}$$

$$\mu_2 = \sum_{i=t+1}^L i \frac{p_i}{\omega_2(t)}$$

Con esta definición se tiene que el valor esperado para la distribución de toda la imagen se dará en el siguiente caso:

$$\omega_1 \mu_1 + \omega_2 \mu_2 = \mu_T \quad \text{Ec. 9}$$

$$\omega_1 + \omega_2 = 1$$

La varianza entre clases para la imagen umbralizada se define como:

$$\sigma_B^2 = \omega_1 (\mu_1 - \mu_T)^2 + \omega_2 (\mu_2 - \mu_T)^2 \quad \text{Ec. 10}$$

Otsu demostró que el umbral óptimo para una clasificación en dos conjuntos está en el máximo de la Ec. 10, por lo tanto el umbral óptimo para la imagen será:

$$\sigma = \text{Arg} \left( \text{Max} \left( \sigma_B^2(t) \right) \right) \quad 1 \leq t < L \quad \text{Ec. 11}$$

**Algoritmo basado en curtosis**

A continuación se describe el proceso.

*Momento estadístico*

Los momentos son medidas de las características estadísticas de distribuciones de datos y tienen múltiples usos [32]. Los momentos de orden  $h$  respecto al origen de una variable estadística se definen según la Ec. 12 [32].

$$a_h = x_1^h \frac{n_1}{N} + x_2^h \frac{n_2}{N} + \dots + x_r^h \frac{n_r}{N} = \sum_{i=1}^r x_i^h \frac{n_i}{N} \tag{Ec. 12}$$

Los momentos se pueden centralizar con respecto a la media aritmética. Es así como después de centralizar los momentos descritos por la Ec. 12 se convierte en la Ec. 13 [32]:

$$m_h = (x_1 - \bar{x})^h \frac{n_1}{N} + (x_2 - \bar{x})^h \frac{n_2}{N} + \dots + (x_r - \bar{x})^h \frac{n_r}{N} = \sum_{i=1}^r (x_i - \bar{x})^h \frac{n_i}{N} \tag{Ec. 13}$$

Una vez definido que es un momento, se expone como éstos pueden describir la forma de una distribución estadística.

*El índice de curtosis o de apuntalamiento de Fischer*

El índice de curtosis es muy utilizado en el cálculo del umbral óptimo. Es un momento central de 4<sup>to</sup> orden y compara la distribución de entrada con la distribución normal.

La comparación consiste en:

- Si el índice de curtosis es igual a cero entonces la distribución de datos estudiada tiene la forma de la variable normal.
- Si el índice es mayor que cero la distribución es más apuntada que la normal
- Si el índice es menor a cero la distribución es menos apuntada.

La Ec. 14 muestra la fórmula para realizar el cálculo del índice de curtosis [32].

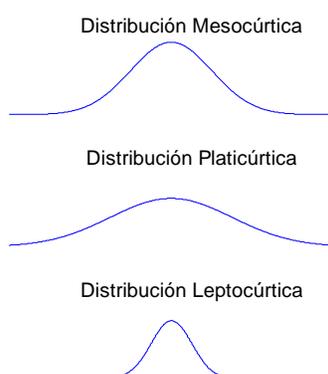
$$K = \frac{\sum_{i=1}^r (x_i - \bar{x})^4 \frac{n_i}{N}}{\sigma_x^4} - 3 \tag{Ec. 14}$$

En la Tabla 3 se aprecia la clasificación de la distribución según su valor de curtosis.

**Tabla 3.** Clasificación distribuciones según su curtosis

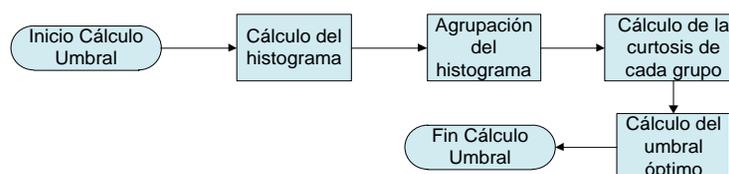
Índice Curtosis	Denominación Distribución
=0	Mesocúrtica
<0	Platicúrtica
>0	Leptocúrtica

Como se ve en la Figura 18, el índice de curtosis tiene una noción aproximada del ancho de una distribución.



**Figura 18.** Clasificación de distribuciones según su curtosis

En la Figura 19 se muestra el diagrama de flujo del proceso de cálculo del umbral.



**Figura 19.** Proceso de cálculo del umbral

A continuación se explica cada fase:

- Se establece el histograma de la imagen, para determinar la distribución de los valores de intensidad lumínica.
- Se agrupa el histograma en 8 clases de 32 valores diferentes cada una, como se observa en la **Tabla 4**; esta categorización se realiza teniendo en cuenta los máximos comunes divisores de 256 (número máximo de posibles valores que puede tomar la intensidad lumínica de un píxel en una imagen; sus máximos

comunes divisores son 1, 2, 4, 8, 16, 32, 64, 128 y 256), evitando la conformación de grupos pequeños que harían el procesamiento ineficiente.

**Tabla 4.** División histograma proceso cálculo umbral óptimo

Grupo	Valores
1	000-031
2	032-063
3	064-095
4	096-127
5	128-159
6	160-191
7	192-223
8	224-255

- Se obtiene la curtosis para cada uno de los grupos de la Tabla 4; el resultado de la operación es la función que describe la variación de la curtosis a través del histograma. Al evaluar el mínimo de la segunda derivada se obtiene el segmento de histograma donde se presentan cambios de mayor a menor valor en la función de curtosis.
- Se fija el umbral óptimo por medio del cálculo del mínimo valor de intensidad lumínica del grupo seleccionado. Este valor no puede ser determinado por posición (primer o último valor de la clase) ya que puede pertenecer al principio o al final de un pico. De lo anterior se puede evidenciar que el mínimo valor marca la diferencia entre el descenso final de un pico y el ascenso inicial del siguiente.

Obtenido el valor del umbral, el proceso de binarización da como resultado una segmentación inicial de la imagen en dos clases: fondo y objetos, en donde los píxeles marcados como “1” son los pertenecientes al objeto, y los marcados como “0” conforman el fondo. Dado que el número de objetos en la escena puede ser mayor que uno, debe existir una forma de seleccionar uno solo de ellos.

Este algoritmo de umbralización fue usado por su robustez en la discriminación de las sombras. En el capítulo de resultados se hace una comparación entre el algoritmo presentado aquí y el de Otsu (se encuentra implementado en MATLAB).

Cabe destacar que, cuando los objetos son más oscuros que el fondo de la imagen, el resultado de la segmentación puede alterarse y requerirá una inversión de la imagen antes de concluir la operación para unificar el resultado de la operación.

### 4.3 Cálculo del centro de masa y la orientación

Suponiendo que el centro de masa se encuentra ubicado en  $X_{cm}$  y  $Y_{cm}$ , su cálculo se hace mediante las siguientes ecuaciones:

$$X_{cm} = \frac{m_{10}}{m_{00}} \quad \text{Ec. 15}$$

$$Y_{cm} = \frac{m_{01}}{m_{00}} \quad \text{Ec. 16}$$

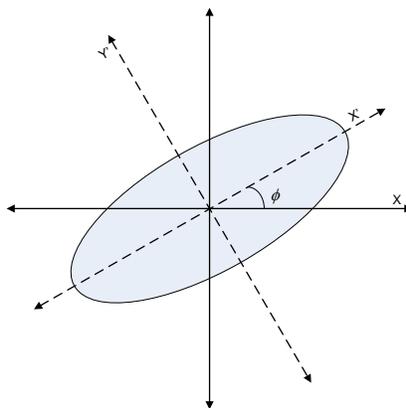
Donde  $m_{10}$ ,  $m_{01}$  y  $m_{00}$  son los momentos geométricos de orden 10, 01 y 00 respectivamente, sus expresiones matemáticas, para una imagen de tamaño  $M \times N$ , se describen en Ec. 17, Ec. 18 y Ec. 19 respectivamente

$$m_{10} = \sum_{x=0}^M \sum_{y=0}^N x * f(x, y) \quad \text{Ec. 17}$$

$$m_{01} = \sum_{x=0}^M \sum_{y=0}^N y * f(x, y) \quad \text{Ec. 18}$$

$$m_{00} = \sum_{x=0}^M \sum_{y=0}^N f(x, y) \quad \text{Ec. 19}$$

La orientación es el ángulo comprendido entre la abscisa y el eje de mínima inercia como se aprecia en la **Figura 20**.



**Figura 20.** Definición de ángulo de orientación

De acuerdo a lo anterior, el ángulo  $\phi$  se define por:

$$\phi = \frac{1}{2} \tan^{-1} \left( \frac{2 * mc(1,1)}{mc(2,0) - mc(0,2)} \right) \quad \text{Ec. 20}$$

### 4.4 Determinación del objeto de interés en la escena

Para la elección del objeto, se requiere primero etiquetar cada uno de los cuerpos presentes en la imagen; este proceso se realiza por medio del método de conectividad de píxeles, que se puede aplicar ya que las imágenes son binarias.

Para el desarrollo del proyecto se consideró una vecindad-4 (los 4 pixeles adyacentes), ya que su definición es más sencilla que la vecindad diagonal y la vecindad mixta, además evita algunas ambigüedades que introduce la vecindad-8.

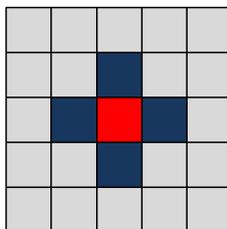


Figura 21. Definición de vecindad-4

De acuerdo a la Figura 21, los cuadros azul oscuro representan la vecindad del píxel rojo, cuya intensidad lumínica es representada por  $f(x,y)$  donde  $x,y$  son sus coordenadas. La Ec. 21 es la expresión matemática de las coordenadas de la vecindad-4.

$$(x+1, y), (x-1, y), (x, y+1), (x, y-1) \tag{Ec. 21}$$

En una imagen binaria, dos pixeles están conectados si y solo si uno de ellos está en la vecindad del otro, en el sentido de vecindad-4 es importante anotar que los dos pixeles deben tener el mismo valor de luminosidad.

Un objeto está definido como el conjunto de pixeles que se encuentran conectados unos con otros y tienen un valor de "255" (los pixeles del fondo tiene valor 0). Para determinar las etiquetas, se busca un píxel perteneciente a un objeto en la imagen original, se examinan sus vecinos, después los vecinos de los vecinos y así sucesivamente hasta que no hayan más elementos, en este momento se crea una nueva imagen que contiene el objeto (etiquetado como "objeto 1"), en la imagen original los pixeles que han sido procesados toman valor cero; el proceso se repite hasta terminar con todos los objetos; el criterio para finalizar el algoritmo es cuando la matriz solo contiene ceros.

Como resultado se obtiene un arreglo, del mismo tamaño que la imagen original, cuyos valores son las etiquetas de los objetos. Con esta matriz se calculan las coordenadas del centro de masa cada elemento individual.

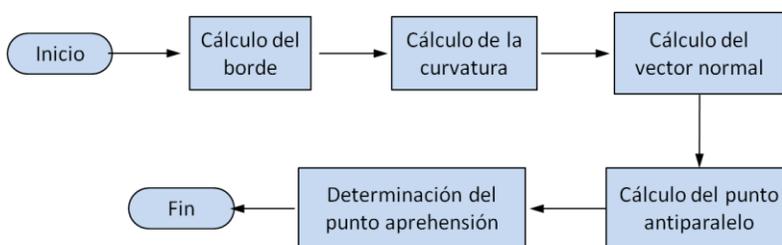
Finalmente, se calculan las distancias de los centros de masa de cada objeto al punto medio de la imagen original. La menor de ellas corresponde al objeto seleccionado como objetivo para el agarre.

Si existe la necesidad de hacer una inversión (debido al procesamiento), el proceso se realiza en este punto para obtener una imagen estándar.



## Capítulo 5. Algoritmo para Identificación de Puntos de Agarre a Partir de una Escena.

Ahora se explican los procesos realizados para reconocer los puntos de agarre de un objeto presente en una imagen. En la Figura 22 se muestra el correspondiente diagrama de flujo.



**Figura 22.** Diagrama de flujo del algoritmo para identificación de puntos de agarre

### 5.1 Detección de bordes

La base del algoritmo es la detección de discontinuidades de la función de intensidad lumínica. En las escenas captadas, las intensidades de los píxeles pueden cambiar sutil o bruscamente, de ahí que la extracción de bordes consista en localizar los cambios abruptos en la luminosidad de los píxeles en una imagen [20]; lo anterior se justifica al considerar que cada objeto en la escena, cada marca en la superficie y cada sombra, genera un borde [20][28][29].

Para elegir el algoritmo de detección de bordes apropiado se consideraron tres aspectos:

- El borde debe ser cerrado.
- El borde debe tener un ancho de 1 píxel en todos sus puntos.
- Menor carga computacional.

Algunos métodos para realizar este proceso son:

- A través del cálculo de la primera derivada local de la función de intensidad lumínica [20].
- Usando la segunda derivada local de la función de intensidad lumínica [20].
- Empleando la operación morfológica de erosión.

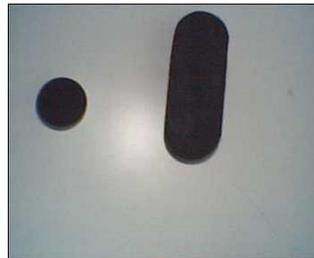
Para hacer el cálculo de la primera y la segunda derivada, se utilizaron máscaras (matrices especialmente calculadas para tales fines) [28]; éstas se operan con la imagen a través de la convolución discreta en dos dimensiones, definida como:

$$h(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t) \quad \text{Ec. 22}$$

Donde

- $w$ : máscara calculada
- $f$ : imagen de entrada (en escala de grises)
- $h$ : imagen de salida

Para obtener la imagen de salida, se debe seleccionar uno de los operadores (tienen la forma de una matriz) de gradiente o de Laplaciano ( $w$  en la Ec. 22). Para visualizar el resultado de aplicar cada uno de los operadores será utilizada la imagen mostrada en la Figura 23.



**Figura 23.** Imagen base para cálculo de bordes

### Operador de Sobel

Utiliza dos máscaras que se basan en las siguientes fórmulas (para el caso 3x3) [20]:

$$H_{sobel} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad \text{Ec. 23}$$

$$G_x = (a_{13} + 2a_{23} + a_{33}) - (a_{11} + 2a_{21} + a_{31}) \quad \text{Ec. 24}$$

$$G_y = (a_{31} + 2a_{32} + a_{33}) - (a_{11} + 2a_{12} + a_{13}) \quad \text{Ec. 25}$$

Aplicando las Ec. 24 y Ec. 25 en la Ec. 23 se obtienen las máscaras de Sobel para el cálculo de los bordes a través de convolución [28][29]:

$$H_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{Ec. 26}$$

$$H_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \text{Ec. 27}$$

### Operador de Prewitt

El funcionamiento del operador básicamente es el mismo que el operador de Sobel, la diferencia está en que las máscaras no hacen énfasis en el pixel del centro, por lo que el resultado de la operación es diferente. Las máscaras para el operador de Prewitt son [29]:

$$H_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{Ec. 28}$$

$$H_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{Ec. 29}$$

### Operador de Roberts

Las máscaras para el operador de Roberts se presentan en las Ec. 30 y Ec. 31 [29]. En la Figura 26 se presenta el resultado de la operación de detección

$$H_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{Ec. 30}$$

$$H_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \text{Ec. 31}$$

## Operador de Robinson

Las máscaras para el operador de Robinson se presentan en las Ec. 32 y Ec. 31 [29]. En la Figura 26 se presenta el resultado de la operación de detección

$$H_x = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix} \quad \text{Ec. 32}$$

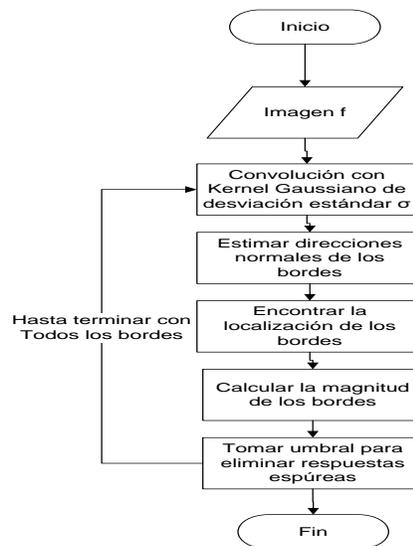
$$H_y = \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix} \quad \text{Ec. 33}$$

## Algoritmo de Canny

Canny propuso un algoritmo que pretende optimizar la detección de bordes en ambientes ruidosos, se basa en tres criterios muy claros [29]:

1. El criterio de detección: ningún borde importante debe ser perdido.
2. El criterio de localización: la distancia entre el borde detectado y el borde real sea mínima
3. El criterio de respuesta única: todo borde debe tener solo una respuesta.

Las tareas del detector de Canny se muestran en la Figura 24; este algoritmo genera bordes con un ancho de 1 píxel lo que marca la diferencia con los demás. El resultado de la operación se evidencia en la Figura 26.



**Figura 24.** Algoritmo de Canny

**Morfología**

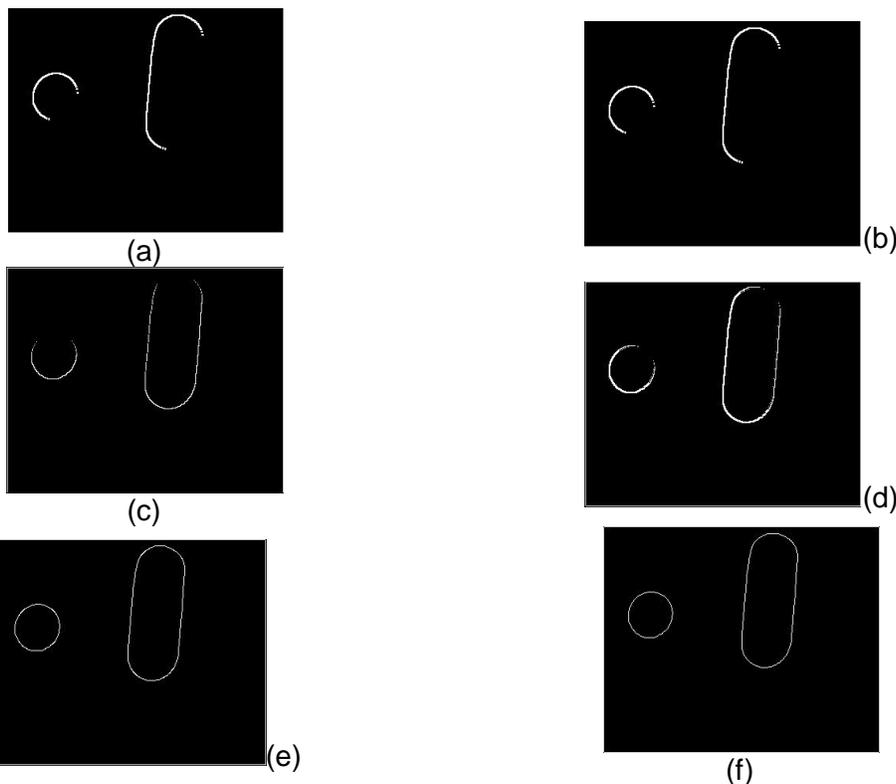
No usa ni máscaras ni convolución, adicionalmente requiere que la imagen sea binaria. Los bordes definidos como  $\beta(x, y)$  son descritos en la Ec. 34

$$\beta(x, y) = A(x, y) - (A(x, y) \ominus B(x, y)) \tag{Ec. 34}$$

Donde  $A(x, y)$  es la imagen,  $B(x, y)$  se conoce como elemento estructurante y  $\ominus$  representa la operación morfológica de erosión. El elemento estructurante fue determinado por un método empírico como se muestra en la Figura 25:

0	1	0
1	1	1
0	1	0

**Figura 25.** Elemento estructurante



**Figura 26:** Resultado de las pruebas para la Figura 23. Aplicando: (a) Algoritmo de Sobel, (b) Algoritmo de Prewitt, (c) Algoritmo de Roberts, (d) Algoritmo de Robinson, (e) Algoritmo de Canny y (f) Algoritmo por Morfología

Todos los algoritmos fueron probados a fin de comparar sus resultados; como se observa en la Figura 26, para algunos algoritmos clásicos de detección de bordes por primera derivada no se generan productos aptos para el desarrollo del proyecto, por tal razón se decidió utilizar el de morfología, ya que entrega un borde cerrado cuyo ancho es de un pixel y con un tiempo de procesamiento menor, esta solución es suficiente para el cálculo de los puntos de aprehensión. Se considera un producto apto aquel en el que el borde es cerrado y el ancho es de un pixel, porque es el primer paso en la descripción del contorno. En la Tabla 5 se hace una comparación de los resultados obtenidos.

**Tabla 5.** Comparación de algoritmos para la detección de bordes

Algoritmo	Tiempo	Resultado
Canny	28.45	Definición de bordes según las dos primeras condiciones.
Sobel		No entrega borde cerrados ni de ancho un pixel
Prewitt		No entrega borde cerrados ni de ancho un pixel
Roberts		No entrega borde cerrados ni de ancho un pixel
Morfológico	26.541	Definición de bordes según las dos primeras condiciones.

Para aquellos casos donde el algoritmo de morfología presente confusión al momento de cerrar el borde (debido a cambios de textura, ruido, entre otros) se utiliza la operación morfológica de cerradura y otra para aquellos objetos que se ubican en el borde de la imagen.

## 5.2 Identificación de puntos de aprehensión

En este punto es fundamental el cálculo de la curvatura de los vectores  $\beta$  y  $\beta'$ , por eso lo primero que se hace es estimar la curvatura de cada uno de los pixeles que componen el borde del objeto seleccionado.

### 5.2.1 Cálculo de la curvatura

El cálculo de la curvatura puede ser explicado a partir de la Figura 27. Se tiene un objeto descrito por su borde, la curvatura  $\alpha$  de un pixel cualquiera se calcula de la siguiente manera: se crean dos vectores  $A$  y  $B$ , que van desde el pixel de prueba hasta  $n$  pixeles más adelante ( $A$ ) y  $n$  pixeles más atrás ( $B$ ).

La curvatura del pixel está dada por el ángulo comprendido entre los vectores  $A$  y  $B$ . Según [2], los  $n$  pixeles hacia adelante o hacia atrás dependen de la cantidad de pixeles que forman el borde, el número más común de pixeles es tres, valor determinado por experimentación.

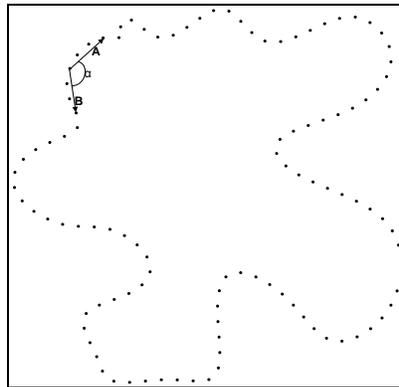


Figura 27. Definición de curvatura

### 5.2.2 Identificación de los puntos de aprehensión estable para un agarre con dos dedos.

De acuerdo a la Figura 28, se tiene un objeto con centro de masa en el punto  $C$ , descrito por su borde (línea naranja), dentro de ese borde existe un punto  $P$  (será examinado para comprobar su pertenencia a las zonas de aprehensión). Existen dos vectores  $\vec{P}$  (comprendido entre el origen y el punto  $P$ ) y  $\vec{C}$  (comprendido entre el origen y el punto  $C$ ), la curvatura en el punto  $P$  está dada por el ángulo  $\alpha$ . Se define el vector  $\vec{N}$  como el vector normal a la superficie del objeto en el punto  $P$ . El vector  $\vec{A}$  se define como el vector desde el punto  $P$  al punto  $C$ .  $\beta$  es el ángulo comprendido entre los vectores:

$$\beta = \angle \vec{A} \vec{N} \quad \text{Ec. 35}$$

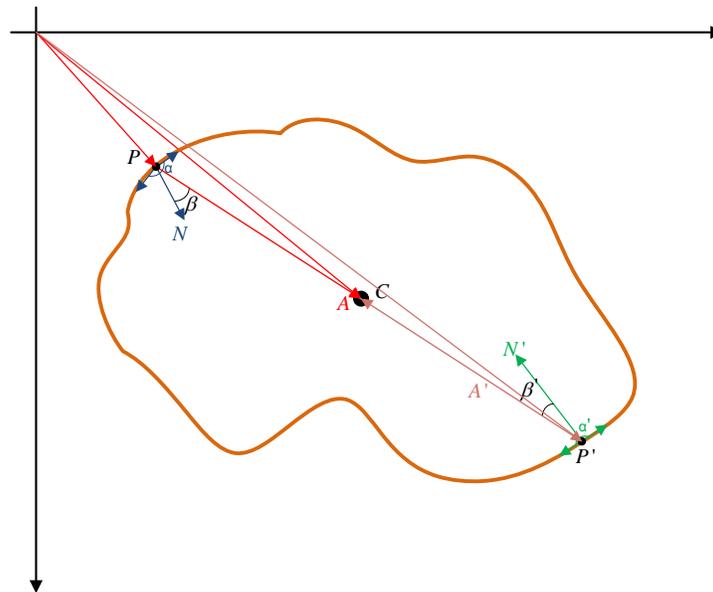


Figura 28. Cálculo de  $\beta$  y  $\beta'$

Según lo anterior, antes de poder calcular  $\beta$  se deben calcular  $\vec{A}$  y  $\vec{N}$ .

