



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Centro de Computación Gráfica

*Visión Artificial para
detección y ubicación espacial
de objetos en una escena*

Trabajo Especial de Grado
presentado ante la Ilustre
Universidad Central de Venezuela
por los Bachilleres
César Andrés Lizarraga Arreaza
Luis Ricardo Ortega Arias
Para optar al título de Licenciado en Computación

Tutores:
Omaira Rodríguez
Bricelis Urbina

Caracas, Octubre 2008

RESUMEN

El Centro de Computación Gráfica (CCG) en conjunto con el Instituto de Cirugía Experimental (ICE) de la Facultad de Medicina de la Universidad Central de Venezuela (UCV) y el Instituto Nacional de Bioingeniería (INABIO) están desarrollando un simulador virtual para entrenamiento en cirugía laparoscópica, en el marco del proyecto LAPAROS. Actualmente, el componente de hardware del simulador está conformado por un simulador mecánico y un subsistema de rastreo. El subsistema de rastreo está basado en un dispositivo comercial llamado ISOTRAK II, el cual envía al computador la posición de la pinza cuando esta se encuentra en la cavidad del simulador mecánico. Sin embargo, el CCG cuenta solamente con un equipo, el cual es de alto costo y presenta limitaciones en cuanto a la cantidad de rastreos simultáneos y la forma de interactuar con los instrumentos. Es por ello, que en este Trabajo Especial de Grado se presenta un sistema basado en Visión Artificial con un enfoque más general capaz de reconocer cualquier objeto en base a su color y ubicarlo en un ambiente tridimensional, presentando una manera alterna de interacción económica, portable e innovadora. El sistema propuesto utiliza un conjunto de cámaras digitales para capturar la imagen del objeto de interés. Estas imágenes son sometidas a un procesamiento que permite detectar y obtener las características relevantes del objeto de interés. Luego, estas características son la entrada de varios algoritmos que permiten calcular la posición del objeto y su ubicación en el espacio.

AGRADECIMIENTOS

A nuestros familiares, que siempre nos apoyaron en las buenas y en las malas.

A todo el personal del Centro de Computación Gráfica (CCG), por contar con sus ayudas, mejoras y buena disposición en todo momento.

Al Centro de Cálculo Científico y tecnológico (CCCT), por aclarar nuestras dudas y estar siempre dispuestos a ayudarnos cada vez que fuese necesario.

Al Centro de Geometría Gráfica Aplicada (CGGA), por ayudarnos a tener un entendimiento más profundo de los cálculos necesarios para el planteamiento de la geometría aplicada.

A los jurados por toda la disponibilidad y ayuda impartida.

Acta

Quienes suscriben, miembros del jurado por el Consejo de Escuela de Computación de la Facultad de Ciencias, para examinar el Trabajo Especial de Grado presentado por los Bachilleres César Lizarraga, CI. 16.379.295 y Luis Ortega, CI 16.368.742, con el título: “Visión Artificial para detección y ubicación espacial de objetos en una escena”, a los fines de optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído como fue, dicho trabajo por cada uno de los miembros del jurado, se fijó el día 28 de Octubre de 2008, a las 3:30pm para que sus autores lo defiendan en forma pública, lo que hizo en el Centro de Computación Gráfica, de la Escuela de Computación, mediante la presentación oral de su contenido, luego de la cual se dio respuesta a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

En fe de la cual se levanta la siguiente Acta en Caracas a los 28 días del mes de Octubre del año 2008 constancia de que actuó como Coordinador del Jurado la Profesora Tutora Omaira Rodríguez.