El cálculo de  $\vec{A}$  se basa en la resta de vectores como lo indica Ec. 36

$$\vec{A} = \vec{C} - \vec{P} \quad \text{Ec. 36}$$

Para el cálculo de  $\vec{P}$  y  $\vec{C}$  se hace uso de los pares ordenados de los puntos  $P$  y  $C$  respectivamente.

El cálculo del vector  $\vec{N}$  se hace mediante el uso de la curvatura  $\alpha$ . Si el par ordenado que describe a  $\vec{N}$  es  $x_N, y_N$  entonces:

$$\begin{aligned} x_N &= \cos\left(\frac{\alpha}{2}\right) \\ y_N &= \sin\left(\frac{\alpha}{2}\right) \end{aligned} \quad \text{Ec. 37}$$

Una vez los vectores  $\vec{A}$  y  $\vec{N}$  han sido calculados, se procede a determinar el ángulo  $\beta$ . Para esto se hace uso del producto punto:

$$\beta = \cos^{-1}\left(\frac{\vec{A} \cdot \vec{N}}{\|\vec{A}\| \cdot \|\vec{N}\|}\right) \quad \text{Ec. 38}$$

Para hacer el cálculo de  $\beta'$  es muy importante determinar el punto  $P'$ . Este punto es antiparalelo a  $P$ . Por esta característica es posible utilizar el vector  $\vec{A}$ , con éste se hace un producto punto con todos los posibles vectores  $\vec{A}'$  de todos los posibles puntos  $P'$  en el borde del objeto, ahora se puede hacer el cálculo de todos los posibles ángulos entre  $\vec{A}$  y  $\vec{A}'$ , el máximo valor de ángulo es el correspondiente al  $\vec{A}'$  del punto  $P'$  que es antiparalelo a  $P$ . A este punto  $P'$  se le calcula el vector  $\vec{N}'$  y del ángulo  $\beta'$  mediante la ecuación:

$$\beta' = \cos^{-1}\left(\frac{\vec{A}' \cdot \vec{N}'}{\|\vec{A}'\| \cdot \|\vec{N}'\|}\right) \quad \text{Ec. 39}$$

Usando la Ec. 3, se hace la comparación para obtener los puntos de aprehensión (ver apartado 2.2.2). Para esto se utiliza un coeficiente estático de rozamiento igual a 0.56 debido al recubrimiento que presentan los dedos de la prótesis.

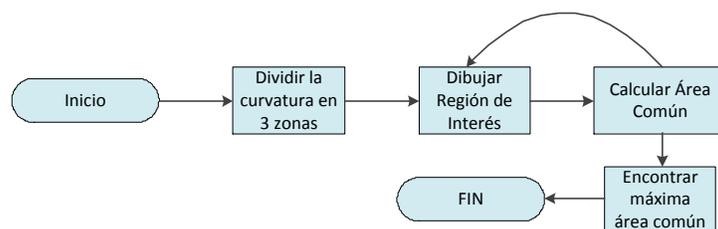
Una vez se identifican todos los posibles puntos de aprehensión, son seleccionados solo algunos de ellos, teniendo en cuenta lo dicho por Montana en [5]: se eligen aquellos en los cuales la distancia entre los dos posibles puntos sea la mínima de todos.

### 5.2.3 Identificación de los puntos de aprehensión estable para un agarre con tres dedos

Para resolver este punto se tuvieron en cuenta las siguientes consideraciones:

- Se determinaron contactos puntuales (de tamaño un pixel), para disminuir el tiempo de computo.
- Los vectores normales (construidos anteriormente) fueron reutilizados.
- Para tener soluciones alternas, el ángulo del cono de fricción fue entregado como un parámetro del algoritmo de cálculo de puntos de aprehensión.

El algoritmo consta de cuatro fases, como se ve en la Figura 29:



**Figura 29.** Algoritmo para cálculo de puntos de aprehensión en agarres con tres dedos

- Dividir la curvatura en tres zonas iguales y tomar el primer pixel de cada zona, esta es fue una manera sencilla que garantizó que los tres vectores normales cubrieran al espacio  $\mathbb{R}^2$ .
- Dibujar las zonas de interés: corresponden a los conos de fricción en cada uno de los puntos ya seleccionados. Para hacer el dibujo fue necesario introducir el ángulo de separación entre la normal y una de las líneas punteadas respectivas;  $t$ . Teniendo en cuenta la **Figura 6**, el ángulo a introducir es  $\theta$ . Ver Figura 9. Diagrama para aprehensión con tres dedos
- Con los tres conos de fricción se hizo el cálculo del área común de ellos. Posteriormente se tomó el segundo pixel de cada zona y se volvió a dibujar la región de interés, así sucesivamente hasta terminar todos los puntos de la zona; aunque lo más indicado para este punto sería hacer la combinatoria de los pixeles de las zonas, el cálculo tomaría mucho tiempo. Ejemplo: si la curvatura tuviera 99 pixeles de largo, cada una de las zonas tendría una longitud de 33 pixeles. Por el método anteriormente descrito se requerirán 33 cálculos de área-

as mientras que la solución óptima requerirá de  $33 \times 33 \times 33 = 35937$  cálculos de áreas.

- Se determinó la tripleta de conos (identificados por el punto central) cuya área en común fuera la máxima posible. Los tres puntos de aprehensión son los tres puntos centrales de la tripleta de conos de fricción.

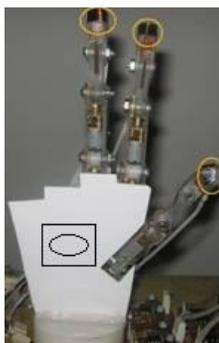
Cabe reiterar que todo el proceso fue posible ya que los tres vectores normales habían sido calculados para realizar el cálculo de puntos de aprehensión para el agarre a dos dedos.

## Capítulo 6. Análisis de resultados

### 6.1 Descripción de la solución al problema

De acuerdo con el planteamiento del problema (ver sección 1.1), se hizo el cálculo de los puntos de aprehensión estable para una prótesis de mano robótica; se escogió la visión artificial como método de captura de información, por la capacidad que tiene de extraer gran cantidad de datos de la escena donde se encuentran los posibles objetos que serán tomados; para esto se coloca una cámara de video (sus características fueron descritas en la sección 3.1) en la palma de la prótesis.

La elección de este lugar como punto de anclaje se sustenta con un proceso de experimentación que buscaba evitar las distorsiones propias de la perspectiva y de la geometría del lente de la cámara. El esquema es muy parecido al mostrado en la Figura 30.



**Figura 30.** Hipótesis de solución para el problema del proyecto

Fuente: Tomada de [http://www.ai.unicauca.edu.co/Proyecto\\_Protesis/Detalles.html](http://www.ai.unicauca.edu.co/Proyecto_Protesis/Detalles.html)

Para cumplir con el objetivo, la captura de la escena exigió total claridad (perfecta diferenciación entre objetos y fondo de la escena) de forma tal que en la máquina no se generaran dudas. Cuando el objeto y el fondo son de color similar (imágenes 055 a 073 de la tabla anexa) se evidenciaron problemas de segmentación. A continuación el algoritmo debe clasificar los píxeles pertenecientes a los objetos y al fondo en la escena, posteriormente selecciona un único cuerpo en la escena y por último establece qué puntos son los más apropiados para la tarea de agarre.

Resumiendo, se tienen las siguientes tareas básicas para cumplir con el objetivo:

1. Realizar la captura de la escena por medio de una cámara
2. Determinar el objeto de interés para el cálculo de los puntos de agarre
3. Establecer las características físicas del objeto de interés
4. Calcular los puntos de aprehensión estable para el objeto seleccionado

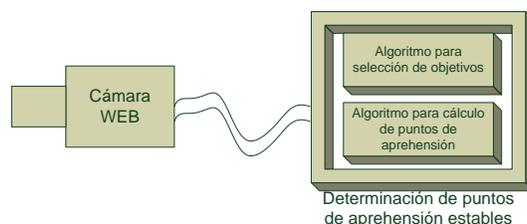


Figura 31. Esquema General de la solución

En la Figura 31, se aprecia el esquema general del proyecto. Una vez la cámara web empieza a generar las imágenes, éstas deben ser procesadas por el algoritmo diseñado, compuesto básicamente por dos grandes bloques funcionales: el algoritmo que permite seleccionar el objetivo de la escena y el algoritmo que calcula los puntos de aprehensión.

## 6.2 Descripción General del Algoritmo

Como se mencionó anteriormente, el algoritmo diseñado tiene dos bloques funcionales: la selección del objetivo y el cálculo de los puntos de aprehensión.

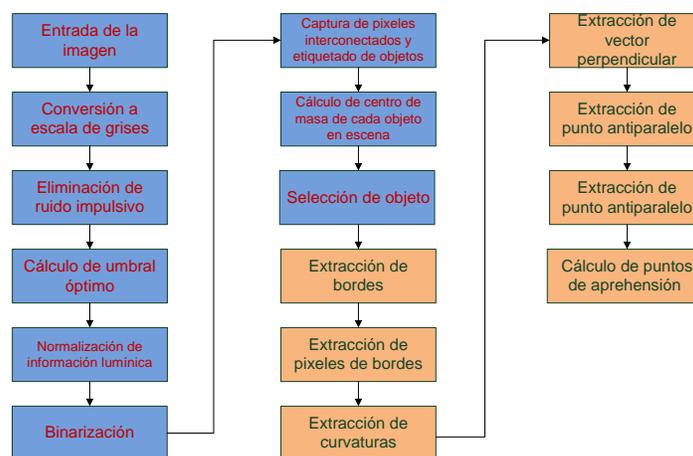


Figura 32. Algoritmo General del proyecto

En el diagrama de la Figura 32 se observa cada uno de los pasos requeridos para hacer el cálculo de los puntos de aprehensión estable. Los procesos azules son los que conforman el algoritmo de selección del objetivo, mientras que los naranjas son los que hacen el cálculo del punto de aprehensión estable.

6.2.1 Algoritmo para seleccionar el objetivo

Los pasos a seguir son:

Usando la cámara web descrita en la Tabla 2, se captura la imagen de entrada (en color ver sección 4.1.1) usando una iluminación semicontrolada (ver 3.1 y 3.2).

Se elimina el ruido a fin de mejorar la calidad de la imagen y hacer una primera fase de normalización de iluminación (se hicieron pruebas sobre ambos tipos de ruido: ruido impulsivo y ruido fotón-electrónico). En la Figura 33 se aprecian los resultados de usar el filtro de mediana.

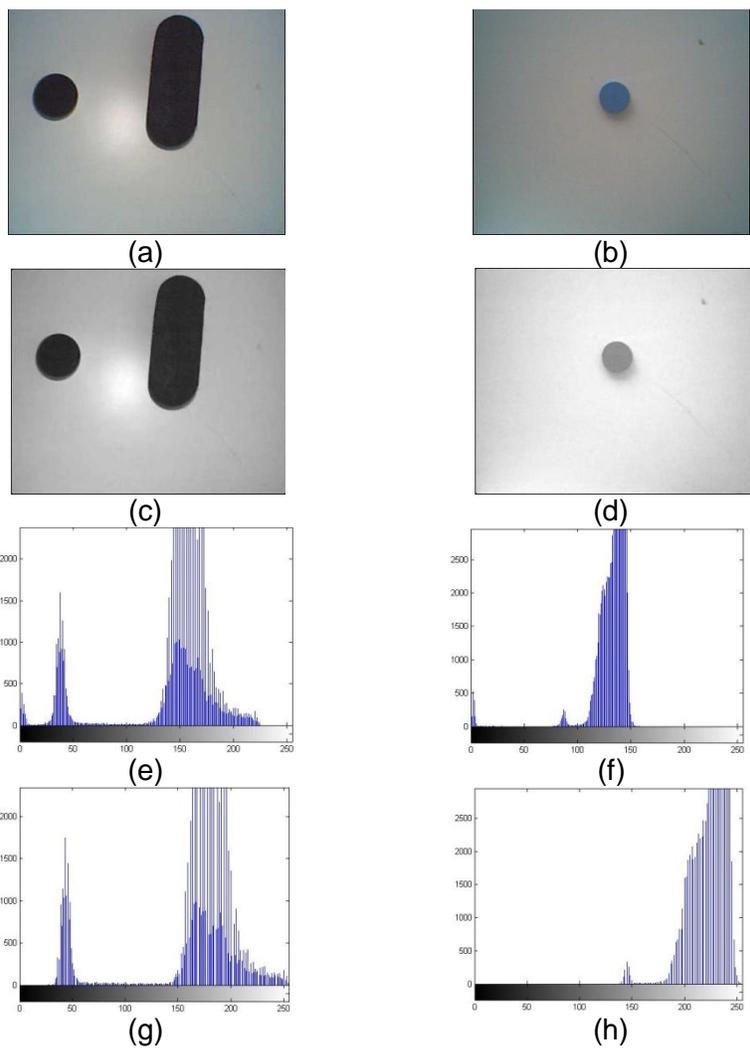


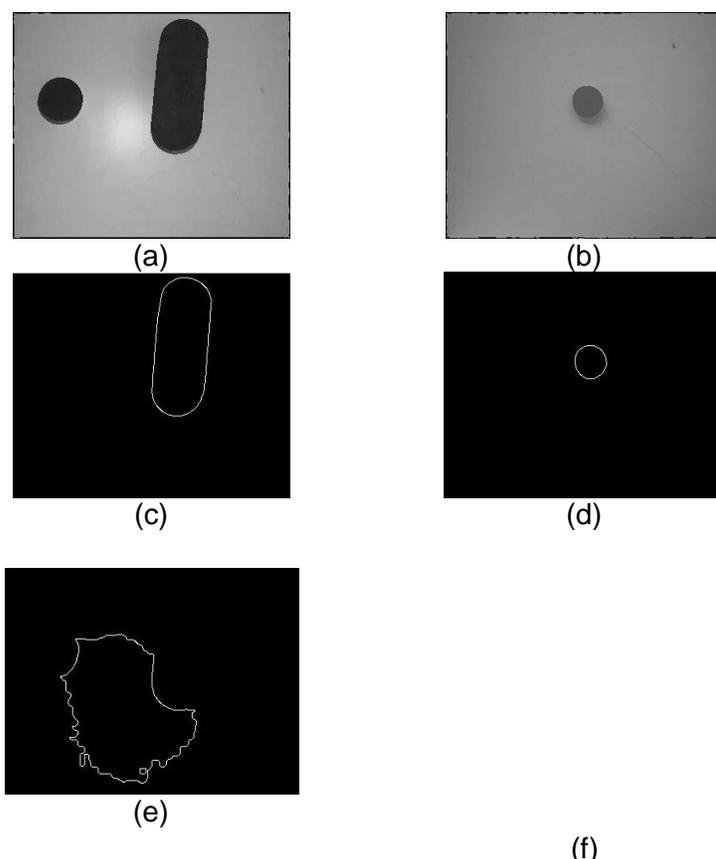
Figura 33. Utilización Filtro de mediana

En la Figura 33 (a) y (b) se aprecian las imágenes originales provenientes de la cámara web, en (c) y (d) el resultado de aplicar el filtro de mediana, (e) es el histo-

grama de la imagen (a), y (f) es el histograma de la imagen (b); (g) y (h) son los histogramas de las imágenes (c) y (d) respectivamente.

En las imágenes se buscó contrastar objetos fácilmente diferenciables (a) con otros de mayor dificultad (b). El resultado obtenido muestra que al comparar (e) y (f) respecto a (g) y (h), la variación entre los histogramas (e) y (g) no es muy grande pero entre los histogramas (f) y (h) es significativa, lo cual refuerza el uso del filtro de mediana.

En la Figura 34 se pueden observar los resultados del uso del filtro Nagao-Matsuyama para la eliminación del ruido foto-electrónico. Las imágenes (a) y (b) son el resultado de aplicar el filtro a las imágenes (a) y (b) de la Figura 33. En (c) y (d) se aprecia el cálculo de bordes para las imágenes originales y en (e) y (f) se muestra el cálculo de bordes de las imágenes (a) y (b) filtradas previamente con Nagao-Matsuyama; en (f) la imagen resultante no tiene ningún elemento; como se observa los resultados muestran una grave alteración de los datos originales, lo cual hace al filtro Nagao-Matsuyama totalmente inconveniente.

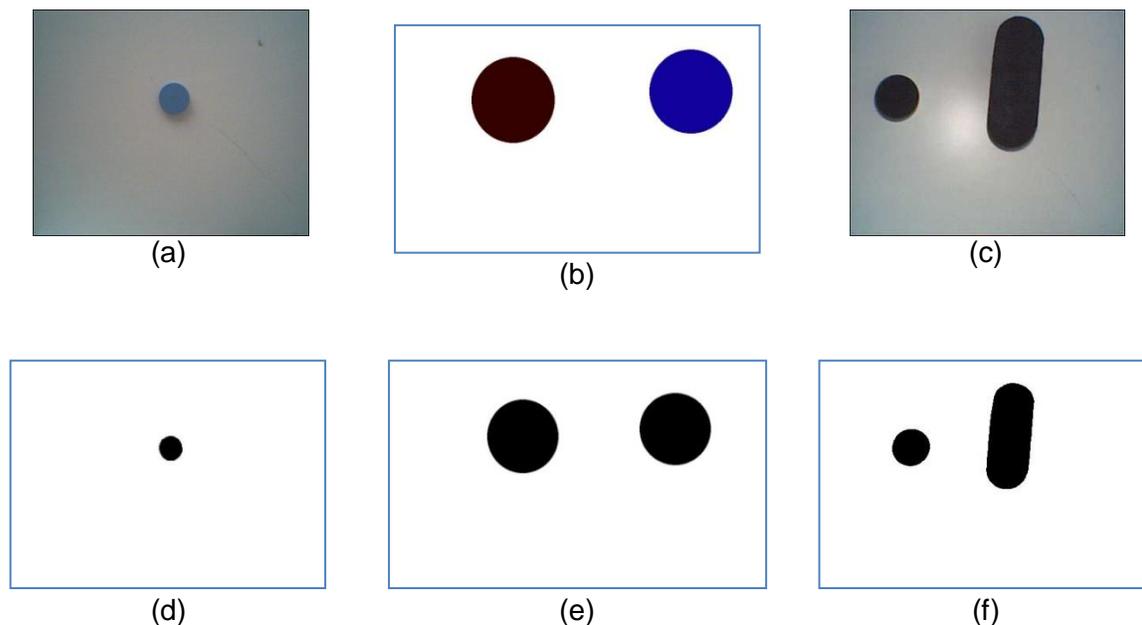


**Figura 34.** Aplicación del algoritmo de NAGAO-MATSUYAMA

Se hace una normalización a fin de utilizar todo el rango de cuantización para los píxeles de la imagen (0-255); este procedimiento tiene como ventaja adicional disminuir el efecto de la iluminación no uniforme.

En el inicio del proyecto, para el proceso de *segmentación* se usaba el algoritmo de Otsu (se encuentra implementado por defecto en MATLAB), pero en los resultados se obtuvieron efectos no deseados, como la aparición de bordes falsos debido a las sombras, razón por la cual se opta por el diseño de un algoritmo. Este método alternativo de segmentación (más específicamente el cálculo de umbral óptimo) se basa en el momento estadístico conocido como curtosis.

La Figura 35 (a), (b) y (c) muestra tres imágenes distintas las cuales se sometieron a un proceso de segmentación; el resultado se muestra en (d), (e) y (f).



**Figura 35.** Comprobación algoritmo segmentación

Lo anterior constituye una pequeña muestra de las pruebas realizadas, pues en total se utilizaron 116 imágenes, la mayoría sintéticas para comprobaciones del algoritmo. Una imagen sintética es construida usando un software de edición de imágenes como Paint.

Otros experimentos realizados se enfocaron a comparar los resultados arrojados por el algoritmo de Otsu y los del cálculo del umbral óptimo, usando para ello 116 imágenes diferentes. De acuerdo con el análisis de los datos obtenidos, el algoritmo diseñado para este trabajo presenta mejoras importantes respecto al procedimiento de Otsu en cuanto a diferenciación de imágenes de alta luminancia.

En la Tabla 6 se muestran los resultados; los aspectos considerados fueron matiz, saturación y luminancia que constituyen las componentes de color en el plano de color HSI; se utilizó la misma figura (forma constante).

**Tabla 6.** Comparación de algoritmos de Curtosis Vs Otsu para imágenes de alta luminancia

Imagen 015			
Colores		Curtosis	Otsu
Matiz	43	Ok	Error
Saturación	230		
Luminancia	96		
Forma	Constante		

Imagen 016			
Colores		Curtosis	Otsu
Matiz	43	Ok	Error
Saturación	215		
Luminancia	96		
Forma	Constante		

Imagen 017			
Colores		Curtosis	Otsu
Matiz	43	Ok	Error
Saturación	200		
Luminancia	96		
Forma	Constante		

Imagen 018			
Colores		Curtosis	Otsu
Matiz	43	Ok	Error
Saturación	175		
Luminancia	96		
Forma	Constante		

Imagen 019			
Colores		Curtosis	Otsu
Matiz	43	Ok	Ok
Saturación	150		
Luminancia	96		
Forma	Constante		

Otro punto a destacar respecto a los resultados conseguidos con el algoritmo implementado, es la clara diferenciación de objetos en fondos coloridos como se observa en la Tabla 7.

**Tabla 7.** Comparación de algoritmos de Curtosis Vs Otsu frente a identificación de objetos en fondos coloridos

Im	Objeto			Fondo			Curtosis	Otsu
	Matiz	Saturación	Luminancia	Matiz	Saturación	Luminancia		
074	80	240	125	80	240	233	Ok	Error
075	80	240	100	80	240	233	Ok	Error
080	30	240	2	0	240	2	Ok	Error
081	50	240	2	0	240	2	Ok	Error
090	0	120	2	0	240	2	Ok	Error
091	0	240	25	0	240	2	Ok	Error

**Nota.** En el Anexo A se encuentran las tablas completas de comprobación de datos.

De otro lado, es importante mencionar que el procedimiento implementado (algoritmo de segmentación) tiene algunos limitantes cuando los objetos presentes en la escena tienen algún tipo de estampado como es el caso mostrado en la Figura 36; en (a) y (b) se observan los elementos de interés, los resultados de la segmentación se ven en (c) y (d). Para el caso de (b), es evidente el error que se genera.

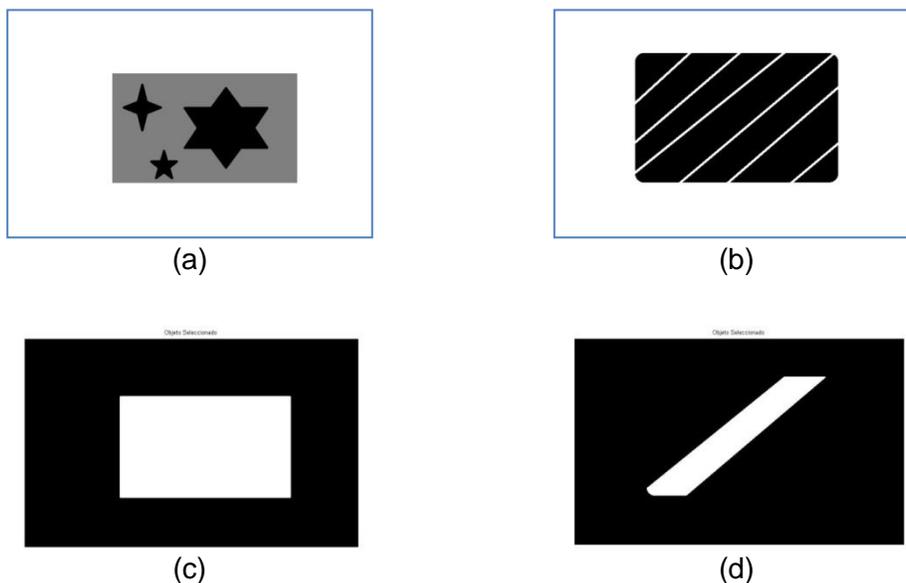


Figura 36. Resultados de segmentación con objetos estampados

El paso siguiente consiste en seleccionar un único elemento de la escena para hacer la aprehensión, para ello se calcula la distancia desde el centro de la imagen al centro de masa de los objetos. En la Figura 37 (a) y (b) se aprecian dos escenas claramente segmentables que contienen varios objetos, en (c) y (d) se identifican los objetivos de interés.

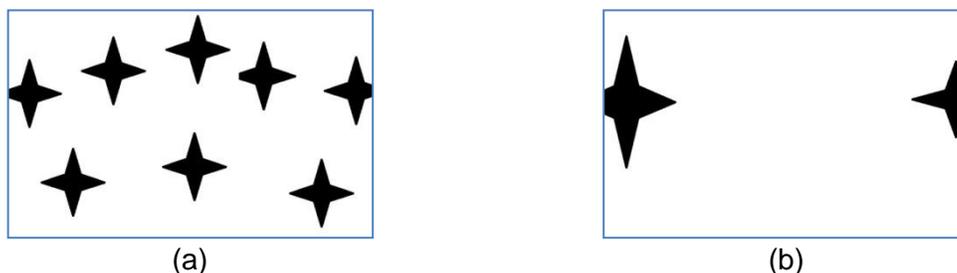




Figura 37. Resultados del algoritmo de selección de objetivo

### 6.2.2 Cálculo de puntos de aprehensión estable

El punto de arranque en este procedimiento es el cálculo del borde del objeto, para lo cual se usó el algoritmo basado en morfología ya que se requiere que el borde sea completamente cerrado, con un ancho de 1 pixel y con la menor carga computacional posible (condición que no se garantiza al utilizar las aproximaciones de gradiente, ni Canny); para el caso específico que se presenta cuando el elemento se encuentra en el borde de la imagen se implementó una funcionalidad que cierra el objeto automáticamente (se unen los dos puntos extremos del borde). En la Figura 38 (a) y (b) se observan los resultados de la detección de borde para las imágenes presentadas en la Figura 37 (c) y (d).



Figura 38. Resultados Algoritmo detección de bordes

En algunos casos especiales, como figuras con alto grado de irregularidad, fue necesario utilizar la operación morfológica cerradura pues no se generaban bordes totalmente cerrados.

En la Tabla 8 se presentan los resultados del cálculo de puntos de aprehensión para el caso de la Figura 37 (b), los fundamentos teóricos que sustentan este proceso fueron tratados en el Capítulo 5. Los datos mostrados en la tabla se explican a continuación:

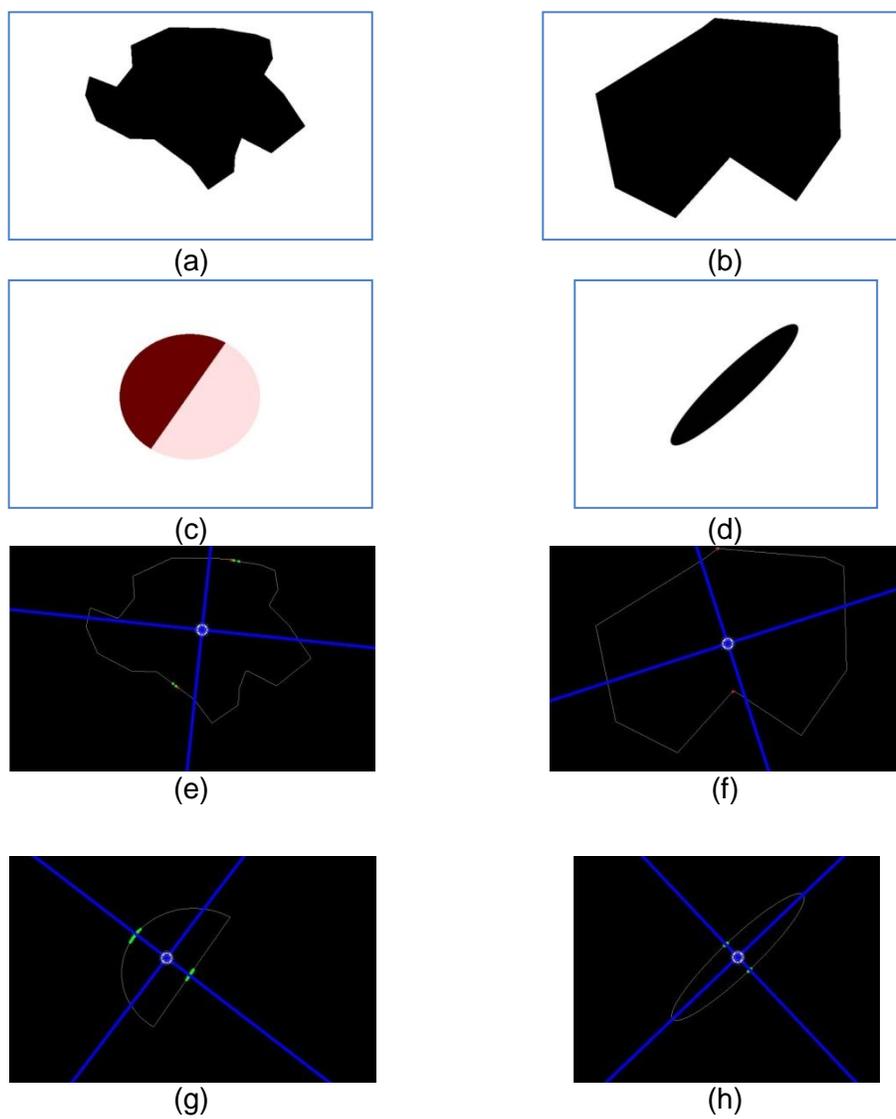
- La primera columna es el número de pixel de la curvatura (no utilizada para los cálculos).

- Las columnas II y III son la abscisa y la ordenada de  $\vec{P}$ .
- La columna IV es la curvatura  $\alpha$  del punto P.
- Las columnas V y VI son la abscisa y la ordenada de  $\vec{A}$ .
- Las columnas VII y VIII son la abscisa y la ordenada de  $\vec{P}'$ .
- La columna IX es la curvatura  $\alpha'$  del punto P'.
- Las columnas X y XI son la abscisa y la ordenada de  $\vec{A}'$ .
- La columna XII indica si el punto es de aprehensión: 1 lo es, 0 no lo es.

En la Figura 39 se observan los resultados del procesamiento de algunas imágenes. En (a), (b), (c) y (d) se ven las imágenes originales de partida; en (e), (f), (g) y (h) se aprecian en color verde los puntos de aprehensión calculados para cada figura y además los ejes de simetría en color azul.

Tabla 8. Resumen de resultados del cálculo de puntos de aprehensión para una imagen

# Pixel	Posición Y	Posición X	Curvatura	Normal al punto Co-ord. Y	Normal al punto Co-ord. X	Punto Anti-paralelo Y	Punto Anti-paralelo X	Punto Anti-paralelo Curvatura	Normal al pto Antiparalelo Y	Normal al pto Antiparalelo X	Es Punto de Aprehensión?
1	162,0	2,0	123,7	162,0	3,0	215,0	96,0	195,3	216,0	96,0	0,0
2	161,0	3,0	135,0	161,4	3,9	216,0	95,0	180,0	215,1	94,7	0,0
3	161,0	4,0	171,9	161,0	5,0	216,0	94,0	195,3	217,0	94,0	0,0
4	160,0	5,0	164,7	161,0	5,0	217,0	93,0	164,7	216,0	93,0	0,0
5	160,0	6,0	180,0	160,9	6,3	217,0	92,0	180,0	216,1	91,7	0,0
6	160,0	7,0	195,3	159,0	7,0	217,0	92,0	180,0	216,1	91,7	0,0
7	159,0	8,0	180,0	159,9	8,3	218,0	91,0	164,7	217,0	91,0	0,0
8	159,0	9,0	195,3	158,0	9,0	218,0	90,0	180,0	217,1	89,7	0,0
9	158,0	10,0	164,7	159,0	10,0	218,0	89,0	195,3	219,0	89,0	0,0
10	158,0	11,0	180,0	158,9	11,3	219,0	88,0	180,0	218,1	87,7	0,0
11	157,0	12,0	164,7	158,0	12,0	219,0	87,0	180,0	218,1	86,7	0,0
12	157,0	13,0	180,0	157,9	13,3	219,0	87,0	180,0	218,1	86,7	0,0
13	157,0	14,0	195,3	156,0	14,0	220,0	86,0	164,7	219,0	86,0	0,0
14	156,0	15,0	206,6	155,3	14,3	220,0	84,0	195,3	221,0	84,0	0,0

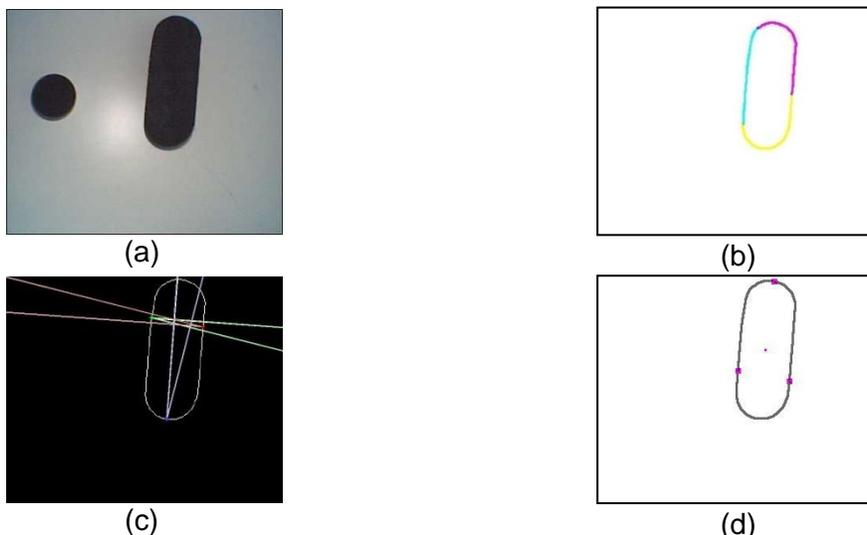


**Figura 39.** Resultado del cálculo de los puntos de aprehensión

En el anexo B a este documento se pueden observar con mayor resolución algunos otros resultados para aprehensión con dos y tres dedos.

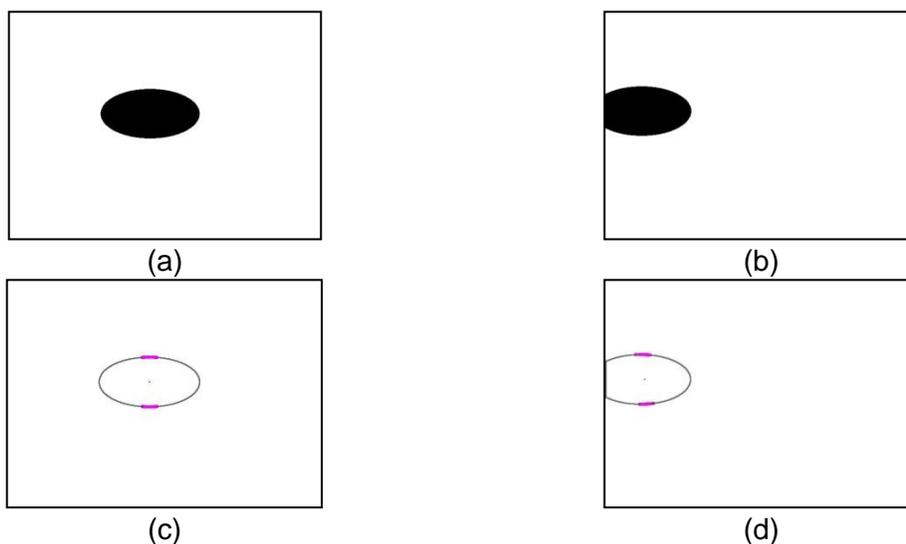
Usando la Figura 40 se analizan los resultados de una detección de puntos de aprehensión para agarre con tres dedos; en (a) se aprecia la imagen original, en (b) se muestra como el algoritmo divide la curvatura en 3 zonas: magenta, amarilla y cyan. Posteriormente se toman los primeros pixeles de cada una de las zonas y se trazan los conos de fricción a fin de calcular el área común entre ellos (figura (c)), este proceso se repite con cada uno de los pixeles de las tres zonas. Por último se selecciona una tripleta de puntos (la que tenga mayor área común) como se muestra en la figura (d).

Todos los resultados expuestos se basan en los cálculos matemáticos presentados en las secciones 2.2.2, 2.2.3, 5.2.2 y 5.2.3, con lo cual se puede comprobar la estabilidad de los puntos de aprehensión.



**Figura 40:** Resultados algoritmo de detección de puntos de aprehensión estable para un agarre con 3 dedos. (a) Imagen Original. (b) División de curvatura en 3 zonas. (c) Diagramado de conos de fricción y detección de área común. (d) Resultado del algoritmo

La ubicación de la cámara en la prótesis es un aspecto importante a tener en cuenta: se debe buscar que los objetivos estén lo más cerca posible del centro de la imagen para evitar la mala ubicación de los puntos de aprehensión; adicionalmente este posicionamiento minimiza el efecto de las malformaciones del lente de la cámara. En la Figura 41 se observa como una desviación del objetivo (figuras (a) y (b)) del centro de la escena, provoca un traslado de los puntos de aprehensión, figuras (c) y (d) o puede generar un desplazamiento completo de los puntos de aprehensión figuras (e) y (f).



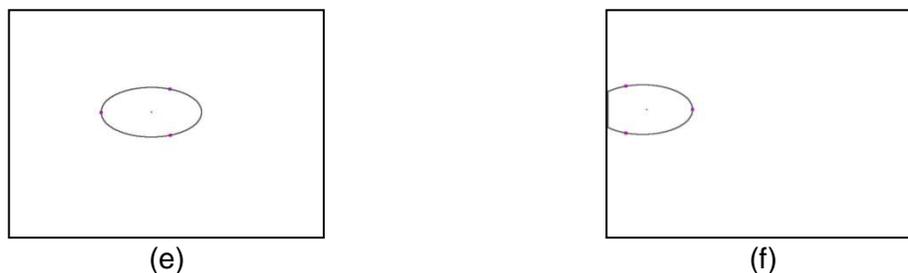


Figura 41: Imágenes para sustentar la ubicación de la cámara

### 6.3 Implementación del Algoritmo.

El algoritmo fue implementado y probado usando MATLAB con ayuda del toolbox de procesamiento de imágenes; está compuesto por 21 funciones y su estructura general se puede ver en la Figura 42.

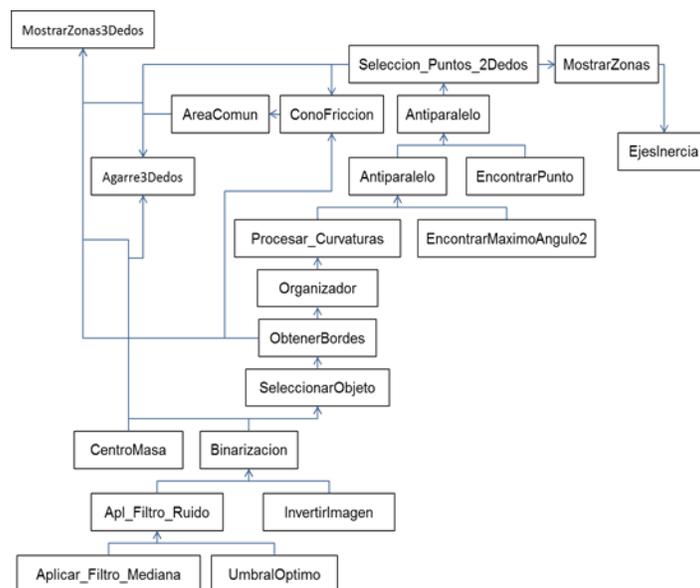


Figura 42. Estructura del algoritmo implementado

A continuación se hace una breve descripción de las diferentes funciones:

- Las funciones que se encargan de realizar la selección del objetivo, son: *Apl\_Filtro\_Ruido*, *Aplicar\_Filtro\_Mediana*, *UmbralOptimo*, *Binarización*, *InvertirImagen*, *SeleccionarObjeto*, *CentroMasa*.
- El cálculo de los puntos de aprehensión 2 dedos es realizado por las funciones: *Obtener\_bordes*, *Organizador* (construye una tabla con todos los pixeles de la curvatura en orden sucesivo), *Procesar\_Curvaturas* (calcula la curvatura del punto), *Antiparalelo*, *EncontrarMaximoAngulo2*, *PixelesAgarre*, *EncontrarPunto* y *MostrarZonas*.

- El cálculo de los puntos de aprehensión para tres dedos es realizado por `ConoFriccion`, `AreaComun`, `Agarre3Dedos` y `MostrarZonas3Dedos`.
- Para entender la Figura 42. Estructura del algoritmo implementado se debe tener en cuenta que las flechas que llegan a los recuadros son las entradas que se requieren para el procesamiento de esa función. Ejemplo: `Agarre3Dedos` requiere la salida de: `Seleccion_Puntos_2Dedos`, `AreaComun`, `CentroMasa` y `ObtenerBordes`

## 6.4 Protocolo de Pruebas del Algoritmo

Las pruebas a las que se sometió el algoritmo abarcan cinco aspectos básicos: estabilidad, inmunidad al desplazamiento, inmunidad a la rotación, efectos de la iluminación y tiempo de procesamiento del algoritmo.

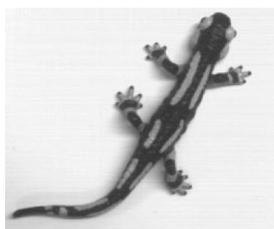
Cada una de las pruebas siguió un protocolo específico que será explicado en cada caso.

**NOTA:** *Las imágenes presentadas en este documento solo son algunos de los resultados. Dentro del CD adjunto al trabajo de grado se pueden observar más de doscientos resultados provenientes de al menos 140 imágenes*

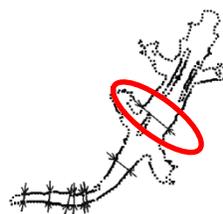
### 6.4.1 Estabilidad

El procedimiento para comprobar la funcionalidad del algoritmo consistió en tomar las imágenes de 3 figuras utilizadas por Recatalá en su trabajo [2] y comparar los resultados obtenidos en ambos casos.

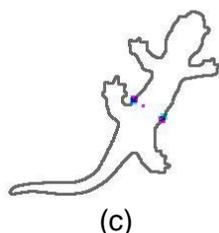
La primera imagen de prueba es la de un juguete con forma de salamandra, y se muestra en la Figura 43 (a). La Figura 43 (b) muestra los puntos de aprehensión del objeto según [2]. Se puede apreciar resaltado con la elipse roja el punto de aprehensión más estable si se considera el criterio de cercanía con el centro de masa. En la Figura 43 (c) se pueden apreciar los puntos de aprehensión calculados por el algoritmo desarrollado en este proyecto; como se puede apreciar, se encuentran en la misma posición que en la figura 43 (b). La diferencia entre los dos resultados es que, como se ve en la Figura 43 (c), el algoritmo de segmentación desarrollado no determina las manchas como agujeros dentro del objeto, como sí se hace en [2] (Figura 43 (a)).



(a)



(b)

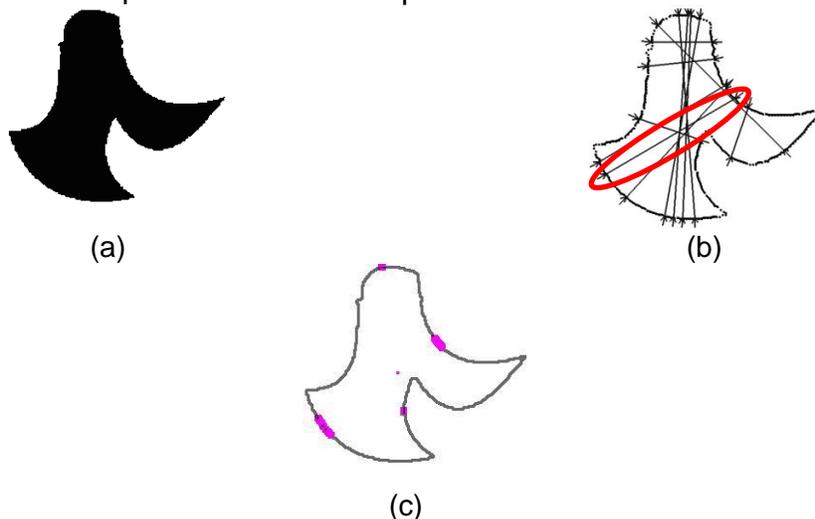


**Figura 43:** Resultado de Puntos de Aprehensión en la referencia [2]. (a) Imagen Original. (b) Imagen resultante (Fuente: [2]). (c) Resultado del procesamiento de (a) con este algoritmo.

En la **Figura 44** se muestra una segunda forma (también tomada de [2]) para seguir la comprobación del algoritmo diseñado. La **Figura 44** (a) es la base del procesamiento, en (b) está el resultado proveniente de [2] y finalmente en (c) se encuentra el resultado de este proyecto.

Dentro de la elipse roja en la **Figura 44** (b) se encuentra la zona de aprehensión, que es común con la encontrada en este proyecto. Gracias a la utilización de las condiciones dadas por Montana (Ver Apartado 2.2.2), el algoritmo desarrollado elimina las otras zonas de aprehensión.

La zona extra que se observa se debe a que el objeto no cumple completamente con las restricciones presentadas en el apartado 1.1.



**Figura 44:** Resultado de Puntos de Aprehensión en la referencia [2]. (a) Imagen Original. (b) Imagen resultante (Fuente [2]). (c) Resultado del procesamiento de (a) con este algoritmo.

Por último se toma la imagen de una tuerca. Los resultados para este tipo de objeto se pueden observar en la **Figura 45**. En la **Figura 45** (a) se presenta el resultado de la referencia [2] y en **Figura 45** (b) el resultado para este algoritmo. Como se observa, las zonas de aprehensión externas son iguales, pero debido a que una tuerca no cumple con todas las restricciones de 1.1, las zonas internas (agujero) no son detectadas.



Figura 45: Resultados del procesamiento para una tuerca. (a) Referencia [2]. (b) Algoritmo Diseñado

### 6.4.2 Desplazamiento

Para comprobar que el algoritmo es inmune a los desplazamientos del objeto, se realizó el procesamiento de dos imágenes del mismo objeto, en una de las cuales éste está desplazado.

Las imágenes que se utilizarán son las presentadas en la Figura 46 y corresponden a una moneda de \$200. Estas imágenes fueron tomadas con iluminación a contraluz, la que se tomó como referencia ya que no tiene sombras resultantes.