Prof. Jaime Parada
Jurado

Prof. Robinson Rivas
Jurado

Prof. Rhadamés Carmona
Jurado Suplente

Prof. Ernesto Coto
Jurado Suplente

Prof. Omaira Rodríguez
Tutora

Prof. Bricelis Urbina
Cotutora

TABLA DE CONTENIDO

ÍNDICE DE FIGURAS	3
ÍNDICE DE TABLAS	5
ÍNDICE DE GRÁFICOS	5
CAPÍTULO 1: INTRODUCCIÓN.....	6
1.1 PLANTEAMIENTO DEL PROBLEMA	7
1.2 OBJETIVO GENERAL	7
1.3 OBJETIVOS ESPECÍFICOS	7
CAPÍTULO 2: MARCO TEÓRICO.....	9
2.1 VISIÓN ARTIFICIAL	9
2.1.1 ETAPAS FUNDAMENTALES DE UN SISTEMA DE VISIÓN ARTIFICIAL.....	9
2.2.1 ADQUISICIÓN DE LA IMAGEN.....	12
2.3 PROCESAMIENTO DE LA IMAGEN	23
2.3.1 PREPROCESAMIENTO	24
2.3.2 EXTRACCIÓN DE CARACTERÍSTICAS.....	33
2.3.3 SEGMENTACIÓN.....	35
2.4 PROCESAMIENTO DE ALTO NIVEL	41
2.4.1 HEURÍSTICAS	41
2.4.2 ESTIMACIÓN DE COORDENADAS TRIDIMENSIONALES	41
CAPÍTULO 3: COMPONENTES DE HARDWARE DEL SISTEMA PROPUESTO	45
3.1 CÁMARA WEB.....	45
3.2 OBJETO DE INTERÉS.....	46
3.3 ILUMINACIÓN.....	46
3.4 AMBIENTES DEL SISTEMA.....	46
3.4.1 AMBIENTE CONTROLADO	46
3.4.2 SIMULAP V-1	47
CAPÍTULO 4: DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN	49
4.1 CASOS DE USO	49
4.1.1 NIVEL 0: DESCRIPCIÓN DE ACTORES Y SU INTERACCIÓN CON EL SISTEMA.....	49
4.1.2 NIVEL 1: CASOS DE USO PRINCIPALES	50
4.1.3 NIVEL 2: REFINAMIENTO DE LOS CASOS DE USO PRINCIPALES Y SUS RELACIONES....	51
4.1.4 NIVEL 3: REFINAMIENTO DE LOS CASOS DE USO SECUNDARIOS Y SUS RELACIONES..	55
4.2 DIAGRAMA DE CLASES	58
4.2.1 ESTRUCTURA DE LAS CLASES MÁS RELEVANTES DEL SISTEMA	59
4.3 PLATAFORMA DE DESARROLLO.....	61
4.3.1 LENGUAJE DE PROGRAMACIÓN C++	61
4.3.2 OPENCV	61
4.3.3 LAPACK.....	62
4.3.4 QT.....	62
4.3.5 OPENGL®	63

4.4 ENTORNO DE VISIÓN ARTIFICIAL.....	63
4.4.1 CONFIGURACIÓN DE LA APLICACIÓN.....	64
4.4.2 COMENZAR VISIÓN ARTIFICIAL.....	70
4.4.3 DESPLIEGUE TRIDIMENSIONAL.....	75
CAPÍTULO 5: PRUEBAS Y RESULTADOS.....	76
5.1 PRUEBAS REALIZADAS.....	76
5.1.1 PINZA LAPAROSCÓPICA.....	76
5.1.2 CUBO.....	76
5.2 RESULTADOS AMBIENTE CONTROLADO.....	77
5.3 RESULTADOS SIMULAP V-1.....	77
5.4 TIEMPO DE RESPUESTA DEL SISTEMA.....	78
5.4.1 TIEMPO DE REALIZACIÓN DEL PROCESO VISIÓN ARTIFICIAL.....	78
5.4.2 TIEMPO DE DESPLIEGUE TRIDIMENSIONAL.....	82
5.4.3 TIEMPO PROMEDIO DE RESPUESTA DEL SISTEMA.....	84
CONCLUSIONES.....	85
REFERENCIAS.....	87
ANEXO A: RECOMENDACIONES PARA LA CREACIÓN DE LAS IMÁGENES DE CALIBRACIÓN.....	89
ANEXO B: INSTRUCTIVO DE INTERFACES.....	92

ÍNDICE DE FIGURAS

<i>Figura 2.1: Esquema de un sistema de Visión Artificial.....</i>	<i>10</i>
<i>Figura 2.2: Escáner plano.....</i>	<i>13</i>
<i>Figura 2.3: Adaptador de escáner negativo 1.....</i>	<i>14</i>
<i>Figura 2.4.a: Adaptador de escáner negativo 2.....</i>	<i>14</i>
<i>Figura 2.4.b: Lámpara de escáner negativo.....</i>	<i>14</i>
<i>Figura 2.5: Escáner de película.....</i>	<i>15</i>
<i>Figura 2.6.a: Cámara digital 1.....</i>	<i>16</i>
<i>Figura 2.6.b: Cámara digital 2.....</i>	<i>16</i>
<i>Figura 2.7: Teléfono móvil con cámara.....</i>	<i>17</i>
<i>Figura 2.8.a: Cámara de video 1.....</i>	<i>17</i>
<i>Figura 2.8.b: Cámara de video 2.....</i>	<i>17</i>
<i>Figura 2.9.a: Webcam.....</i>	<i>18</i>
<i>Figura 2.9.b: Webcam portátil.....</i>	<i>18</i>
<i>Figura 2.10.a: Equipo de rayos X.....</i>	<i>19</i>
<i>Figura 2.10.b: Radiografía humana.....</i>	<i>19</i>
<i>Figura 2.10.c: Radiografía maletín.....</i>	<i>20</i>
<i>Figura 2.10.d: Radiografía.....</i>	<i>20</i>
<i>Figura 2.11.a: Equipo de resonancia magnética.....</i>	<i>20</i>
<i>Figura 2.11.b: Imagen de resonancia magnética.....</i>	<i>20</i>
<i>Figura 2.12.a: Equipo de tomografía axial computarizada (TAC).....</i>	<i>21</i>
<i>Figura 2.12.b: Tomografía pulmones.....</i>	<i>21</i>
<i>Figura 2.12.c: Tomografía de un virus.....</i>	<i>21</i>
<i>Figura 2.13.a: Equipo de Tomografías de emisión de positrones.....</i>	<i>22</i>
<i>Figura 2.13.b: Tomografía de Emisión de Positrones Original.....</i>	<i>22</i>
<i>Figura 2.13.c: Tomografía de Emisión de Positrones Resultado.....</i>	<i>22</i>
<i>Figura 2.14.a: Angiografía de sustracción digital.....</i>	<i>22</i>
<i>Figura 2.14.b: Resultado de una angiografía de sustracción digital.....</i>	<i>22</i>
<i>Figura 2.15.a: Ultrasonido.....</i>	<i>23</i>
<i>Figura 2.15.b: Ultrasonido cardiología.....</i>	<i>23</i>
<i>Figura 2.16.a: Máscara e imagen.....</i>	<i>28</i>
<i>Figura 2.16.b: Proceso de convolución.....</i>	<i>28</i>
<i>Figura 2.17.a: Imagen original (Promedio).....</i>	<i>29</i>
<i>Figura 2.17.b: Máscara de promedio.....</i>	<i>29</i>
<i>Figura 2.17.c: Filtro promedio.....</i>	<i>29</i>
<i>Figura 2.18.a: Imagen original (Gauss).....</i>	<i>29</i>
<i>Figura 2.18.b: Máscara de Gauss.....</i>	<i>29</i>
<i>Figura 2.18.c: Filtro Gaussiano.....</i>	<i>29</i>
<i>Figura 2.19.a: Imagen original (Sobel).....</i>	<i>30</i>
<i>Figura 2.19.b: Máscara de Sobel(Norte).....</i>	<i>30</i>
<i>Figura 2.19.c: Filtro Sobel (Norte).....</i>	<i>30</i>
<i>Figura 2.20.a: Imagen original (Prewitt).....</i>	<i>30</i>
<i>Figura 2.20.b: Máscara de Prewitt (horizontal).....</i>	<i>30</i>
<i>Figura 2.20.c: Filtro Prewitt.....</i>	<i>30</i>