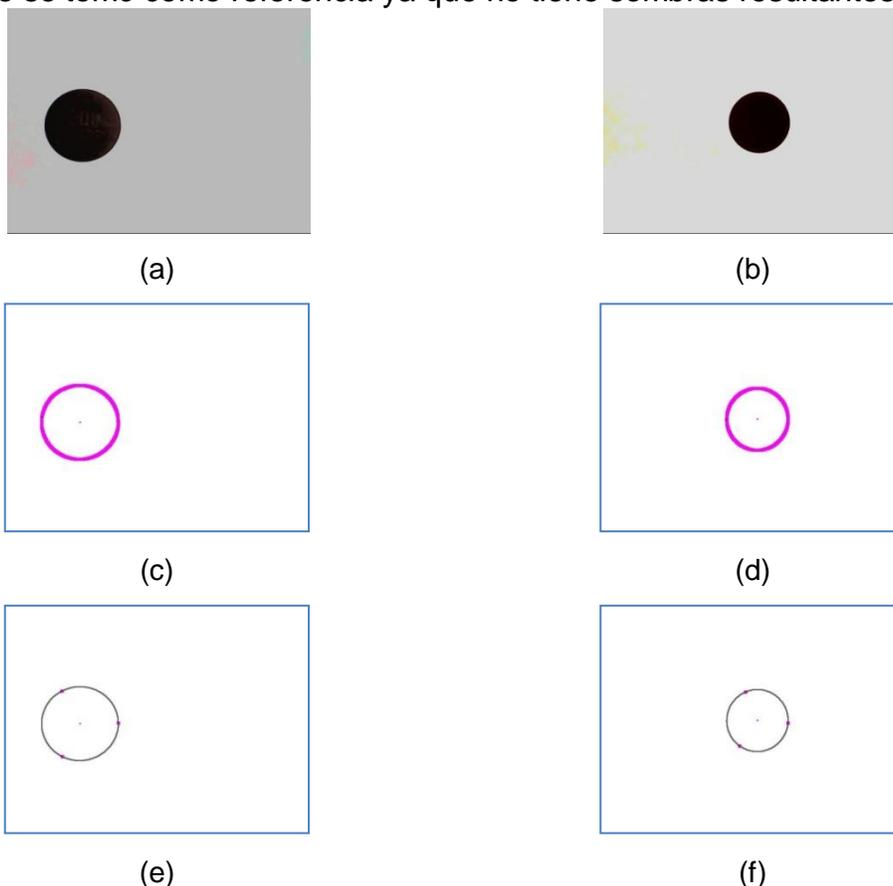


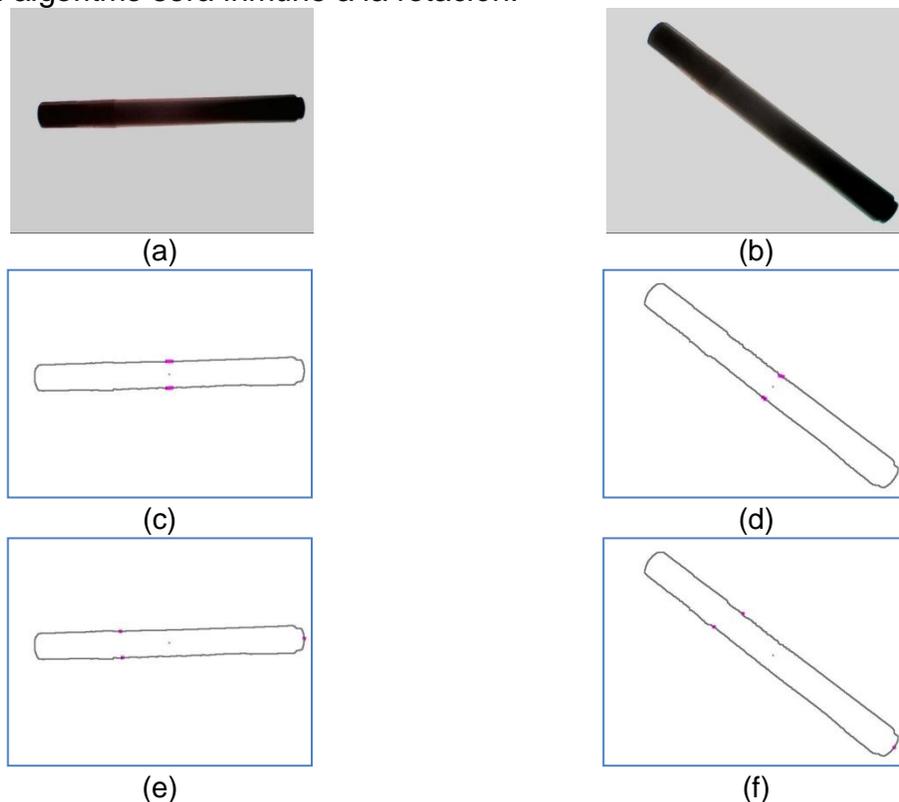
Figura 46: Imágenes para demostrar la inmunidad al desplazamiento del algoritmo.

En las figuras (c), (d) precedentes, se presentan las zonas de aprehensión para agarre con dos dedos de las figuras (a) y (b) respectivamente, y en las figuras (e)

y (f) los puntos para agarres con tres dedos. Como se puede apreciar, las respuestas son exactamente iguales sin importar si está desplazado o no por lo tanto se puede concluir que el algoritmo es inmune al desplazamiento del objeto.

### 6.4.3 Rotación

Se procedió de manera similar que en el punto anterior, es decir dos imágenes con un objeto similar. En una de las ellas el objeto estará rotado con respecto al de la otra imagen. Si al calcular los puntos de aprehensión éstos son iguales, entonces el algoritmo será inmune a la rotación.



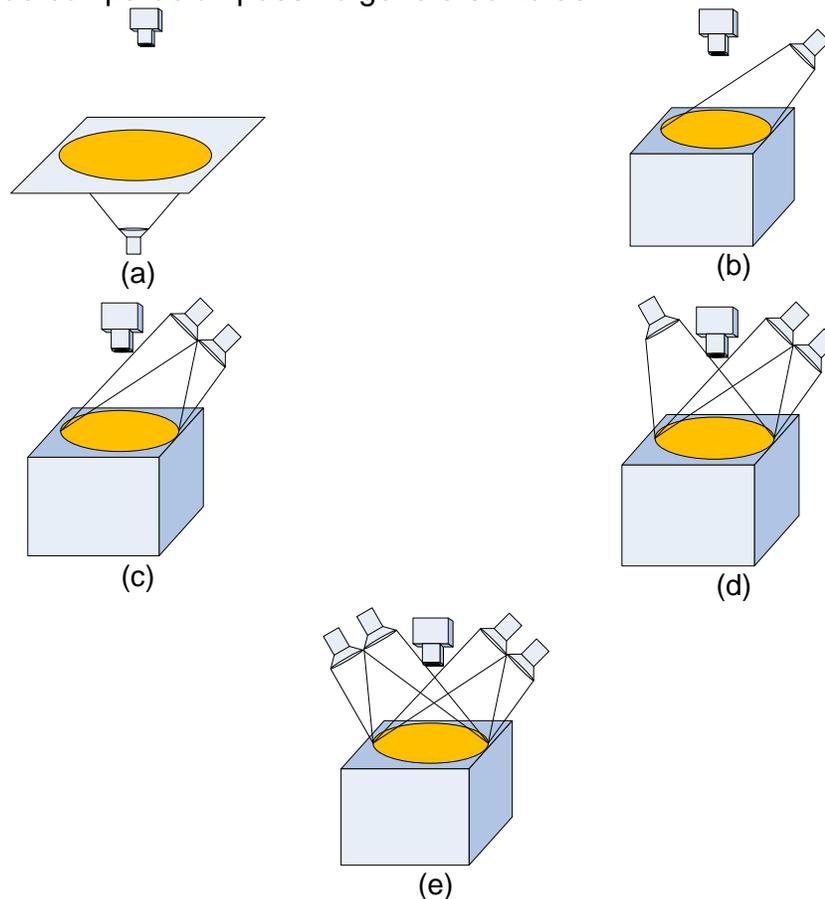
**Figura 47:** Imágenes para comprobación de la inmunidad a la rotación.

En base a la **Figura 47**, las figuras (c) y (d) son las respuestas para dos dedos y las figuras (e) y (f) para tres. Como se puede observar, los puntos de aprehensión son los mismos así se gire el objeto, y es por esta razón que el algoritmo se puede considerar como inmune a la rotación.

### 6.4.4 Iluminación

La iluminación es el factor más importante en la operación de segmentación del objetivo. Para esta prueba se utilizan las imágenes con diferentes focos de iluminación, como se puede apreciar en la **Figura 48**; en (a) se presenta iluminación a contraluz y las siguientes son iluminaciones direccionales: (b) 1 fuente, (c) 2 fuen-

tes, (d) 3 fuentes y (e) 4 fuentes de luz. La iluminación a contraluz se utilizará como patrón de comparación pues no genera sombras.



**Figura 48:** Montaje con diferentes fuentes de iluminación utilizadas para las pruebas

En la **Figura 49** se encuentra la representación de un marcador bajo diferentes iluminaciones. En la figura (a) se encuentra la imagen con iluminación a contraluz, es importante decir que la ausencia de sombras es la característica más importante. En la figura (b) se encuentra una imagen con una fuente de iluminación direccional, se observa gran presencia de sombras y una iluminación totalmente discontinua, es el peor de los escenarios para el algoritmo. En la figura (c) hay una imagen con dos fuentes de iluminación direccional, existe la presencia de sombras aunque la iluminación de la imagen es un poco más continua que en el caso anterior. En la figura (d) se encuentra la imagen con tres fuentes de iluminación direccional, existe una disminución mayor de las sombras; y, finalmente en (e) están presentes las cuatro fuentes de iluminación y las sombras casi son eliminadas.

En las pruebas anteriores ya se ha comprobado el algoritmo de cálculo de puntos de aprehensión, por ello, las 5 figuras serán introducidas únicamente al algoritmo de segmentación de objetos para su comprobación.

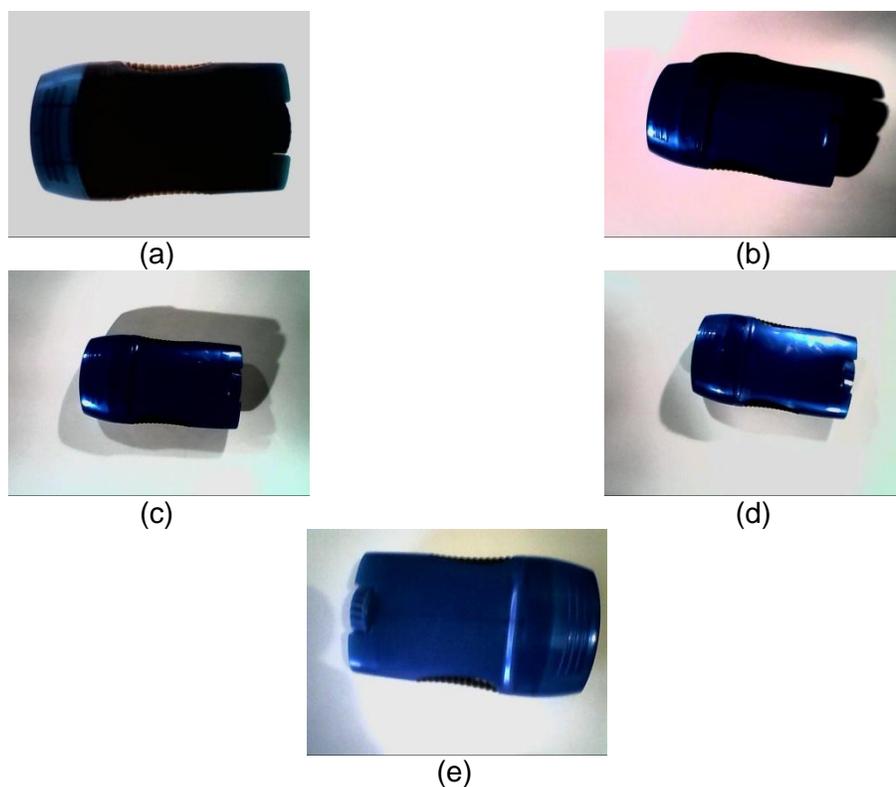
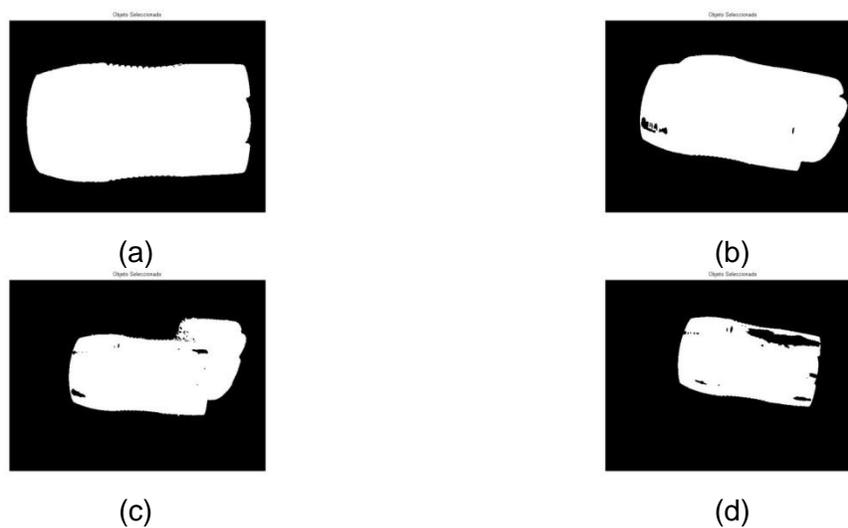


Figura 49: Imágenes con diferentes iluminaciones

Como se puede observar en la **Figura 50**, en (a) está la imagen de comparación. Esta es la imagen a contraluz que no ha generado ningún tipo de sombra y es una iluminación constante, razones por las cuales el algoritmo tiene un buen comportamiento. La imagen (b) es la que se realizó con una sola fuente de luz direccional, ésta genera una gran sombra que deforma totalmente al objeto, y dará como resultado la variación de los puntos de aprehensión.





(e)

Figura 50: Resultados del algoritmo para la Figura 49

La imagen (c) de la **Figura 50** ya tiene 2 fuentes de iluminación direccional, es por ello que el perfil tiene una deformación menor que la figura (b), que para la aplicación es un error todavía muy grande. La imagen (d) está iluminada con 3 fuentes de luz direccionales, la deformación del perfil es nula, pero la presencia de los agujeros ocasionados por la iluminación no uniforme puede afectar la posición del centro de masa, haciendo que se afecte la estabilidad del sistema. Para finalizar se tiene la imagen (d) que es tomada con 4 fuentes de iluminación direccionales lo que ocasiona un buen perfil y una iluminación del objeto aún más uniforme.

### 6.4.5 Color

Se colocaron varias imágenes con diferentes colores (todas ellas sintéticas) las cuales se pueden observar en el Anexo B a este documento.

En la **Figura 51**: Imágenes para comprobación de color se encuentran imágenes extraídas del anexo B para la comprobación de la inmunidad al cambio de color. En la **Figura 52** aparecen los resultados de la aplicación del algoritmo de determinación de puntos de aprehensión; es importante anotar que en los objetos seleccionados existe más de un par de zonas que ofrecen la misma estabilidad de agarre.

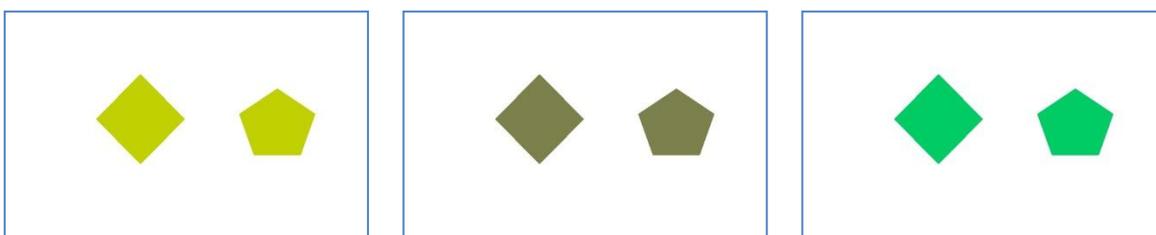


Figura 51: Imágenes para comprobación de color

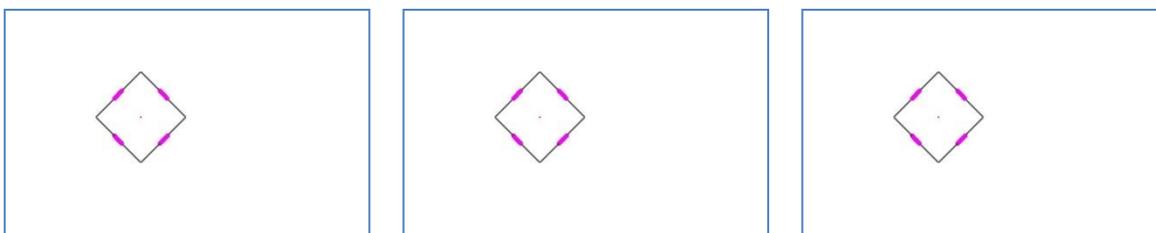
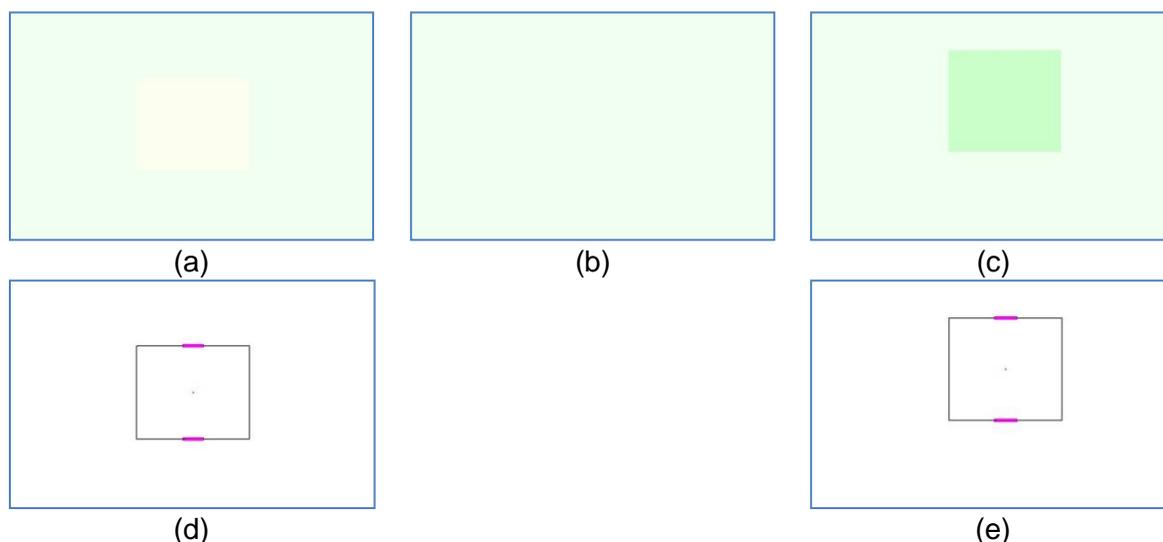


Figura 52: Resultado para imágenes de la Figura 51: Imágenes para comprobación de color

Es importante decir que la inmunidad al efecto del color llega hasta cierto punto como se demostrará en la **Figura 53**. Para esto se toman imagen con un fondo de color y objeto de un color diferente. En (a) el objeto tiene un matiz diferente que el fondo, en (b) el objeto tiene una saturación diferente a la del fondo y en (c) el objeto tiene una intensidad diferente que la del fondo. De las figuras se puede concluir que si existen cambios en el matiz o en la intensidad entonces el algoritmo de segmentación podrá cumplir con su función, como se observa en las figuras (d) y (e).



**Figura 53:** Imágenes para comprobar el límite de la inmunidad al color

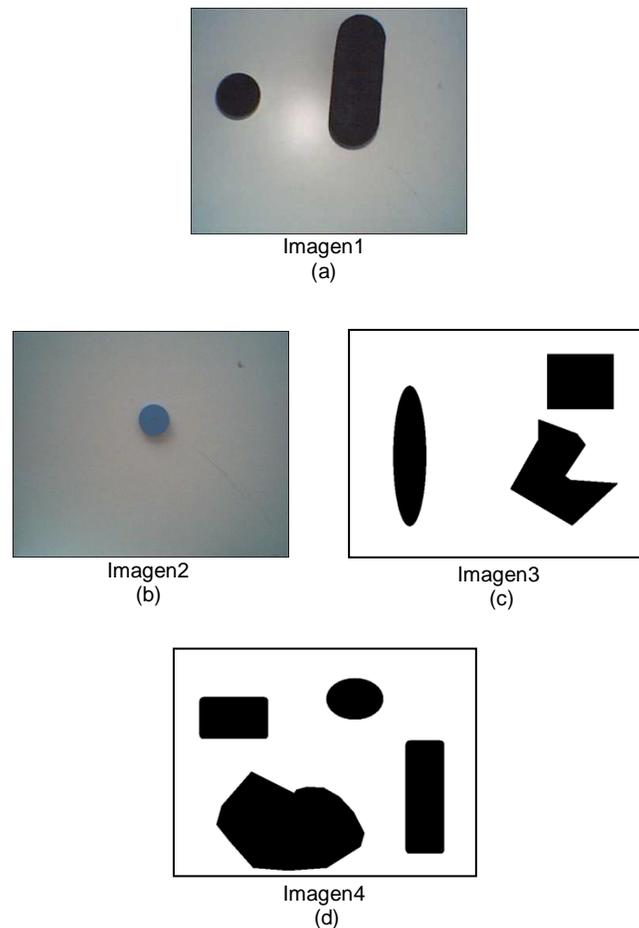
#### 6.4.6 Tiempo de Procesamiento

Para estudiar los resultados del tiempo de procesamiento se emplearon las imágenes mostradas en la Figura 54.

En la **Tabla 9** se muestra una caracterización de dichas imágenes; los datos fueron obtenidos usando la herramienta imageinfo del toolbox de procesamiento de imágenes de MATLAB.

Pos.	Nombre Documento	Forma-to	Ancho	Alto	Res. Bit	Tipo Color	Tamaño (bits)
(a)	Imagen 1	jpg	352	258	8	Verdadero	7153
(b)	Imagen 2	jpg	352	288	8	Verdadero	6821
(c)	Imagen 3	png	640	480	8	Verdadero	9105
(d)	Imagen 4	png	640	480	8	Verdadero	9930

**Tabla 9.** Características de las imágenes de prueba



**Figura 54.** Imágenes para probar el tiempo de procesamiento

En la Figura 55 se identifican los puntos de aprehensión estable (zonas en color rojo) para cada una de las imágenes de la Figura 54 :

En la Tabla 10 se muestran los resultados asociados al tiempo de procesamiento; en los algoritmos de visión artificial y de procesamiento de imágenes, la carga computacional medida como el tiempo que las funciones demoran en hacer lograr su objetivo, es muy susceptible al tamaño de la imagen, el número de objetos presentes en la escena y la resolución de pixel. Lo anterior se evidencia para el caso de la Imagen 4 que es la más grande y con la mayor cantidad de objetos, el tiempo de finalización del algoritmo es mayor que para las demás imágenes.