<i>Figura 2.21.a: Imagen original (Roberts)</i>	31
<i>Figura 2.21.b: Máscara de Roberts (horizontal)</i>	31
<i>Figura 2.21.c: Filtro Roberts</i>	31
<i>Figura 2.22.a: Imagen original (Laplace)</i>	31
<i>Figura 2.22.b: Máscara de Laplace</i>	31
<i>Figura 2.22.c: Filtro Laplaciano</i>	31
<i>Figura 2.23: Histogramas</i>	32
<i>Figura 2.24: Ejemplo de ecualización</i>	32
<i>Figura 2.25: Ejemplos básicos de elementos estructurantes utilizados en la práctica</i>	36
<i>Figura 2.26: Dilatación de X por el elemento estructurante Y</i>	36
<i>Figura 2.27.a: Máscara de Dilatación Binaria</i>	37
<i>Figura 2.27.b: Imagen original (Dilatación)</i>	37
<i>Figura 2.27.c: Resultado Dilatación Binaria</i>	37
<i>Figura 2.28: Erosión de X por el elemento estructurante Y</i>	37
<i>Figura 2.29.a: Máscara de Erosión</i>	38
<i>Figura 2.29.b: Imagen original (Erosión)</i>	38
<i>Figura 2.29.c: Resultado Erosión</i>	38
<i>Figura 2.30.a: Imagen Original (Opening)</i>	38
<i>Figura 2.30.b: Resultado Opening</i>	38
<i>Figura 2.31.a: Imagen original (Closing)</i>	39
<i>Figura 2.31.b: Resultado Closing</i>	39
<i>Figura 2.32.a: Imagen original (Esqueletización)</i>	39
<i>Figura 2.32.b: Resultado Esqueletización</i>	39
<i>Figura 2.33: Geometría epipolar</i>	42
<i>Figura 3.34.a: Posición de cámaras</i>	45
<i>Figura 3.34.b: Logitech Quickcam Pro 9000®</i>	45
<i>Figura 3.35: Diversos objetos de interés</i>	46
<i>Figura 3.36: Lámpara casera</i>	46
<i>Figura 3.37.a: Ambiente controlado</i>	47
<i>Figura 3.37.b: Disposición de objetos en el ambiente controlado</i>	47
<i>Figura 3.38.a: SIMULAP V-1</i>	48
<i>Figura 3.38.b: Disposición de objetos en el SIMULAP V-1</i>	48
<i>Figura 4.39: Extracto de diagrama de clases</i>	58
<i>Figura 4.40: Esquema global del sistema</i>	64
<i>Figura 4.41: Módulo de configuración</i>	64
<i>Figura 4.42.a: Patrón de calibración</i>	66
<i>Figura 4.42.b: Resultado de la calibración</i>	66
<i>Figura 4.43: Estructura archivo configuración</i>	68
<i>Figura 4.44: Ejemplo de configuración</i>	69
<i>Figura 4.45: Módulo de Visión Artificial</i>	70
<i>Figura 4.46.a: Imagen de entrada</i>	71
<i>Figura 4.46.b: Resultado de preprocesamiento</i>	71
<i>Figura 4.47: Resultado de la segmentación</i>	72
<i>Figura 4.48: Ejemplo de varios objetos resultado de la extracción de características</i>	73
<i>Figura 4.49: Objeto de interés identificado</i>	74

<i>Figura 4.50: Escena tridimensional</i>	75
<i>Figura 5.51: Pinza laparoscópica con marca de color</i>	76
<i>Figura 5.52: Cubo</i>	77
<i>Figura 53: Patrón de calibración</i>	89
<i>Figura 54: Posibles ubicaciones del tablero (múltiples tomas)</i>	90
<i>Figura 55: Ubicaciones de la cámara</i>	90
<i>Figura 56: Angulo de inclinación (ejemplo de 45°)</i>	91
<i>Figura 57: Rotaciones de la cámara</i>	91
<i>Figura 58: Notación</i>	92
<i>Figura 59: Ventana de configuración</i>	92
<i>Figura 60: Ventana de selección de dispositivos</i>	92
<i>Figura 61: Ventana de calibración</i>	93
<i>Figura 62: Ventana de objeto de interés</i>	93
<i>Figura 63: Ventana consultar configuración actual</i>	94
<i>Figura 64: Ventana consultar configuración actual</i>	95

ÍNDICE DE TABLAS

<i>Tabla 4.1: Clase Calibracion</i>	59
<i>Tabla 4.2: Clase Camara</i>	59
<i>Tabla 4.3: Clase ModeloAplicacion</i>	60
<i>Tabla 5.4: Tiempo con pinza presente y ausente</i>	79
<i>Tabla 5.5: Tiempo promedio de pinza</i>	80
<i>Tabla 5.6: Tiempo con cubo presente y ausente</i>	81
<i>Tabla 5.7: Tiempos del cubo</i>	82
<i>Tabla 5.8: Tiempo de despliegue</i>	83
<i>Tabla 5.9: Tiempos de despliegue</i>	84
<i>Tabla 5.10: Tiempos de respuesta del sistema</i>	84

ÍNDICE DE GRÁFICOS

<i>Gráfico 5.1: Tiempo de Visión Artificial con pinza presente y ausente</i>	79
<i>Gráfico 5.2: Tiempo de Visión Artificial con cubo presente y ausente</i>	81
<i>Gráfico 5.3: Tiempo de despliegue</i>	83

CAPÍTULO 1: INTRODUCCIÓN

La Visión Artificial es una rama reciente de la computación que reúne áreas de investigación muy diversas, con la finalidad de crear sistemas artificiales que permitan simular procesos humanos basados en la visión, tales como: retener imágenes, reconocer patrones similares e identificar objetos; también busca mejorar procedimientos ya establecidos ofreciendo nuevas alternativas.

Problemas en el área de la medicina, como asistentes robóticos [2] [11], y otras áreas como detección de incendios [8], reconocimiento de gestos [7], entre otros, han sido solucionados con la técnica de Visión Artificial. En la actualidad es necesario utilizar elementos de hardware y software para ubicación espacial de objetos en una escena.

Con base en lo anterior, es posible crear un software de Visión Artificial capaz de detectar, seguir y representar en un ambiente tridimensional cualquier objeto basándose en sus colores característicos o marcas de colores. Este sistema será capaz de configurar dispositivos de captura, analizar el color característico del objeto a detectar y representar su ubicación en tres dimensiones.

La aplicación propuesta será capaz de acoplar y adaptar el sistema a cualquier situación que implique detectar la presencia de un objeto, estimar su posición y tomar una decisión en base a la respuesta obtenida.

Este Trabajo Especial de Grado presenta y desarrolla la técnica de Visión Artificial como una alternativa para el reemplazo de sistemas de rastreo. En los siguientes capítulos se describen las bases teóricas necesarias, los componentes de hardware utilizados en la creación del sistema, descripción del software de Visión Artificial creado, las pruebas realizadas y los resultados obtenidos. Finalmente se presentan las conclusiones.

1.1 PLANTEAMIENTO DEL PROBLEMA

En la actualidad existen componentes de hardware especializados en rastreo y ubicación espacial de objetos pero debido a su gran tamaño y altos costos es necesario buscar alternativas que permitan mejorar sistemas basados en estos componentes.

Específicamente el Centro de Computación Gráfica (CCG) en conjunto con el Instituto de Cirugía Experimental (ICE) de la Facultad de Medicina de la Universidad Central de Venezuela (UCV) y el Instituto Nacional de Bioingeniería (INABIO) están desarrollando un simulador virtual para entrenamiento en cirugía laparoscópica, en el marco del proyecto LAPAROS [25]. Este proyecto posee un subsistema de rastreo que está basado en un dispositivo comercial llamado ISOTRAK II, el cual devuelve al computador la posición de la pinza cuando ésta se encuentra en la cavidad del simulador mecánico. Sin embargo, el CCG cuenta solamente con un equipo, el cual es de alto costo y presenta limitaciones en cuanto a la cantidad de rastreos simultáneos y la forma de interactuar con los instrumentos.

Para solucionar estos inconvenientes, se plantea la siguiente pregunta: ¿Es posible el desarrollo de un Sistema de Visión Artificial que sea capaz de detectar y ubicar en un espacio tridimensional un objeto de interés con base en un color característico?

1.2 OBJETIVO GENERAL

El objetivo de este trabajo es diseñar, construir e implementar un Sistema de Visión Artificial que sea capaz de detectar un objeto de interés basándose en su marca de color y estimar su posición en un espacio tridimensional.