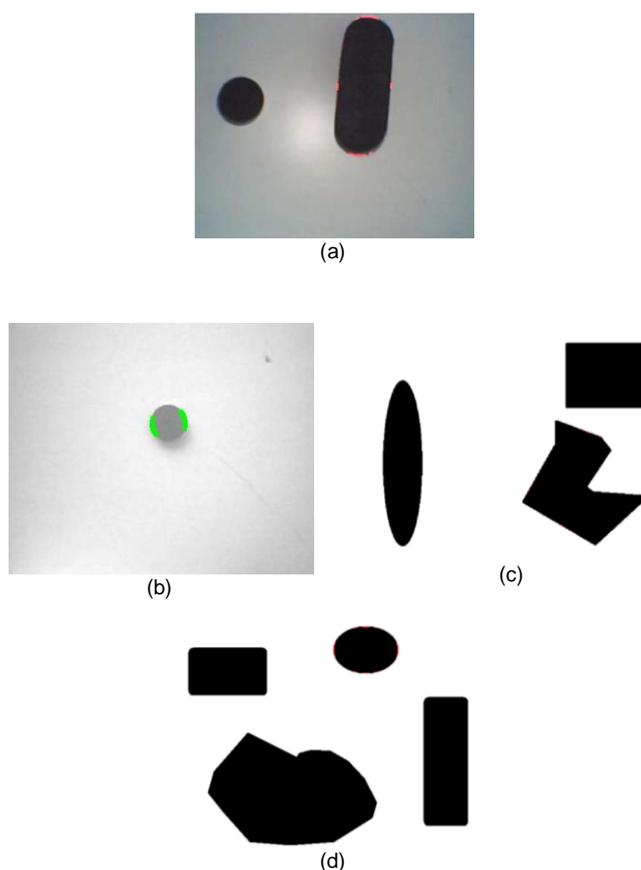


Figura 55. Resultados de las imágenes

Tabla 10. Resumen de tiempos de procesamiento

	Imagen 1	Imagen 2	Imagen 3	Imagen 4	Promedio
<u>Principal</u>	<b>0,949</b>	<b>0,859</b>	<b>0,804</b>	<b>1,304</b>	<b>0,9462</b>
Apl_Filtro_Ruido	<b>0,233</b>	<b>0,281</b>	<b>0,247</b>	<b>0,218</b>	<b>0,2492</b>
Aplicar_Filtro_Mediana	0,079	0,11	0,077	0,078	0,0906
UmbralOptimo	0,046	0,047	0,046	0,063	0,0498
Instrucciones Varias	0,108	0,124	0,124	0,077	0,1088
Binarizacion	<b>0,032</b>	<b>0,032</b>	<b>0,031</b>	<b>0,062</b>	<b>0,0346</b>
InvertirImagen	0	0	0,016	0,016	0,0064
Instrucciones Varias	0,032	0,032	0,015	0,046	0,0282
SeleccionarObjeto	<b>0,201</b>	<b>0,202</b>	<b>0,202</b>	<b>0,404</b>	<b>0,2426</b>
CentroMasa	0	0	0	0,031	0,0062
Instrucciones Varias	0,201	0,202	0,202	0,373	0,2364
Obtener_bordes	<b>0,202</b>	<b>0,187</b>	<b>0,201</b>	<b>0,31</b>	<b>0,2174</b>
Instrucciones Varias	0,202	0,187	0,201	0,31	0,2174
Organizador	<b>0,032</b>	<b>0,016</b>	<b>0,016</b>	<b>0,016</b>	<b>0,016</b>

Instrucciones Varias	0,032	0,016	0,016	0,016	0,016
Procesar_Curvaturas	<b>0,031</b>	<b>0,016</b>	<b>0,015</b>	<b>0,031</b>	<b>0,0248</b>
Instrucciones Varias	0,031	0,016	0,015	0,031	0,0248
Antiparalelo	<b>0,109</b>	<b>0,015</b>	<b>0,016</b>	<b>0,078</b>	<b>0,0468</b>
EncontrarMaximoAngulo2	0,109	0,015	0,016	0,078	0,0468
Instrucciones Varias	0	0	0	0	0
PixelesAgarre	<b>0,047</b>	<b>0,015</b>	<b>0,016</b>	<b>0,047</b>	<b>0,028</b>
EncontrarPunto	0	0	0	0,016	0,0032
Instrucciones Varias	0,047	0,015	0,016	0,031	0,0248
MostrarZonas	<b>0,016</b>	<b>0</b>	<b>0,015</b>	<b>0,031</b>	<b>0,0156</b>
Instrucciones Varias	0,016	0	0,015	0,031	0,0156
Otras instrucciones	<b>0,046</b>	<b>0,095</b>	<b>0,045</b>	<b>0,107</b>	<b>0,0712</b>



# Capítulo 7. Conclusiones y recomendaciones

## 7.1 Conclusiones

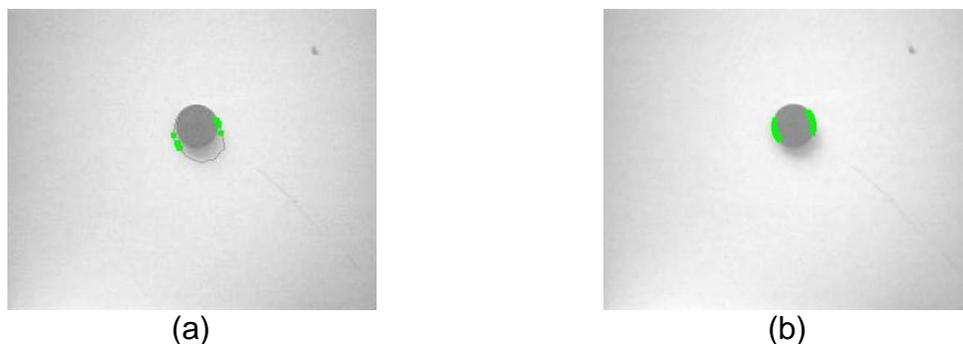
Una vez finalizado el trabajo se puede concluir que:

Se lograron identificar los puntos de aprehensión estable para un elemento de interés en una imagen por medio del desarrollo de un algoritmo, que tiene las siguientes características:

1. Realiza la captura de la escena usando una cámara web.
2. Selecciona el objeto de interés entre los objetos presentes en la imagen obtenida.
3. Determina las características físicas del objeto seleccionado, necesarias para el cálculo de los puntos de agarre: centro de masa y curvatura
4. Calcula los puntos de aprehensión estable para el objeto de interés, para dos y tres dedos.

Existen algunos factores que deben ser considerados para el cálculo de los puntos de aprehensión: el ruido presente en la imagen (ver Figura 56), la cantidad de objetos presentes en la imagen y el tamaño del objeto de interés.

En promedio el 27% del tiempo de ejecución del algoritmo está dedicado a la eliminación del ruido; el 26% se emplea en seleccionar el objeto; siendo estadísticamente estas dos funciones las que más cargan al equipo. Un dato importante es que aproximadamente el 78% del tiempo de ejecución está dado por instrucciones internas de MATLAB; esto significa que el algoritmo sería mucho más rápido si fuera implementado en un sistema embebido. Sería interesante analizar la respuesta en un sistema de tiempo real como LINUX.



**Figura 56:** Necesidad de sistema de eliminación de ruido. (a) Sin sistema de eliminación. (b) Con sistema de eliminación.

La cantidad de objetos en la escena influye directamente en la carga computacional: a mayor número de objetos, se incrementa el número de veces que se debe ejecutar la rutina para calcular el centro de masa (Ver Resultados en la Tabla 10); lo anterior a fin de identificar el objeto de interés, al cual se le encontrará los puntos de aprehensión.

El tamaño del objeto de interés también afecta la carga computacional: a mayor tamaño, el borde del objeto tendrá una mayor cantidad de píxeles; esto implica que el algoritmo de descripción de curvatura tendrá que ser ejecutado más veces; además existirán una mayor cantidad de posibles puntos de aprehensión..

El paquete software MATLAB es una herramienta poderosa en el diseño y construcción de algoritmos de procesamiento de imágenes. Tiene una desventaja y es que se requiere un cambio en la conceptualización del código, pues la típica codificación orientada a objetos da como resultado códigos pesados y lentos.

En cuanto a algunas de las dificultades encontradas durante el desarrollo del trabajo, se sugieren las siguientes dos recomendaciones:

El desarrollo de este tipo de trabajos debe ser soportado por un grupo interdisciplinario que incluya físicos, matemáticos, ingenieros biomédicos, etc, que aporten conocimientos y experiencia desde su campo específico de trabajo, garantizando resultados más concluyentes.

Se recomienda que para la integración del algoritmo desarrollado en este proyecto, con la prótesis de mano robótica, se haga la implementación de este algoritmo sobre un circuito hardware en base a FPGA, a fin de mejorar su desempeño.

Durante el desarrollo del proyecto fueron probadas con éxito las teorías planteadas por los diferentes autores consultados (ver Capítulo 8), referidos al cálculo de los puntos de aprehensión estable.

Considero que se cumplieron los objetivos propuestos para este proyecto de investigación, sin embargo, la solución presenta algunas limitaciones funcionales tales como: imposibilidad de incluir una fuente de iluminación en la prótesis, fondo

no uniforme, traducción de información de posición de los puntos en el plano de la imagen a coordenadas reales, dificultad de seleccionar objetos ante la falta de información de profundidad. Estos inconvenientes deben ser solucionados en proyectos posteriores como se menciona en el apartado de trabajos futuros (sección 7.3)

Para el desarrollo de este tipo de proyectos, es indispensable contar con la prótesis de mano robótica funcional que permita hacer las pruebas necesarias, y así garantizar una delimitación real del problema al iniciar el trabajo.

## **7.2 Aportes de este trabajo**

Construcción de un sistema que permite el cálculo de los puntos de aprehensión estable de un objeto en una escena, basado en el fundamento teórico del procesamiento de imágenes y la robótica.

A partir de la literatura consultada sobre la aprehensión robótica mediante el procesamiento de imágenes, donde se reportaban los pasos a seguir para el cálculo de los agarres, fueron implementados las rutinas y algoritmos (scripts de MATLAB) que permiten la ejecución efectiva de dichos procedimientos.

La revisión literaria evidenció que a nivel regional y nacional, el proyecto es un buen primer avance para hacer una integración de las técnicas de agarre y de procesamiento de video.

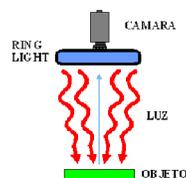
Se diseñó un algoritmo fundamentado en el momento estadístico de la curtosis, que presenta algunas ventajas sobre el tradicional algoritmo de Otsu (implementado por MATLAB), las cuales fueron discutidas en la sección 4.2 y en los Anexos.

## **7.3 Trabajos Futuros**

A partir de las limitaciones de este proyecto, de los resultados generados y de las propuestas hechas en otros trabajos consultados se plantean nuevos proyectos de investigación en este campo del conocimiento

### **7.3.1 Imágenes**

Para solventar el problema de la iluminación es importante la ubicación de una fuente de luz dentro de la prótesis. Ya que como se demostró en las pruebas la ubicación de una sola fuente de luz es un grave inconveniente, la fuente de luz debería estar justamente alrededor de la cámara, como se muestra en la Figura 57



**Figura 57:** Iluminación Radial

Fuente: Tomada de España E. Flórez J. "Curso de Visión de Máquina". Universidad del Cauca. Colombia 2009

Esto generaría una iluminación total del objeto, generando bordes perfectamente definidos y sin sombras. En estas circunstancias, los puntos de aprehensión resultarían ubicados exactamente en las mismas posiciones que a contraluz, pues la deformación del perfil resultaría nula.

En cuanto a la forma de diferenciar los objetos, será necesaria la utilización de técnicas de inteligencia artificial, ya sea redes neuronales o lógica difusa así como también minería de datos, de modo que se pueda hacer una segmentación aún en la presencia de las peores condiciones.

### 7.3.2 Objetos

Si los objetos no tienen masa uniforme se puede deber a su forma o al material del que está hecho. En el primer caso se tendrá como ejemplo una taza de café, en el cual debido a su forma, la masa no está uniformemente distribuida. El segundo caso es el que se tiene cuando existe más de un material formando el objeto, por ejemplo un martillo, la cabeza de éste (metal), está conformada por un material mucho más denso que el material del cuerpo (madera). Teniendo en cuenta estos dos tipos de problemas en la uniformidad de la masa, se requerirá una o las dos siguientes condiciones:

- Si la masa del objeto es no uniforme debido a la forma, entonces se requerirá un modelo completo de la física del objeto. Esto implica hacer una reconstrucción en 3D del objeto completamente. Una referencia para este tipo de problemas se da en el trabajo de Miller y Allen titulado Examples of 3D grasp quality computations.
- Si el objeto no es uniforme debido a su material, la información de una imagen podría no ser suficiente, requiriendo de información externa. La información puede provenir de los sensores de fuerza de la mano robotizada.
- Otra solución para la falta de información que crearía la heterogeneidad de la masa de los objetos es permitir que el algoritmo aprenda de sus experiencias anteriores. Esto se ha reportado en trabajos como Recatalá "Determinación, seguimiento y ejecución visual de agarres 2D usando una aproximación inspirada en comportamientos".

### 7.3.3 Descripción espacial real de Puntos en una Imagen

Con el objetivo de ampliar este trabajo, se propone como problema de investigación la generación de la información espacial real de los puntos de aprehensión aquí calculados.

Como propuesta para este trabajo se plantea la siguiente línea de razonamiento:

1. Calibración de la cámara: en esta operación se hace la descripción completa de la cámara(s) de video haciendo que los ejes X e Y de la escena coincidan completamente con los ejes reales respectivos.
2. Obtener la profundidad de la escena: Dado que la imagen es una función de dos dimensiones, el proceso de calibración no determina el eje z (profundidad) de la escena. Para este proceso se pueden utilizar, según Pajares y de la Cruz en su libro "Visión Por Computador" [20]:
  - a. Un análisis del movimiento en la escena. En el cual se utiliza una sola cámara. Esto se hace por medio de: el análisis del flujo óptico (cambios en la imagen debido al movimiento durante un intervalo de tiempo) y el campo de movimiento (representación bidimensional de un movimiento tridimensional; a cada pixel en la escena se le asigna un vector que representará la velocidad en el sentido del movimiento) o el análisis de la relación de adyacencia (relaciones entre distancia/velocidad de imagen y escena).
  - b. Por visión estereoscópica. En la cual se hace uso de una segunda cámara para tomar la escena y por medio de triangulación se obtiene la profundidad.
  - c. Utilizando la textura de los materiales
  - d. Midiendo el enfoque necesario para una visualización óptima del objeto
3. Hacer la reconstrucción en 3D del objeto. Es la descripción de un objeto a través de toda la información 3D que puede dar la(s) cámara(s).

Para una mayor información del desarrollo matemático de este problema se sugiere la lectura de la referencia [20]. Aquí el problema es explicado detalladamente por los autores. Para el tema de visión estereoscópica se recomienda [35][36][37]; para el problema de la detección por el método de texturas [38][39][40]; para la detección a través de un análisis del enfoque de la imagen leer referencias [41][42][43]; para análisis de profundidad a partir del movimiento se pueden revisar [44][45][46].



## Capítulo 8. Bibliografía

- [1]. Hauck, G.; Passig, J.; Rüttinger, M.; Sorg, G. y Farber; "Biologically motivated hand-eye coordination for the autonomous grasping of unknown objects". Universidad Técnica de Munich. En *Autonome Mobile Systeme*, Noviembre 1999.
- [2]. Recatalá G.; "Determinación, seguimiento y ejecución visual de agarres 2D usando una aproximación inspirada en comportamientos". Tesis Doctoral. Universitat Jaume, Castelló 2003.
- [3]. Mackenzie C.; Iberall T. "Advances in Psychology: The Grasping Hand". North Holland, Elsevier Science B.V., Holanda 1994.
- [4]. Nguyen, V. "The Synthesis of Stable Force-Closure Grasps". Technical Report MIT AI-TR-905, MIT AI Lab, 1986.
- [5]. D. Montana, "Contact stability for two fingered grasps," IEEE Trans. Robot. Automat., vol. 8, pp. 421–430, 1992
- [6]. Bicci, A. Kumar, V. "Robotic grasping and contact: a review". En memorias de Conferencia Internacional en Robótica y Automatización IEEE. USA. 2000
- [7]. Morales A., "Learning to predict grasp reliability with a multifinger robot hand by using visual features". Tesis doctoral, Departamento de Ciencias de la computación e Ingeniería, Universidad Jaume I, Castellón, España, Enero 2004.
- [8]. Hauck, A. and Färber, G. "Hybrid Hand-Eye Coordination with a Single Stationary Camera". In 4th Int. Conf. on Control, Automation, Robotics and Computer Vision (ICARCV'96). 1996
- [9]. Hauck A., J. Rüttinger. M. Sorg and G. Färber. "Visual Determination of 3D Grasping Points on Unknown Objects with a Binocular Camera System". En memorias de IEEE/RSJ IROS. 1999
- [10]. Stanley, Kevin. "A Hybrid Motion Vision Guided Robotic System with Image Based Grasp Planning". Tesis de maestría. Escuela de Ciencias Ingenieriles. Universidad Simon Fraser. Agosto 1999.
- [11]. Hauck, A. "Vision-based reach to grasp movements: from the human example to an autonomous robotic system" Ph.D. dissertation, Technische Universität at Munich, Alemania, 1999.
- [12]. Ramnath K., "A framework for robotic vision-based grasping task". Universidad de Carnegie Mellon. 2004.
- [13]. MIURA, K. "Robot Hand Positioning and Grasping Using Vision". Thèse franco-japonaise : Universidad de Strasbourg I, Universidad de Tohoku. Febrero 2004

- 
- [14]. Boudaba, M. Casals, A. y Woern H. "Grasping Points Determination Using Visual Features", *Vision Systems: Applications*, Goro Obinata and Ashish Dutta (Ed.), ISBN: 978-3-902613-01-1, I-Tech Education and Publishing, Austria 2007
- [15]. Sanz P., Traver J., Morales A., Recatalá G., "Razonamiento Geométrico para la determinación 2D de puntos de agarre". Universidad Jaume I. En VIII Encuentros de Geometría Computacional, Colección "Treballs d'Informàtica i Tecnologia", pages 394-401, Castelló, España, Julio, 1999
- [16]. Morales A., Sanz P., del Pobil A., "Vision-Based Computation of Three Finger Grasps of Unknown Planar Objects". En memorías del IEEE/RSJ IROS, pp. 1711-1716, Suiza 2002.
- [17]. J. Cornellà, R. Suárez. "Determining Independent Grasp Regions on 2D Discrete Objects". En 18th IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'2005, (ISBN 0-7803-8913-1), Edmonton, Alberta, Canada, August 2-6, 2005, pp.2936-2941.
- [18]. Ponce J. y Faverjon B. "On computing three-finger force-closure grasps of polygonal objects". En IEEE Transactions on Robotics and Automation, 11(6):868–881, December 1995.
- [19]. Coulomb C. "Théorie des machines simples en ayant égard au frottement de leurs parties et à roideur des cordages" Mémoires de Mathématique et de Physique présentés à l'Académie Royale des Sciences, par divers savans, Paris, 1781.
- [20]. Pajares G.; De la Cruz J. "Visión por Computador". Alfaomega. Primera Edición, México 2004.
- [21]. Davies R. "Machine Vision: Theory, Algorithms Practicalities". Morgan Kaufman, Tercera Edición, EUA 2005.
- [22]. Litwiller D. "CCD vs CMOS: Facts and Fictions", *Photonics Spectra*, Enero 2001.
- [23]. <http://www.twenga.es/precios-VideoCAM-GE111-GENIUS-Webcam-173152-0>
- [24]. Ricolfe C, Sánchez A y Simarro R. "Técnicas de Calibrado de Cámaras".XXIV Jornadas de Automática. Universidad de León. España 2003.
- [25]. De La Escalera A. "Visión por Computador: Fundamentos y Métodos". Prentice Hall, España 2001.
- [26]. Sezgin, M.; Sankur, B.; "Survey over image thresholding techniques and quantitative performance evaluation" . En *Journal of Electronic Imaging*. Enero de 2004
- [27]. Kotoulas L., Andreades I. "image Analisis using Moments". En 5<sup>th</sup> Conferencia Internacional en Tecnología y Automatización. Grecia 2005.
- [28]. Gonzalez R.; Woods R. "Digital Image Processing". Prentice Hall, Segunda Edición. USA 2002.
- [29]. Milan S.; Hlavac V.; Boyle R. "Image Processing, Analysis and Machine Vision". Thomson Learning. USA 2008.
- [30]. Schowengerdt R. Digital Image Processing class notes. Digital image analysis laboratory. Universidad de Arizona, USA 2003.

- [31]. Abreu E., Lighthstone M., Mitra S., Arakawa K. "A new efficient approach for the removal of impulse noise from highly corrupted images". *IEEE Transactions on Image Processing*. 1012 – 1025.1996
- [32]. Cantero J. *Medidas de forma: Asimetría y curtosis. Momentos*. Material de clase. Escuela Universitaria de Osuna. España 2008
- [33]. A.J. Calabria and M.D. Fairchild, "Perceived image contrast and observer preference II: Empirical Modeling of perceived image contrast and observer preference data," *En Journal of Imaging Science & Technology*, 47 494-508 (2003).
- [34]. P. Liao, T. Chen and P. Chung, "A Fast Algorithm for Multilevel Thresholding", *Journal of Information. Science and Engineering*, 17, 2001.
- [35]. Koch R, "3-D Surface Reconstruction from Stereoscopic Image Sequences" *Memorias de Fifth International Conference on Computer Vision, USA 1995*.
- [36]. Zhao, Y; Chen Z; Zhu C; Tan Y; Yu L; , "Binocular Just-Noticeable-Difference Model for Stereoscopic Images," *Signal Processing Letters, IEEE* , vol.18, no.1, pp.19-22, Jan. 2011
- [37]. T. Alexander, J. Conradi, C. Winkelholz, "Depth Perception and Visual After-effects at Stereoscopic Workbench Displays," *vr*, pp.269, *IEEE Virtual Reality Conference 2003 (VR 2003)*, 2003
- [38]. Romdhani, S, Vetter, T "Estimating 3D Shape and Texture Using Pixel Intensity, Edges, Specular Highlights, Texture Constraints and a Prior," *cvpr*, vol. 2, pp.986-993, 2005 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, 2005
- [39]. Merckel, L, Nishida, T "Low-Overhead Texture Mapping on 3D Models," *cit*, vol. 1, pp.128-133, 2009 *Ninth IEEE International Conference on Computer and Information Technology*, 2009
- [40]. Wang, J, Zhai Z, Cai X, "Real-Time Continuous Collision Detection Based on Swept Volume and Depth Texture," *icat*, pp.137-140, 16th *International Conference on Artificial Reality and Telexistence--Workshops (ICAT'06)*, 2006
- [41]. Darrell, T.; Wohn, K.; , "Pyramid based depth from focus," *Computer Vision and Pattern Recognition, 1988. Proceedings CVPR '88., Computer Society Conference on* , vol., no., pp.504-509, 5-9 Junio 1988
- [42]. Junlan Yang; Schonfeld, D.; , "Virtual Focus and Depth Estimation From Defocused Video Sequences," *Image Processing, IEEE Transactions on* , vol.19, no.3, pp.668-679, Marzo 2010
- [43]. Hiura, S.; Matsuyama, T.; , "Depth measurement by the multi-focus camera," *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on* , vol., no., pp.953-959, 23-25 Junio 1998
- [44]. Xiaoming Lu; Mei Yu; Yun Zhang; Gangyi Jiang; , "Motion detection based on temporal-to-spatial conversion of depth maps for multi-view video," *Wireless Communications & Signal Processing, 2009. WCSP 2009. International Conference on* , vol., no., pp.1-5, 13-15 Noviembre. 2009
- [45]. Ono, H.; , "Perception of Depth, Motion, and Stability with Motion Parallax (Invited Paper)," *Universal Communication, 2008. ISUC '08. Second International Symposium on* , vol., no., pp.193-198, 15-16 Diciembre. 2008

- [46]. Ohya, I.; Kosaka, A.; Kak, A.; , "Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing," *Robotics and Automation, IEEE Transactions on* , vol.14, no.6, pp.969-978, Diciembre 1998

**Ing. PABLO EDUARDO CAICEDO RODRÍGUEZ**



**APREHENSION ESTABLE CON UNA MANO ROBOTICA USANDO  
INFORMACIÓN VISUAL**

**Universidad del Cauca  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Maestría en Electrónica y Telecomunicaciones**

**Popayán  
2010**



**PABLO EDUARDO CAICEDO RODRÍGUEZ**

**APREHENSION ESTABLE CON UNA MANO ROBOTICA USANDO  
INFORMACIÓN VISUAL**

**Tesis presentada a la Facultad de Ingeniería  
Electrónica y Telecomunicaciones de la  
Universidad del Cauca para la obtención del  
Título de**

**Magíster en Ingeniería, Área  
Electrónica y Telecomunicaciones**

**Director:  
Jaime Oscar Díaz Ch.  
Magister**

**Popayán  
2010**



## Anexo A

En las siguientes tablas se encuentra una descripción cualitativa de las imágenes utilizadas para comprobar del buen funcionamiento de los algoritmos.