1.3 OBJETIVOS ESPECÍFICOS

- Estructurar y desarrollar el proceso de Visión Artificial que se adapte al problema.
- Determinar los dispositivos de captura adecuados basándose en su resolución, desempeño y tamaño, así como su ubicación dentro de la escena.
- Realizar la gestión y calibración de los dispositivos.
- Determinar el color característico del objeto de interés y configurarlo en el sistema.
- Diseño y construcción de un ambiente controlado para obtención de resultados óptimos del sistema.
- Determinar e implementar las técnicas apropiadas de procesamiento digital de imágenes asociadas a las etapas de preprocesamiento, segmentación y extracción de características.

- Determinar e implementar las técnicas asociadas al procesamiento de alto nivel.
- Diseño y despliegue de un ambiente tridimensional para la simulación de la escena y el objeto de interés.
- Evaluación de desempeño del sistema en varios ambientes y objetos de interés.

CAPÍTULO 2: MARCO TEÓRICO

Debido a la necesidad de identificar y sustentar las bases teóricas relacionadas al los objetivos propuestos, se debe reunir la información documental y confeccionar el diseño metodológico. A continuación, la información necesaria para la construcción e implementación de un sistema de Visión Artificial.

2.1 VISIÓN ARTIFICIAL

Según Nalwa [12], la Visión Artificial o comprensión de imágenes describe la deducción automática de la estructura y propiedades de un mundo tridimensional, posiblemente dinámico, bien a partir de una o varias imágenes bidimensionales de ese mundo.

En general, las técnicas de Visión Artificial son procedimientos que generan una respuesta basada en el análisis de información extraída de imágenes digitales.

Debido a que la Visión Artificial implica procesos bastante complejos, ésta se apoya en otros campos científicos con más experiencia, como lo son: la biología, la inteligencia artificial, la robótica, el aprendizaje de máquina, la computación gráfica, el procesamiento de señales, entre otras. Además, la investigación en esta área impulsa el desarrollo de otras áreas relacionadas, como los son: el procesamiento de imágenes, y la visión de máquina.

2.1.1 ETAPAS FUNDAMENTALES DE UN SISTEMA DE VISIÓN ARTIFICIAL

Las etapas de un sistema de Visión Artificial varían de manera significativa dependiendo de su propósito y de la calidad de las imágenes que se están tratando. Sin embargo, Olague [14] presenta un esquema de las etapas típicas encontradas en cualquier sistema de Visión Artificial:

- Adquisición de la imagen.
- Preprocesamiento.
- Extracción de características.
- Segmentación.
- Procesamiento de alto nivel.

La implementación de cada una de estas etapas requiere del conocimiento de otros campos relacionados. Para la adquisición de la imagen se debe conocer la forma como trabajan los dispositivos de captura de imágenes y sus capacidades. Para el preprocesamiento, la extracción de características, segmentación, el procesamiento digital de imágenes toma un papel fundamental, así como también el campo de la inteligencia artificial o los distintos algoritmos, modelos matemáticos y/o geométricos que permiten resolver problemas en base a la información obtenida.