- Las imágenes 004 a 048 contienen un solo objetivo (de color variado) y fueron empleadas para comparar el algoritmo basado en curtosis y el algoritmo de Otsu.
- Con las imágenes 049 a 100, se prueba la capacidad del algoritmo para diferenciar colores.
- Con las imágenes 101 a 111 se demuestra la habilidad del algoritmo para procesar objetos multicolores (con estampado).
- Con las imágenes 112 a 116 se evidencia que el algoritmo es capaz de procesar objetos de forma indefinida.

**Nota:** *Ok* significa que el algoritmo cumple su objetivo y *Erro* hace referencia a que no realiza su cometido.

Imagen 004			
Colores		Curtosis	Otsu
Matiz	43	Ok	Ok
Saturación	240		
Luminancia	30		
Forma	Constante		

Imagen 005			
Colores		Curtosis	Otsu
Matiz	43	Ok	Ok
Saturación	240		
Luminancia	30		
Forma	Constante		

Imagen 006			
Colores		Curtosis	Otsu
Matiz	43	Ok	Ok
Saturación	240		
Luminancia	50		
Forma	Constante		

Imagen 007			
Colores		Curtosis	Otsu
Matiz	43	Ok	Ok
Saturación	240		

Luminancia	65		
Forma		Constante	

Imagen 008			
Colores		Curtosis	Otsu
Matiz	43	Ok	Ok
Saturación	240		
Luminancia	75		
Forma		Constante	

Imagen 009			
Colores		Curtosis	Otsu
Matiz	43	Ok	Error
Saturación	240		
Luminancia	95		
Forma		Constante	

Imagen 010			
Colores		Curtosis	Otsu
Matiz	43	Error	Error
Saturación	240		
Luminancia	96		
Forma		Constante	

Imagen 011			
Colores		Curtosis	Otsu
Matiz	43	Error	Error
Saturación	240		
Luminancia	98		
Forma		Constante	

Imagen 012			
Colores		Curtosis	Otsu
Matiz	43	Error	Error
Saturación	240		
Luminancia	97		
Forma		Constante	

Imagen 013			
Colores		Curtosis	Otsu
Matiz	43	Error	Error
Saturación	240		
Luminancia	96		
Forma		Constante	

Imagen 014			
Colores		Curtosis	Otsu
Matiz	43	Error	Error
Saturación	240		
Luminancia	96		
Forma		Constante	

Imagen 015			
Colores		Curtosis	Otsu
Matiz	43	Ok	Error
Saturación	230		
Luminancia	96		
Forma		Constante	

Imagen 016			
Colores		Curtosis	Otsu
Matiz	43	Ok	Error
Saturación	215		
Luminancia	96		

Forma	Constante
-------	-----------

Imagen 017			
Colores		Curtosis	Otsu
Matiz	43	Ok	Error
Saturación	200		
Luminancia	96		
Forma	Constante		

Imagen 018			
Colores		Curtosis	Otsu
Matiz	43	Ok	Error
Saturación	175		
Luminancia	96		
Forma	Constante		

Imagen 019			
Colores		Curtosis	Otsu
Matiz	43	Ok	Ok
Saturación	150		
Luminancia	96		
Forma	Constante		

Imagen 020			
Colores		Curtosis	Otsu
Matiz	43	Error	Ok
Saturación	100		
Luminancia	96		
Forma	Constante		

Imagen 021			
Colores		Curtosis	Otsu
Matiz	43	Ok	Ok
Saturación	80		
Luminancia	96		
Forma	Constante		

Imagen 022			
Colores		Curtosis	Otsu
Matiz	43	Ok	Ok
Saturación	110		
Luminancia	96		
Forma	Constante		

Imagen 023			
Colores		Curtosis	Otsu
Matiz	43	Ok	Ok
Saturación	70		
Luminancia	96		
Forma	Constante		

Imagen 024			
Colores		Curtosis	Otsu
Matiz	43	Ok	Ok
Saturación	60		
Luminancia	96		
Forma	Constante		

Imagen 025			
Colores		Curtosis	Otsu
Matiz	43	Ok	Ok
Saturación	50		
Luminancia	96		
Forma	Constante		

Imagen 026			
Colores		Curtosis	Otsu
Matiz	43	Ok	Ok
Saturación	20		
Luminancia	96		
Forma	Constante		

Imagen 027			
Colores		Curtosis	Otsu
Matiz	43	Ok	Ok
Saturación	0		
Luminancia	96		
Forma	Constante		

Imagen 028			
Colores		Curtosis	Otsu
Matiz	240	Error	Ok
Saturación	240		
Luminancia	96		
Forma	Constante		

Imagen 029			
Colores		Curtosis	Otsu
Matiz	230	Ok	Ok
Saturación	240		
Luminancia	96		
Forma	Constante		

Imagen 030			
Colores		Curtosis	Otsu
Matiz	220	Ok	Ok
Saturación	240		
Luminancia	96		
Forma	Constante		

Imagen 031			
Colores		Curtosis	Otsu
Matiz	210	Ok	Ok
Saturación	240		
Luminancia	96		
Forma	Constante		

Imagen 032			
Colores		Curtosis	Otsu
Matiz	200	Ok	Ok
Saturación	240		
Luminancia	96		
Forma	Constante		

Imagen 033			
Colores		Curtosis	Otsu
Matiz	180	Ok	Ok
Saturación	240		
Luminancia	96		
Forma	Constante		

Imagen 034			
Colores		Curtosis	Otsu
Matiz	160	Ok	Ok
Saturación	240		
Luminancia	96		
Forma	Constante		

Imagen 035			
Colores		Curtosis	Otsu
Matiz	130	Ok	Ok
Saturación	240		
Luminancia	96		
Forma	Constante		

Imagen 036			
Colores		Curtosis	Otsu
Matiz	100	Ok	Ok
Saturación	240		
Luminancia	96		
Forma	Constante		

Imagen 037			
Colores		Curtosis	Otsu
Matiz	50	Ok	Ok
Saturación	240		
Luminancia	96		
Forma	Constante		

Imagen 038			
Colores		Curtosis	Otsu
Matiz	0	Error	Error
Saturación	240		
Luminancia	96		
Forma			

Imagen 039			
Colores		Curtosis	Otsu
Matiz	160	NA	NA
Saturación	0		
Luminancia	0		
Forma	Triángulo		

Imagen 040			
Colores		Curtosis	Otsu
Matiz	160	NA	NA
Saturación	0		
Luminancia	0		
Forma	Cuadrado		

Imagen 041			
Colores		Curtosis	Otsu
Matiz	160	NA	NA
Saturación	0		
Luminancia	0		
Forma	Pentágono		

Imagen 042			
Colores		Curtosis	Otsu
Matiz	160	NA	NA
Saturación	0		
Luminancia	0		
Forma	Hexágono		

Imagen 043			
Colores		Curtosis	Otsu
Matiz	160	NA	NA
Saturación	0		
Luminancia	0		
Forma	Heptágono		

Imagen 044			
------------	--	--	--

Colores		Curtosis	Otsu
Matiz	160	NA	NA
Saturación	0		
Luminancia	0		
Forma	Figura Irregular 1		

Imagen 045			
Colores		Curtosis	Otsu
Matiz	160	NA	NA
Saturación	0		
Luminancia	0		
Forma	Octágono		

Imagen 046			
Colores		Curtosis	Otsu
Matiz	160	NA	NA
Saturación	0		
Luminancia	0		
Forma	Figura Irregular 2		

Imagen 047			
Colores		Curtosis	Otsu
Matiz	160	NA	NA
Saturación	0		
Luminancia	0		
Forma	Eneágono		

Imagen 048			
Colores		Curtosis	Otsu
Matiz	160	NA	NA
Saturación	0		
Luminancia	0		
Forma	Decágono		

Im	Objeto			Fondo			Curtosis	Otsu
	Matiz	Saturación	Luminancia	Matiz	Saturación	Luminancia		
049	160	0	0	160	0	184	Ok	Ok
050	160	0	0	160	0	89	Ok	Ok
051	160	0	0	160	0	52	Ok	Ok
052	160	0	0	160	0	21	Ok	Ok
053	160	0	0	160	0	6	Ok	Ok
054	160	0	0	160	0	2	Ok	Ok
055	240	240	233	80	240	233	Error	Error
056	230	240	233	80	240	233	Error	Error
057	220	240	233	80	240	233	Error	Error
058	210	240	233	80	240	233	Error	Error
059	200	240	233	80	240	233	Error	Error
060	150	240	233	80	240	233	Error	Error
061	100	240	233	80	240	233	Error	Error
062	50	240	233	80	240	233	Error	Error
063	80	230	233	80	240	233	Error	Error
064	80	208	233	80	240	233	Error	Error
065	80	176	233	80	240	233	Error	Error
066	80	144	233	80	240	233	Error	Error
067	80	112	233	80	240	233	Error	Error
068	80	34	233	80	240	233	Error	Error
069	80	0	233	80	240	233	Error	Error
070	80	240	215	80	240	233	Error	Error
071	80	240	200	80	240	233	Error	Error
072	80	240	175	80	240	233	Error	Error
073	80	240	150	80	240	233	Error	Error
074	80	240	125	80	240	233	Ok	Error
075	80	240	100	80	240	233	Ok	Error
076	80	240	70	80	240	233	Ok	Ok

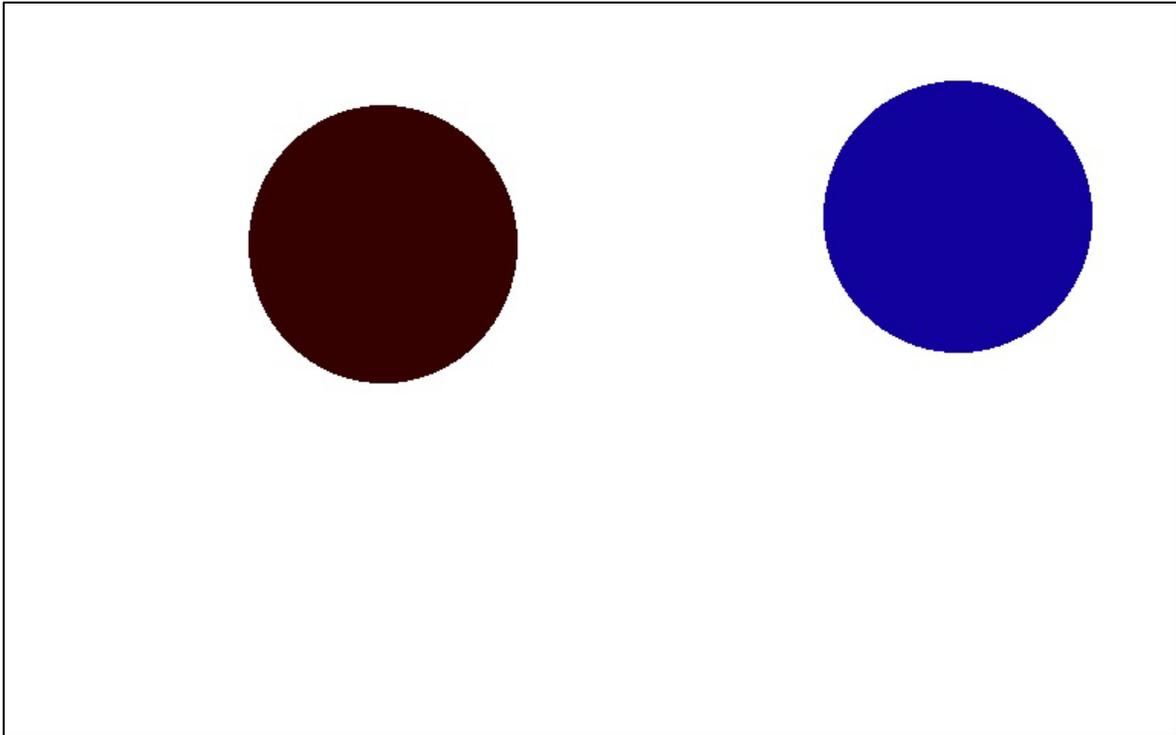
077	80	240	50	80	240	233	Ok	Ok
078	80	240	25	80	240	233	Ok	Ok
079	80	240	0	80	240	233	Ok	Ok
080	30	240	2	0	240	2	Ok	Error
081	50	240	2	0	240	2	Ok	Error
082	70	240	2	0	240	2	Error	Error
083	100	240	2	0	240	2	Error	Error
084	120	240	2	0	240	2	Error	Error
085	150	240	2	0	240	2	Error	Error
086	180	240	2	0	240	2	Error	Error
087	200	240	2	0	240	2	Error	Error
088	220	240	2	0	240	2	Error	Error
089	230	240	2	0	240	2	Error	Error
090	0	120	2	0	240	2	Ok	Error
091	0	240	25	0	240	2	Ok	Error
092	0	240	50	0	240	2	Ok	Ok
093	0	240	75	0	240	2	Ok	Ok
094	0	240	100	0	240	2	Ok	Ok
095	0	240	125	0	240	2	Ok	Ok
096	0	240	150	0	240	2	Ok	Ok
097	0	240	175	0	240	2	Ok	Ok
098	0	240	200	0	240	2	Ok	Ok
099	0	240	225	0	240	2	Ok	Ok
100	0	240	240	0	240	2	Ok	Ok

Objetos multicolor								
Im	Objeto1			Objeto2			Curtosis	Otsu
	Matiz	Saturación	Luminancia	Matiz	Saturación	Luminancia		
101	0	240	225	0	240	200	Error	Ok
102	0	240	225	0	240	175	Error	Ok
103	0	240	225	0	240	150	Ok	Ok
104	0	240	225	0	240	125	Error	Ok
105	0	240	225	0	240	100	Ok	Ok
106	0	240	225	0	240	75	Ok	Ok
107	0	240	225	0	240	50	Error	Ok
108	0	240	75	160	240	75	Ok	Ok
109	0	240	150	160	240	150	Ok	Ok
110	0	240	200	160	240	200	Error	Ok
111	0	240	200	80	240	200	Error	Ok
112	Objeto Estampado 1						Ok	Ok
113	Objeto Estampado 2 Multicolor Fondo rosado						Ok	Ok
114	Objeto Estampado 3 Fondo gris						Ok	Ok
115	MultiObjeto 1 Con objetos en esquina, selección intermedios						Ok	Ok
116	MultiObjeto 2 Con Objetos en esquina, selección esquina						Ok	Ok

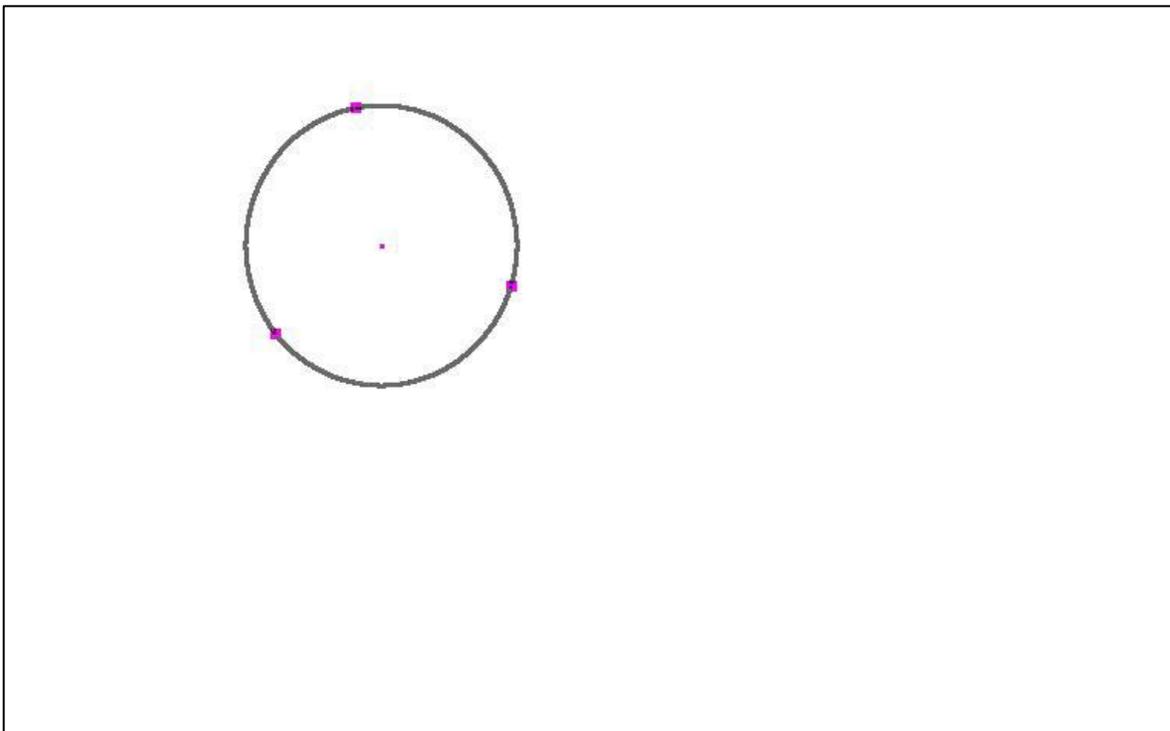


## Anexo B

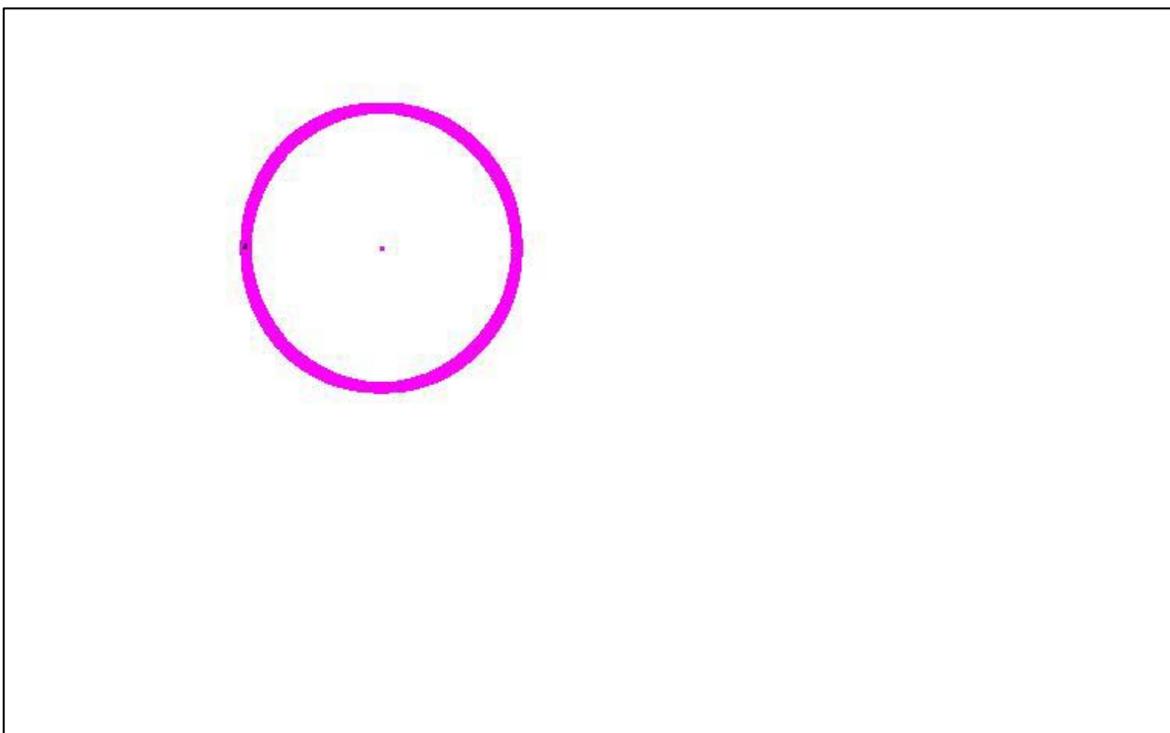
Las siguientes figuras son imágenes de prueba y su correspondiente resultado del procesamiento tanto para dos como para tres dedos. Es importante anotar que se seleccionaron píxeles cuyas distancias sean mínimas, de acuerdo a esto la **Figura Anexo 3**, **Figura Anexo 6** y **Figura Anexo 12** constituyen ejemplos de imágenes que tienen múltiples zonas de aprehensión a igual distancia.



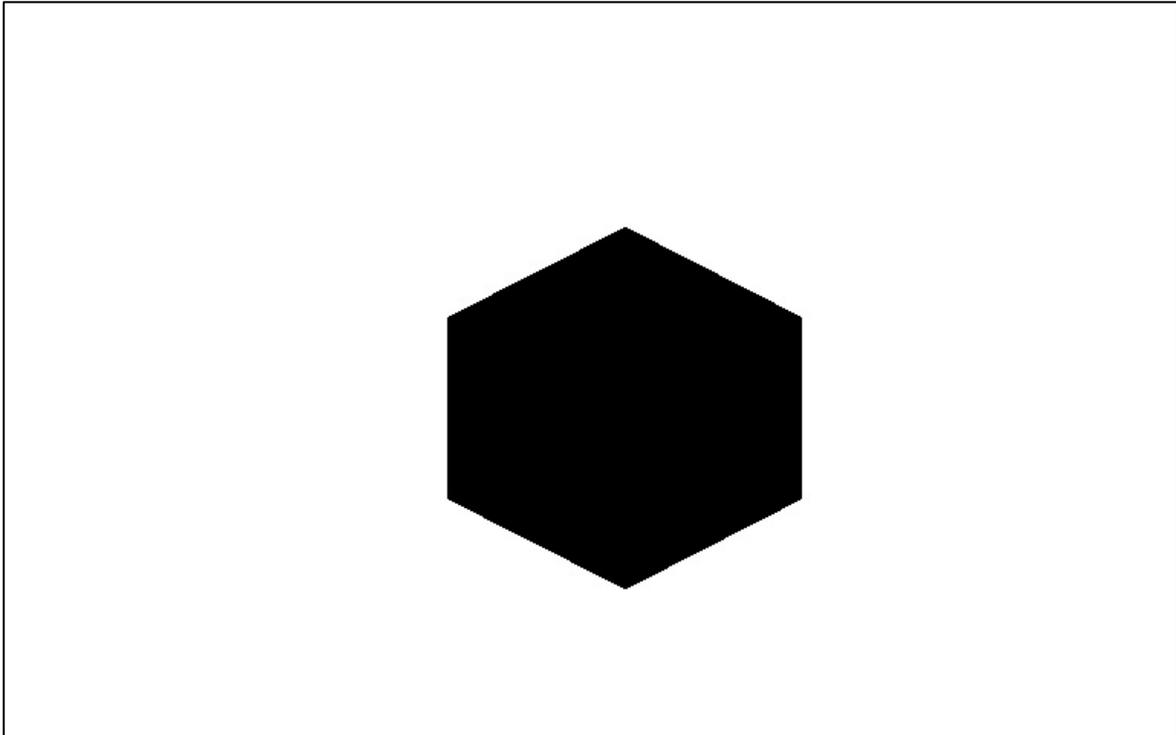
**Figura Anexo 1.** Primera Imagen de prueba