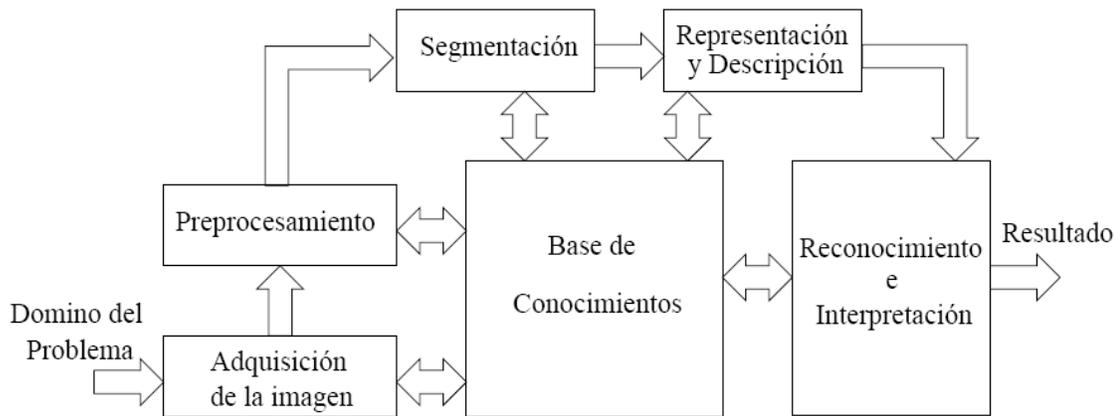


Figura 2.1: Esquema de un sistema de Visión Artificial

Adquisición

Generalmente, el procesamiento de la imagen se realiza digitalmente, por lo tanto es necesario disponer de un dispositivo que permita capturar las imágenes y representarlas internamente en el computador. Para esto se utilizan dispositivos como cámaras digitales, satélites, escáner, equipos médicos, entre otros.

Preprocesamiento

Como su nombre lo indica, consiste en realizar un tratamiento previo a las imágenes. Esto se debe a la necesidad de corregir los errores que introducen los dispositivos de adquisición de imágenes ya sea directa o indirectamente y para facilitar el procesamiento en las etapas posteriores.

Extracción de características

En esta etapa del proceso se desea encontrar en la imagen información relevante que la identifique. Con ayuda del procesamiento de una imagen se pueden localizar tendencias y rasgos que caractericen a la misma, y dependiendo del objetivo final existe la posibilidad de identificar patrones de interés.

Segmentación

En esta etapa se toma la información obtenida anteriormente y se utiliza como criterio de comparación y clasificación, esto con la finalidad de detectar información relevante en la imagen. Es importante resaltar que dependiendo del caso, algunas características aportan más que otras al momento de tomar las decisiones, algunas pueden hasta no aportar nada. Una vez depurada esta información se procede a dividirla en grupos disjuntos (un píxel no puede pertenecer a dos grupos al mismo tiempo), con el propósito de separar los objetos de interés de la imagen. Esta etapa generará información que será utilizada luego para la toma de decisiones así como también se pueden disparar acciones predefinidas.

Procesamiento de alto nivel

El procesamiento de alto nivel es la etapa final del proceso, una vez que la información necesaria fue extraída, filtrada, caracterizada y clasificada se procede a hallar una respuesta. El procesamiento de alto nivel debe estar provisto de un mecanismo que permita analizar la información obtenida para generar los resultados correspondientes o realizar las acciones respectivas.

En esta etapa suelen encontrarse técnicas relacionadas con la inteligencia artificial. Esto se debe a que el sistema debe ser capaz de reconocer, aprender, comparar, identificar y tomar decisiones según su experiencia para alcanzar un entendimiento avanzado del entorno. Sin embargo, en muchos casos también se usan modelos simplemente algorítmicos, matemáticos y/o geométricos para resolver el problema en cuestión. Estos casi siempre necesitan un preprocesamiento inicial, tal como una calibración, para determinar datos iniciales, o distancias iniciales, ángulos, entre otros.

2.2.1 ADQUISICIÓN DE LA IMAGEN

En la Visión Artificial el procesamiento de una imagen se realiza digitalmente, por lo tanto es necesario que un dispositivo de captura sea capaz de proveer, obtener las imágenes y representarlas internamente en el computador. A continuación los dispositivos de captura más comunes.

Dispositivos de captura

Un dispositivo de captura es un hardware que permite tomar una información y representarla en forma digital para que luego pueda ser procesada por el computador. También se les conoce comúnmente como digitalizadores debido a que ellos transforman una información determinada (luz, imagen, sonidos, entre otros) en información representable por el computador, es decir en información digital (números binarios).

El interés y enfoque principal son los dispositivos de captura de imágenes que están compuestos por sensores ópticos, los cuales capturan la intensidad de la luz que reciben y la transforman en números binarios.

Los sensores mencionados anteriormente, por naturaleza sólo captan la intensidad de la luz. Para lograr capturar imágenes a color, es necesario utilizar sensores de color. Un sensor de color se compone de tres sensores idénticos con un filtro cada uno, de modo que sólo sea alcanzado por la intensidad de una determinada componente de la luz. Los filtros empleados para dichos sensores son el verde, el rojo y el azul, de modo que un sensor a color entrega tres valores, uno proveniente de cada sensor.