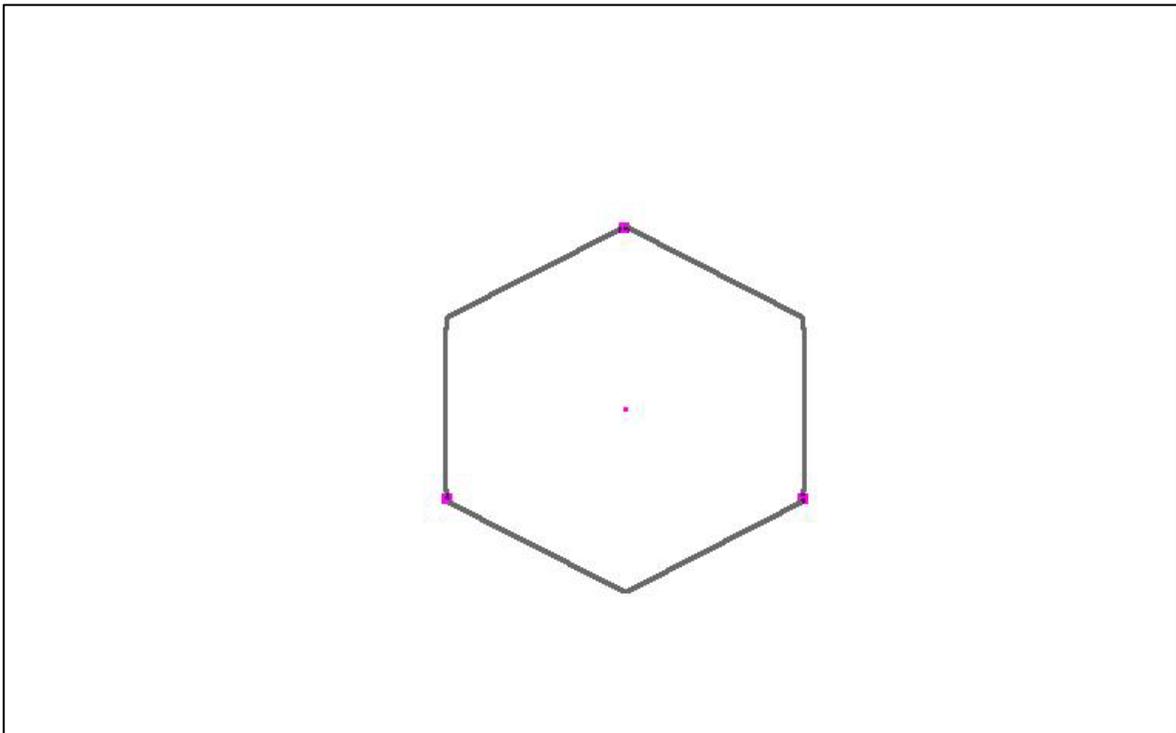
**Figura Anexo 2.** Resultado para algoritmo de detección de puntos de aprehensión estable para agarre con 3 dedos de Figura Anexo 1



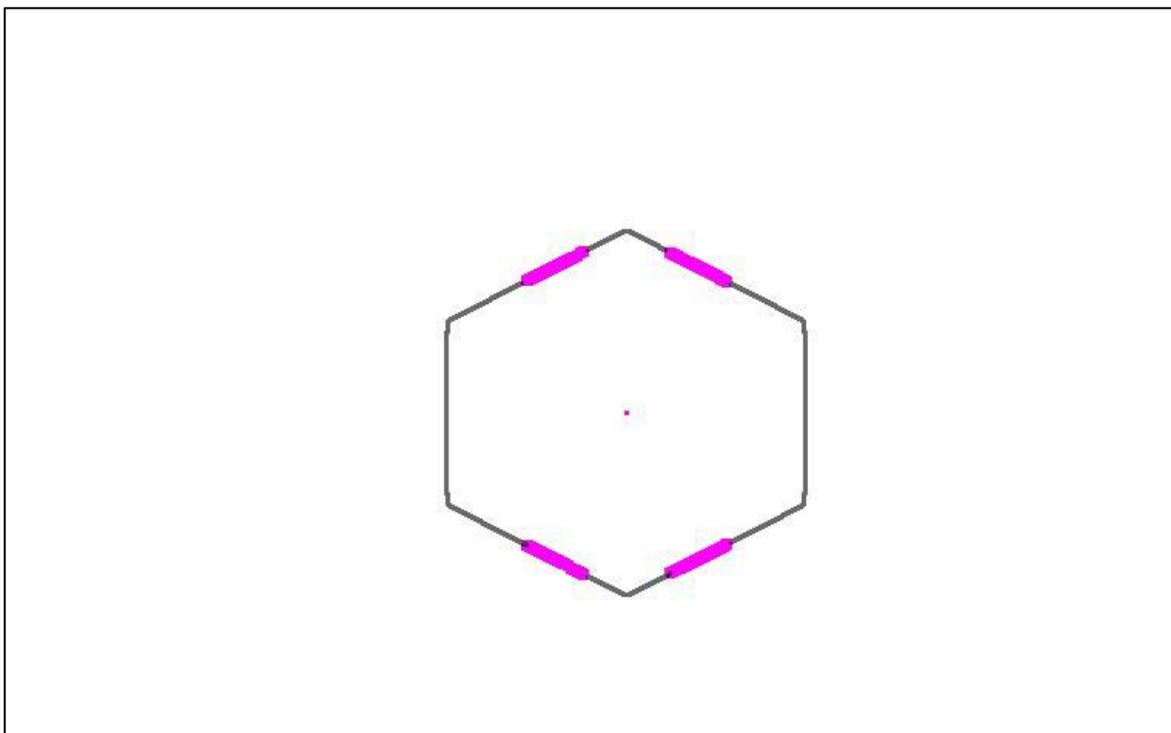
**Figura Anexo 3.** Resultado para algoritmo de detección de puntos de aprehensión estable para agarre con 2 dedos de Figura Anexo 1



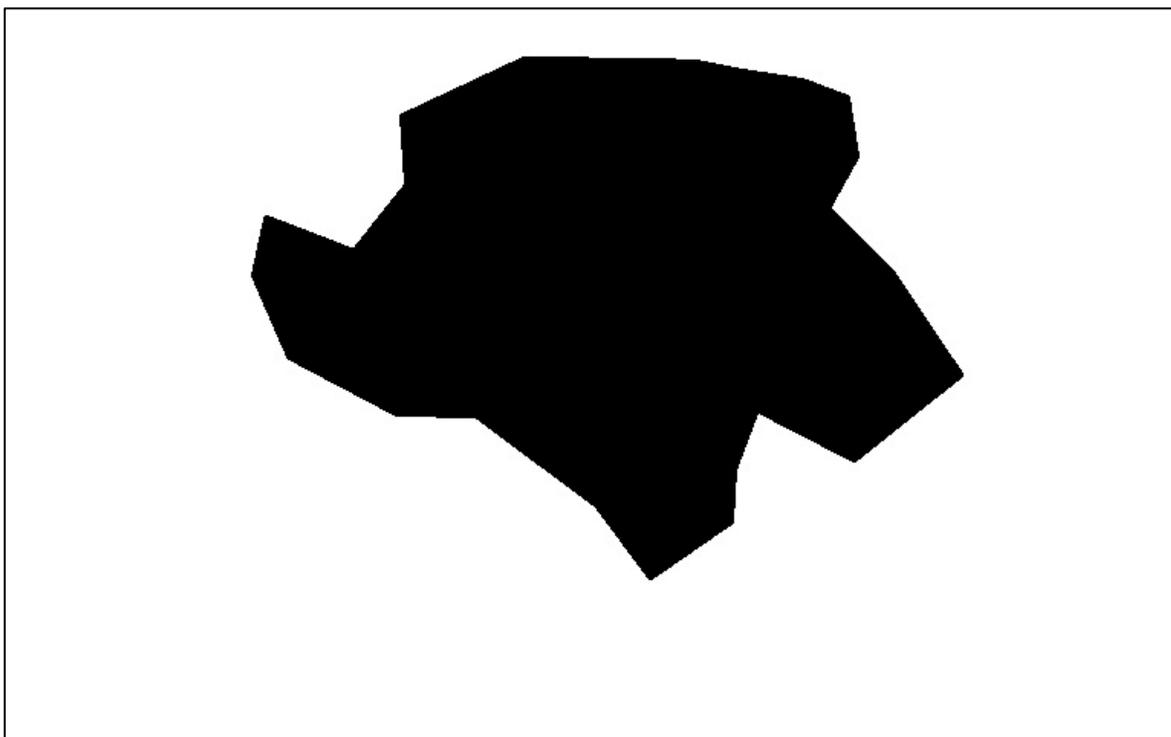
**Figura Anexo 4.** Segunda Imagen de prueba



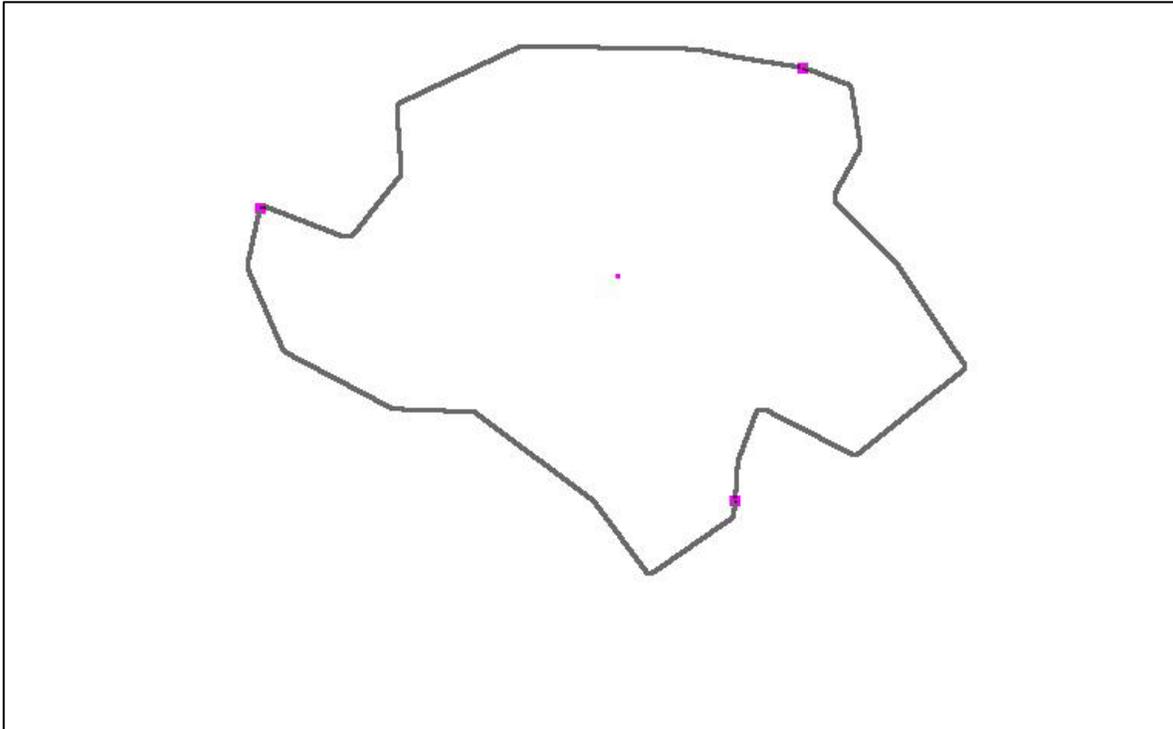
**Figura Anexo 5.** Resultado para algoritmo de detección de puntos de aprehensión estable para agarre con 3 dedos de Figura Anexo 4



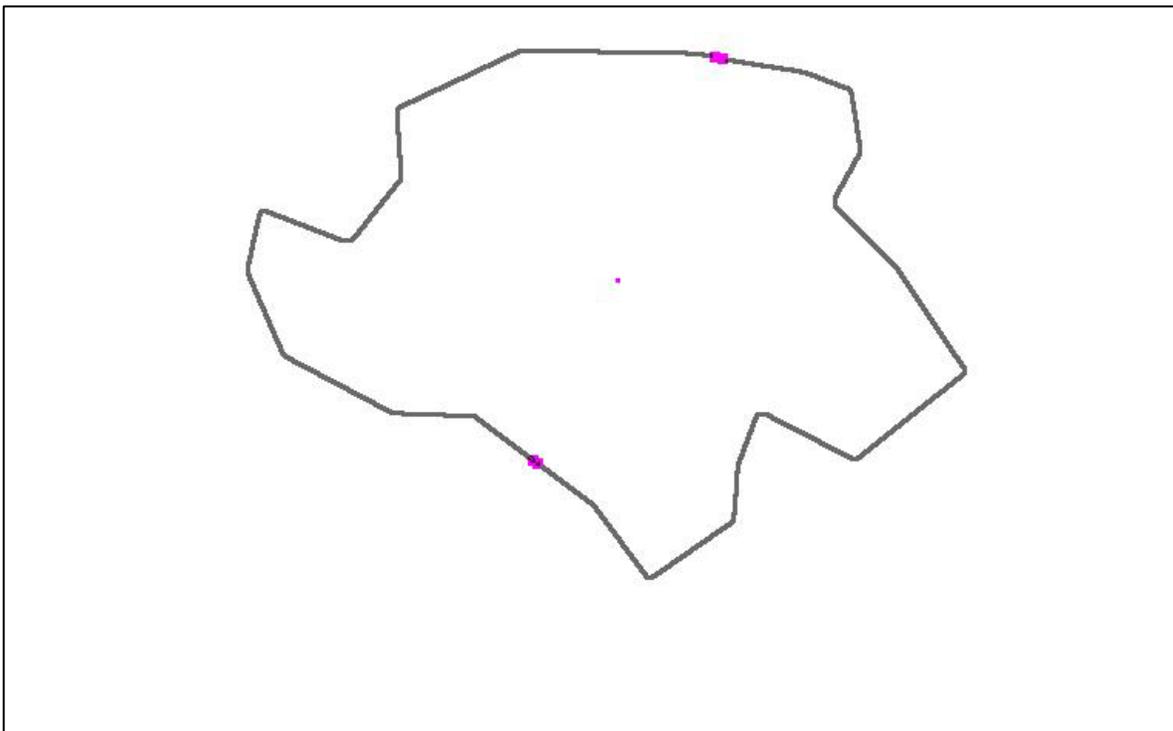
**Figura Anexo 6.** Resultado para algoritmo de detección de puntos de aprehensión estable para agarre con 2 dedos de Figura Anexo 4



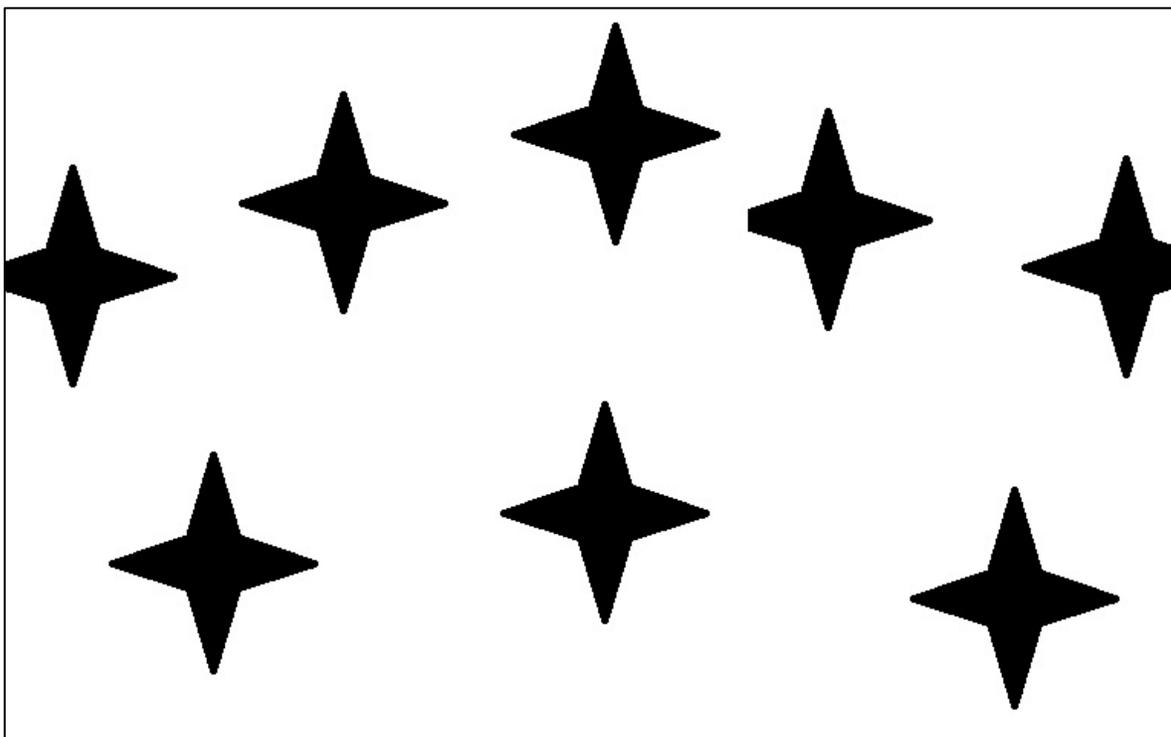
**Figura Anexo 7.** Tercera Imagen de prueba



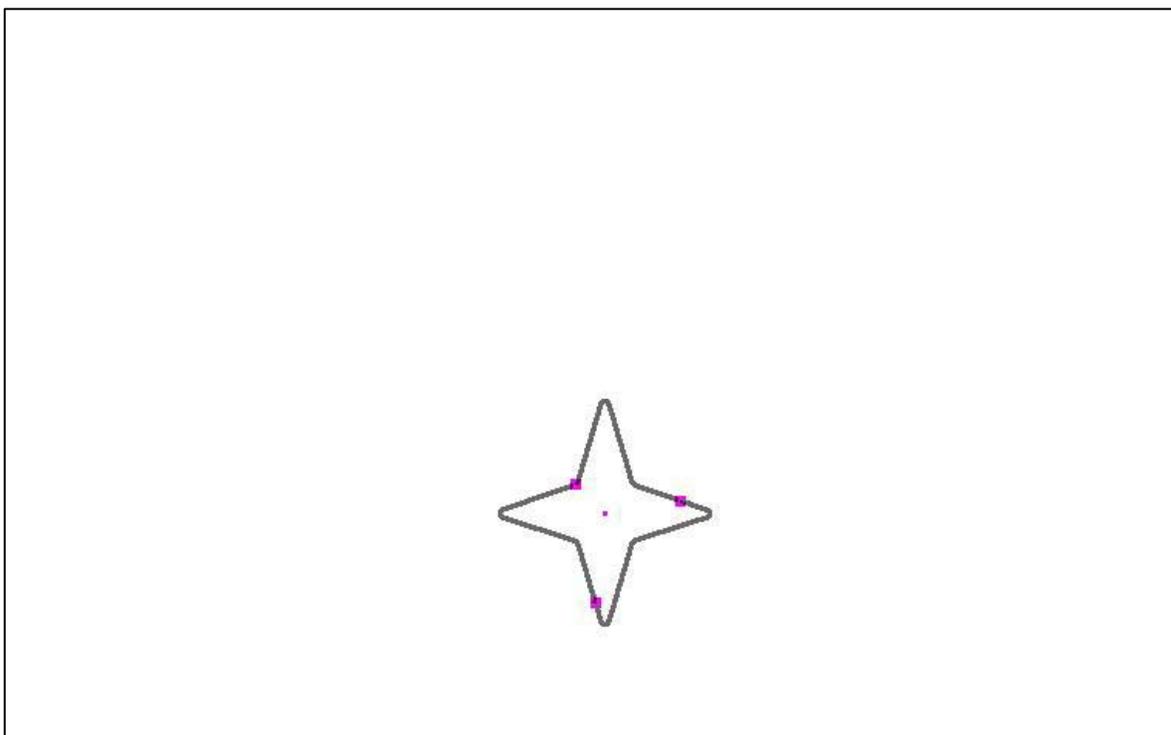
**Figura Anexo 8.** Resultado para algoritmo de detección de puntos de aprehensión estable para agarre con 3 dedos de Figura Anexo 7



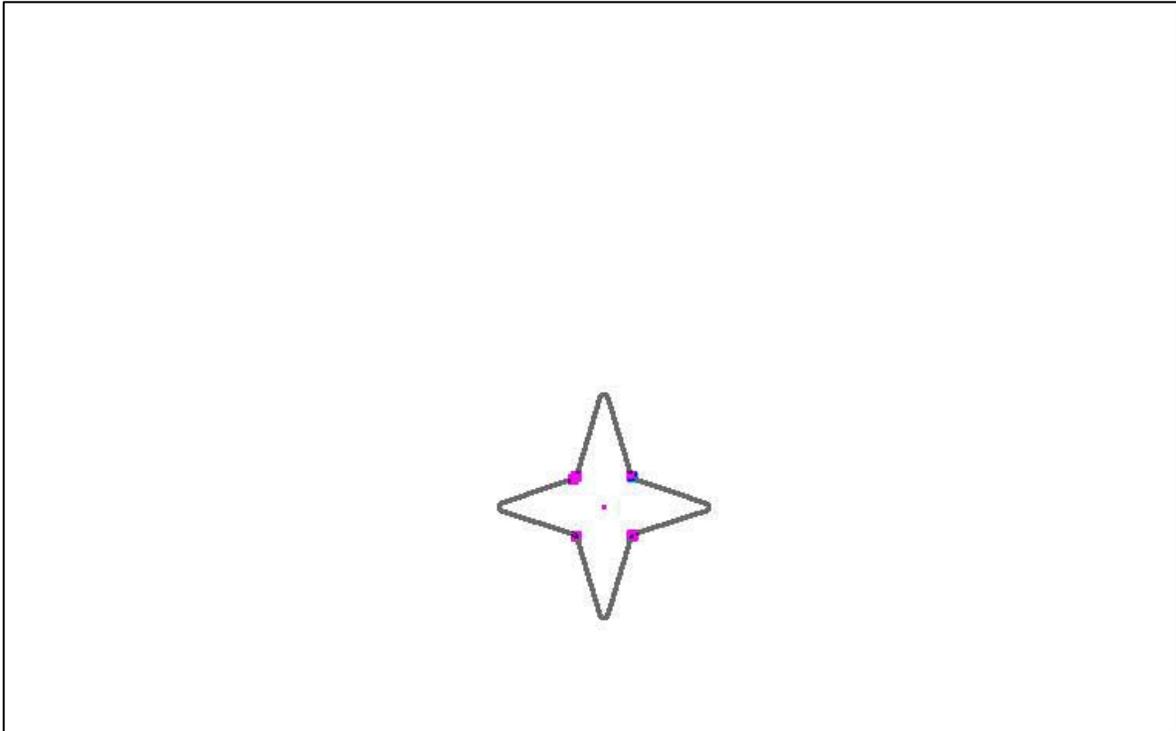
**Figura Anexo 9.** Resultado para algoritmo de detección de puntos de aprehensión estable para agarre con 2 dedos de Figura Anexo 7



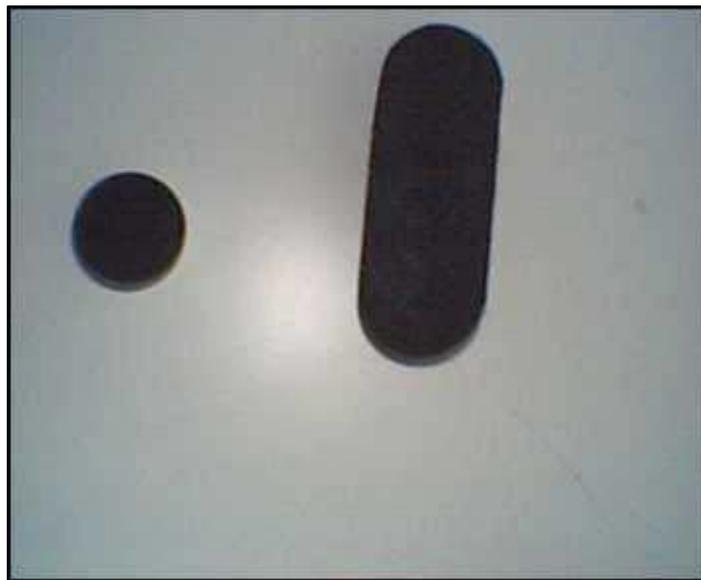
**Figura Anexo 10.** Cuarta Imagen de prueba



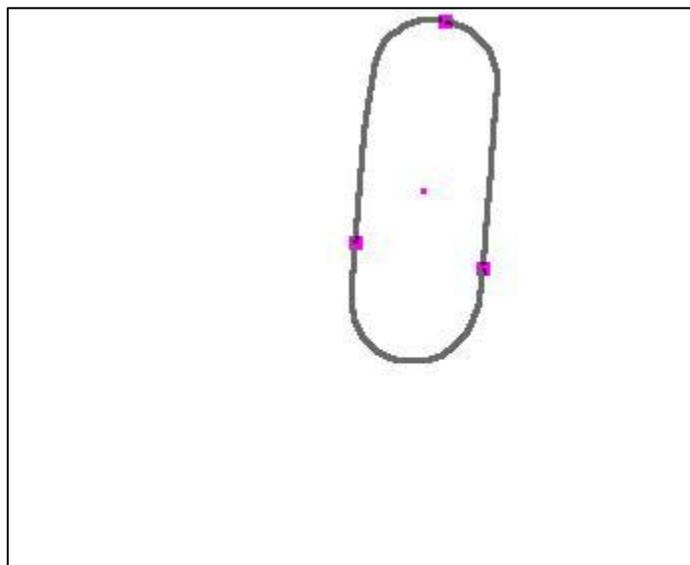
**Figura Anexo 11.** Resultado para algoritmo de detección de puntos de aprehensión estable para agarre con 3 dedos de Figura Anexo 10



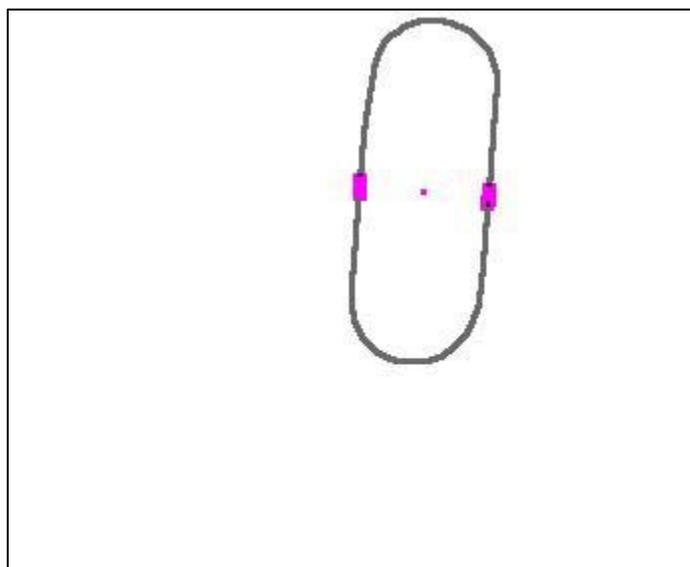
**Figura Anexo 12.** Resultado para algoritmo de detección de puntos de aprehensión estable para agarre con 2 dedos de Figura Anexo 10



**Figura Anexo 13.** Quinta Imagen de prueba



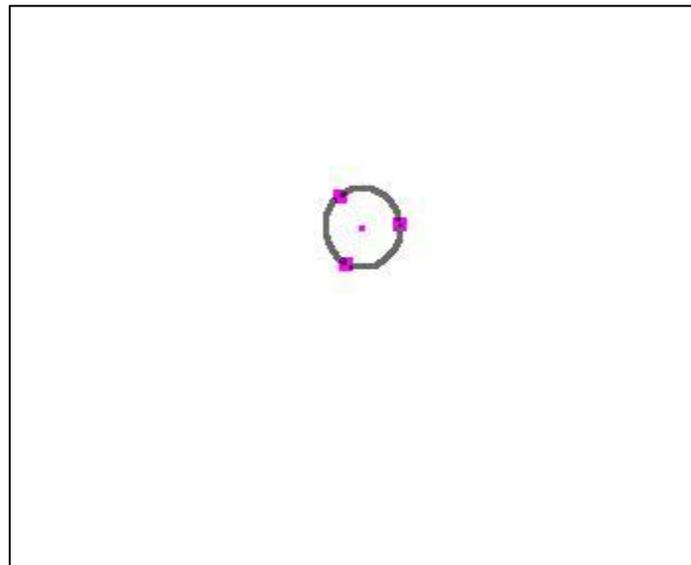
**Figura Anexo 14.** Resultado para algoritmo de detección de puntos de aprehensión estable para agarre con 3 dedos de Figura Anexo 13



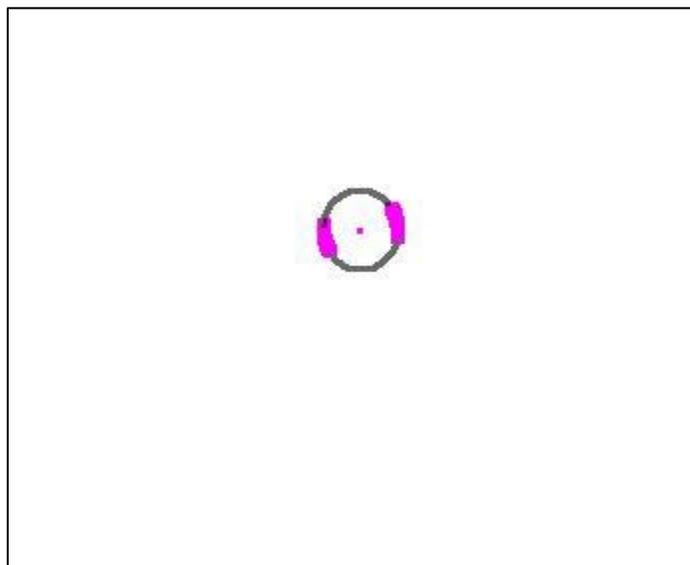
**Figura Anexo 15.** Resultado para algoritmo de detección de puntos de aprehensión estable para agarre con 2 dedos de Figura Anexo 13



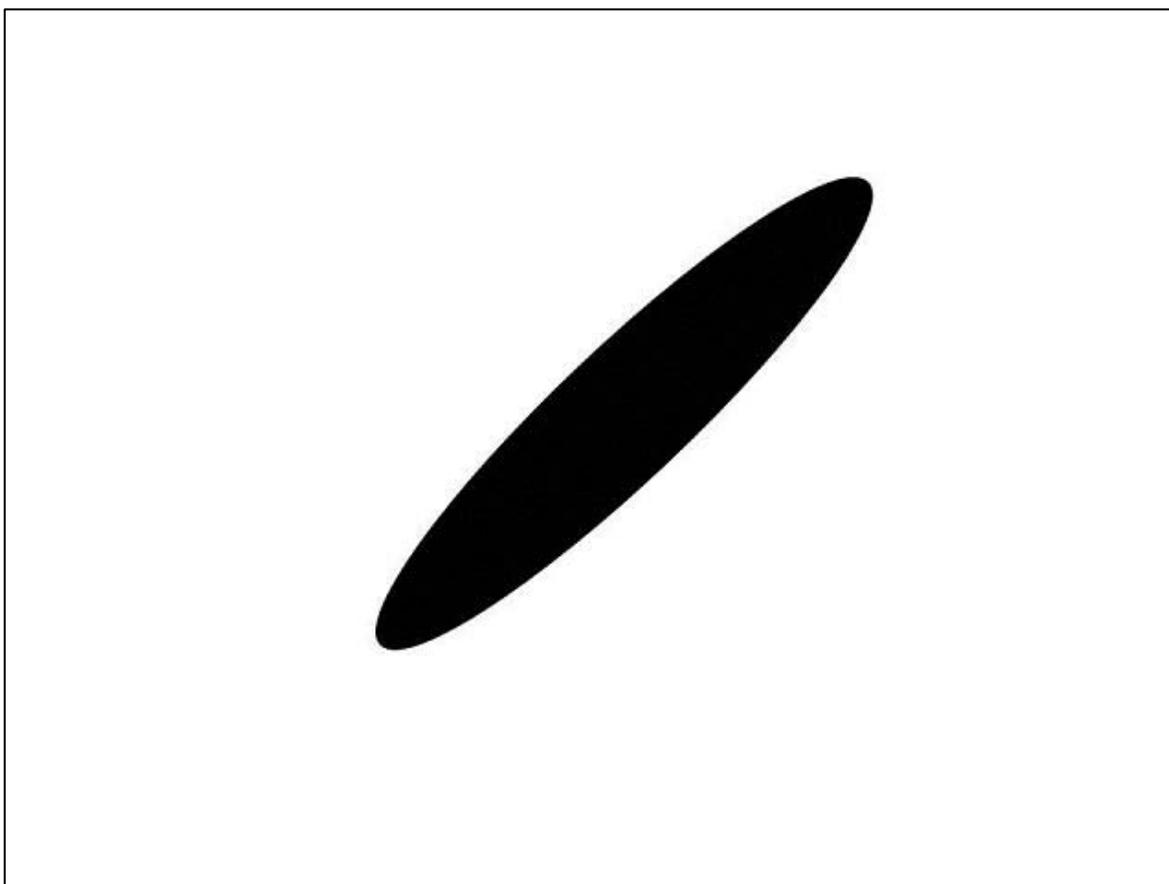
**Figura Anexo 16.** Sexta Imagen de prueba



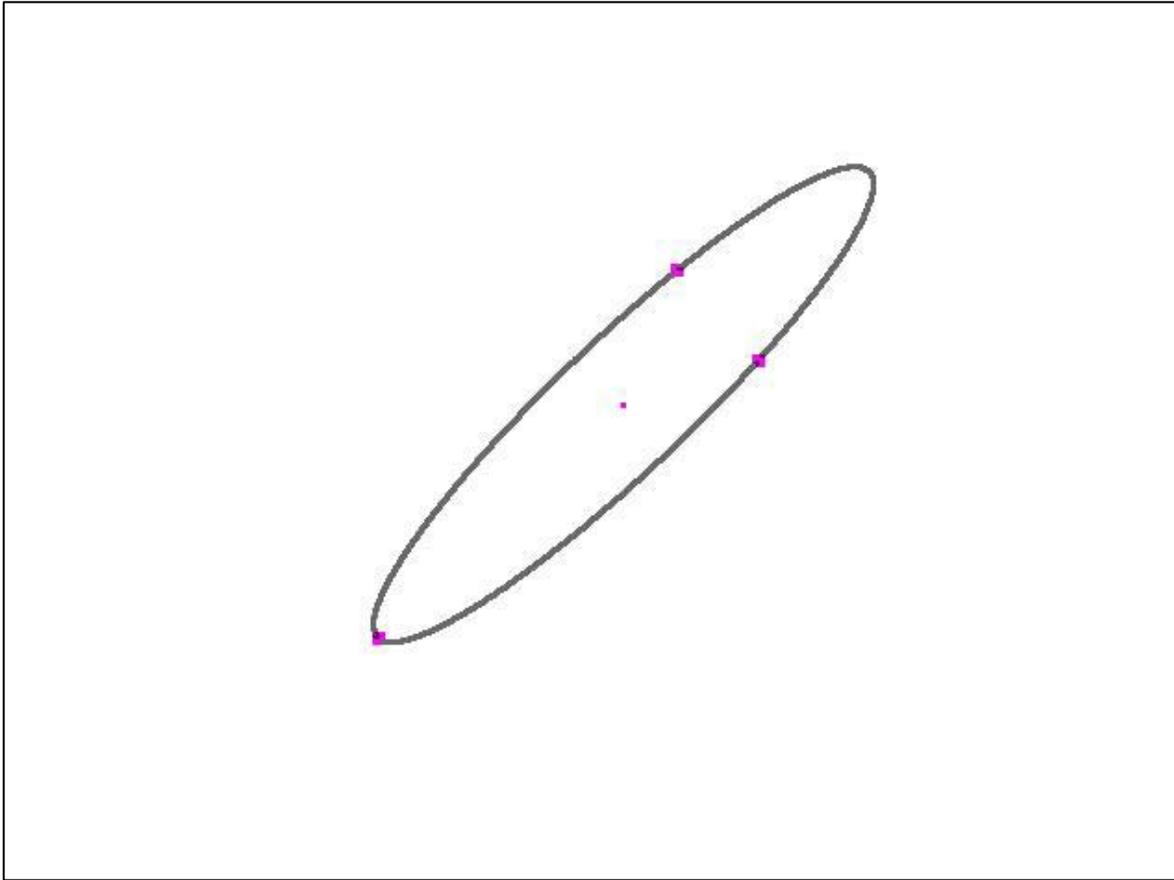
**Figura Anexo 17.** Resultado para algoritmo de detección de puntos de aprehensión estable para agarre con 3 dedos de Figura Anexo 16



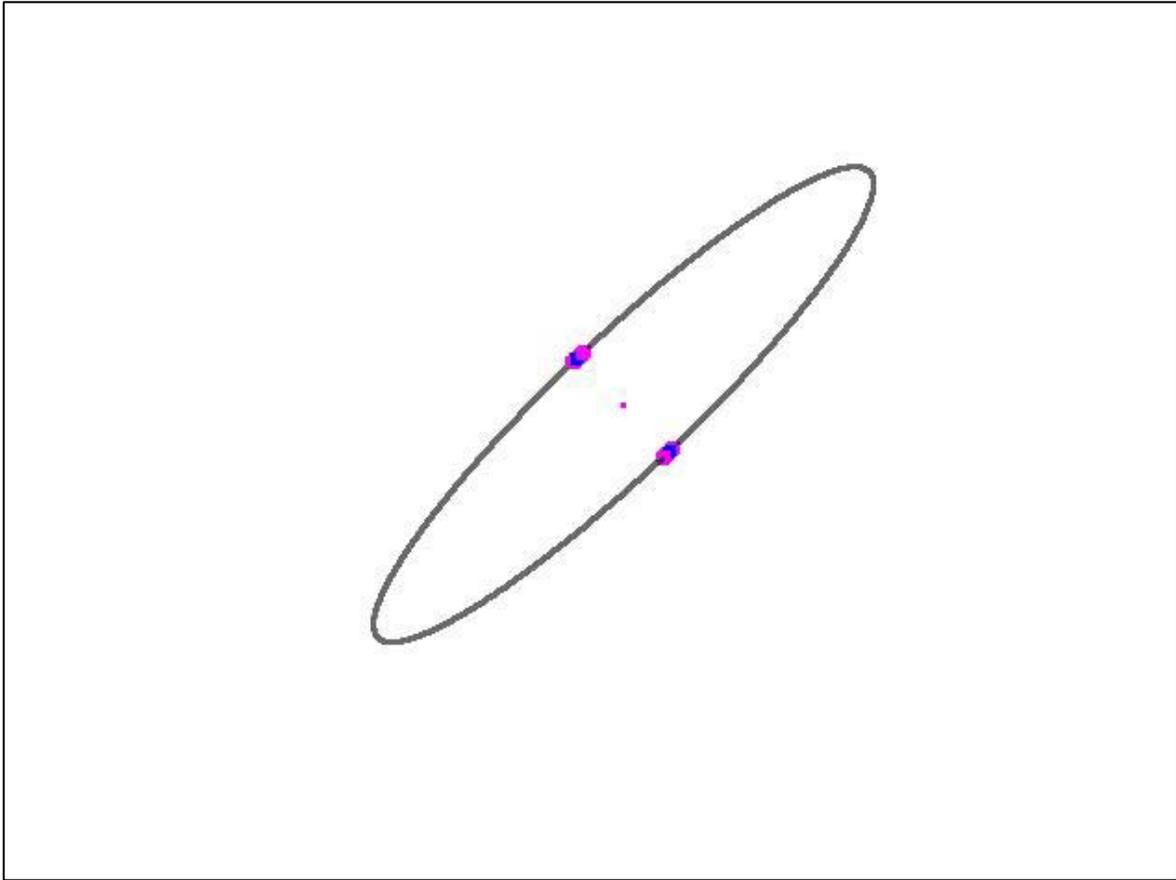
**Figura Anexo 18.** Resultado para algoritmo de detección de puntos de aprehensión estable para agarre con 2 dedos de Figura Anexo 16



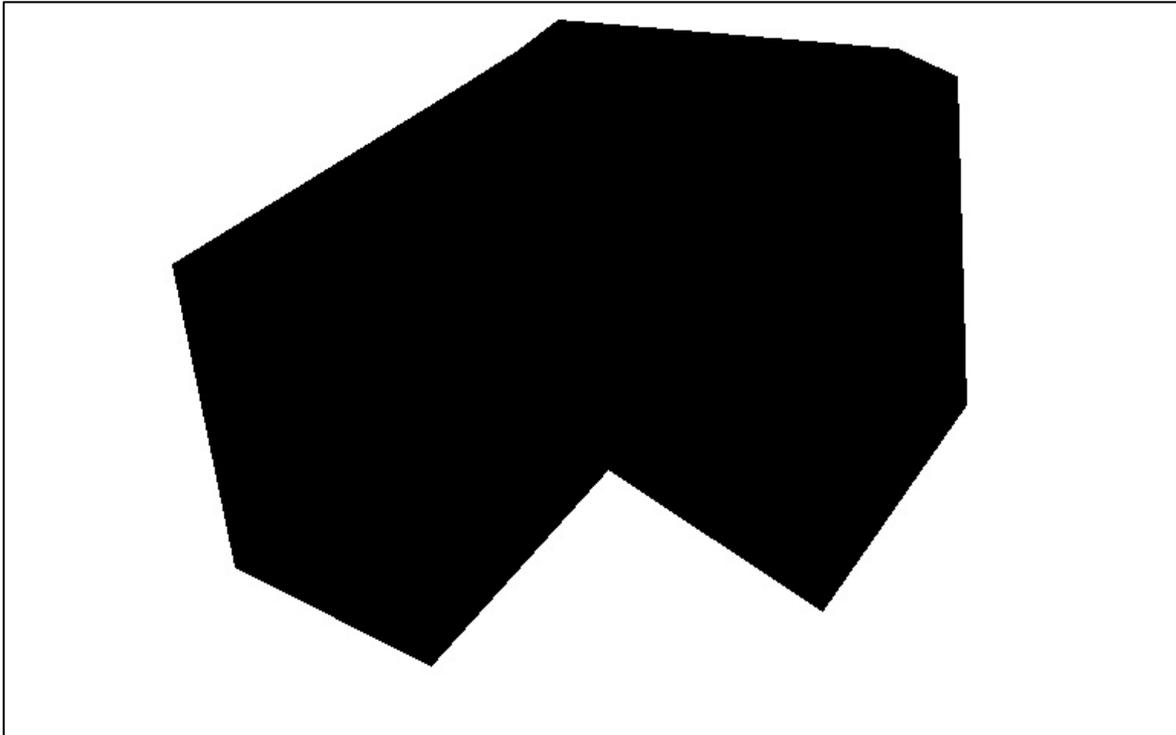
**Figura Anexo 19.** Séptima Imagen de prueba



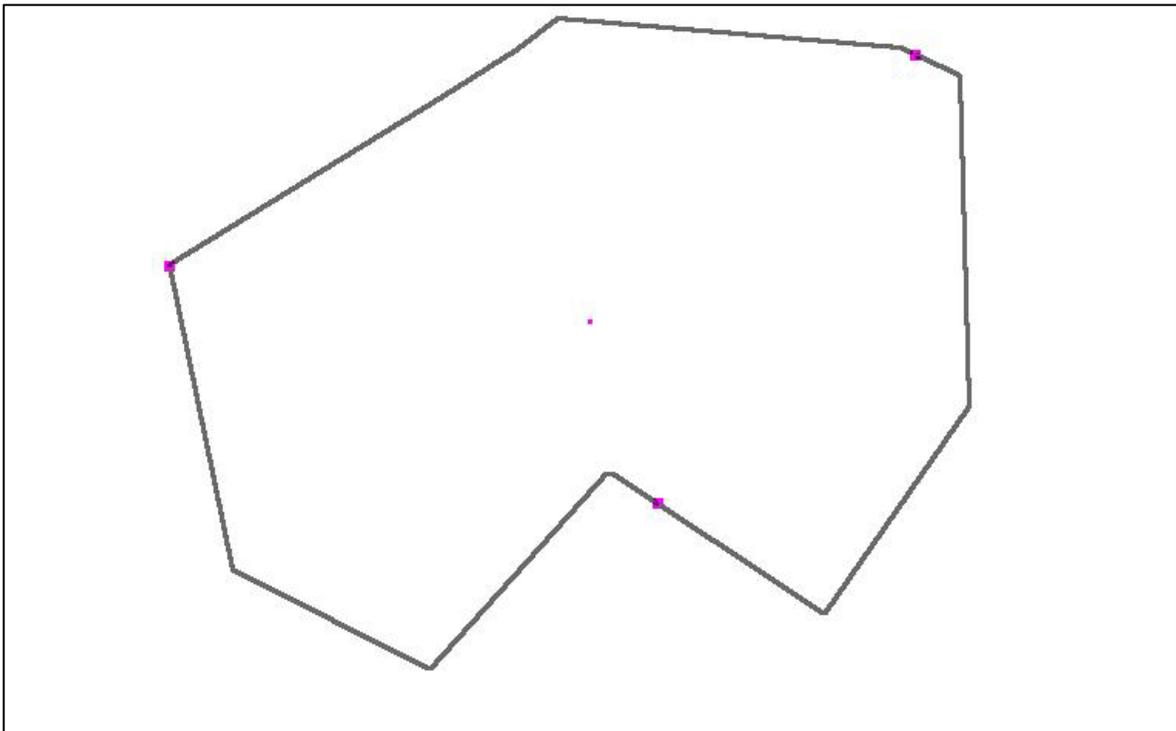
**Figura Anexo 20.** Resultado para algoritmo de detección de puntos de aprehensión estable para agarre con 3 dedos de Figura Anexo 19



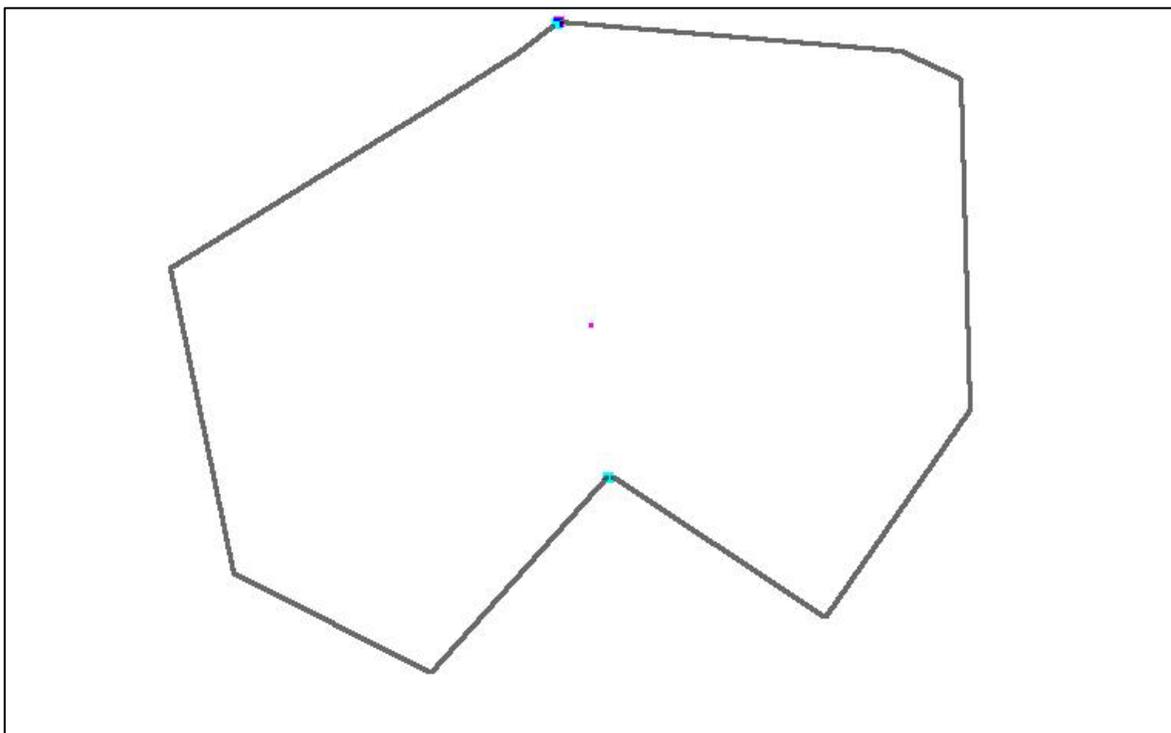
**Figura Anexo 21.** Resultado para algoritmo de detección de puntos de aprehensión estable para agarre con 2 dedos de Figura Anexo 19



**Figura Anexo 22.** Octava Imagen de prueba



**Figura Anexo 23.** Resultado para algoritmo de detección de puntos de aprehensión estable para agarre con 3 dedos de Figura Anexo 22



**Figura Anexo 24.** Resultado para algoritmo de detección de puntos de aprehensión estable para agarre con 2 dedos de Figura Anexo 22