Una vez obtenido un valor por cada canal, se necesita hacer una correspondencia entre el color obtenido y el color equivalente a representar en el monitor, ya que el monitor puede usar alguna codificación distinta a la obtenida con el dispositivo.

Clasificación de dispositivos de captura

Strizinec [23], propone un esquema para clasificar los tipos de dispositivos de captura según su funcionamiento.

Escáneres planos

El escáner es el dispositivo de captura de imágenes más tradicional (Figura 2). Su función principal es capturar una fotografía o texto de un documento, imagen de revista o libro y luego transformarla a una información que pueda ser interpretada y manejada por el computador.



Figura 2.2: Escáner plano

El escáner está compuesto por: un cristal (sobre el cual se apoya la imagen que se desea capturar para su posterior digitalización), una lámpara y un sensor. Su funcionamiento empieza cuando la lámpara ilumina el área objetivo para que el sensor pueda captar la luz y convertirla en impulsos eléctricos. Luego dichos impulsos (de diferentes intensidades) son convertidos a valores numéricos que son transmitidos al computador.

Cada sensor es capaz de discernir entre 256 niveles, por lo que devuelve un número de 8 bits ($2^8 = 256$). Juntando la información de los 3 sensores, se obtienen los 24 bits que indican el color de un píxel. Ahora ($2^{24} = 16777216$) será la cantidad de colores que se puede representar. Actualmente han surgido escáneres que emplean 16 bits para cuantificar los niveles de cada sensor, por lo que cada píxel va codificado en 48 bits (16 bits x 3 sensores).

Adaptadores para escanear negativos y diapositivas



Figura 2.3: Adaptador de escáner negativo 1

Algunos escáneres planos incorporan adaptadores (Figura 3) que posibilitan escanear negativos y diapositivas. El adaptador consiste simplemente en otra lámpara idéntica a la que hay dentro del escáner, pero que puede ser colocada sobre la diapositiva o el negativo, de modo que sea iluminada desde atrás para que la luz atraviese y así llegue al sensor, en lugar de rebotar en una fotografía. Cuando se emplea el adaptador para negativos y diapositivas, no se enciende la lámpara normal del escáner, sino la del adaptador (Figura 4.a, 4.b).



Figura 2.4.a: Adaptador de escáner negativo 2

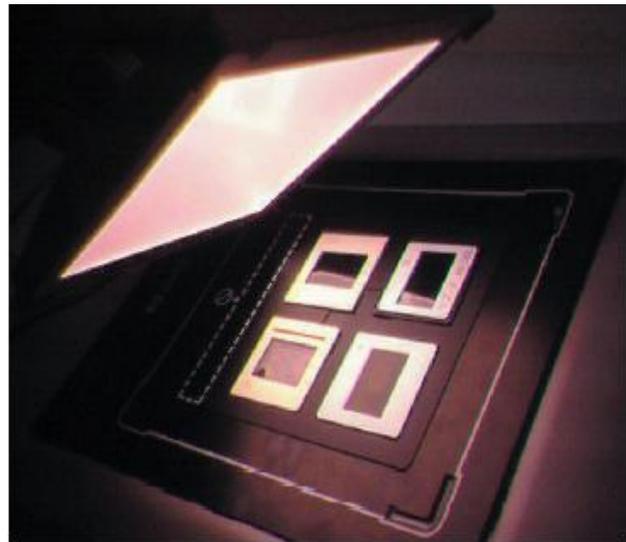


Figura 2.4.b: Lámpara de escáner negativo

Escáner de texto

Además de escanear fotografías también se puede escanear documentos que contengan caracteres, con lo cual se puede obtener un archivo de texto que podrá ser modificado por programas de procesamiento de palabras, tales como Microsoft Office ® [10] u OpenOffice [15]. Sin embargo, esta adquisición en realidad no depende del dispositivo sino más bien de un programa. Un ejemplo de dicho programa es el denominado Reconocimiento Óptico de Caracteres (OCR por sus siglas en inglés), el cual está basado en una técnica que utiliza el escáner como medio para obtener la imagen de alto contraste del texto (en blanco y negro) y luego se realiza el procesamiento a dicha imagen obteniendo el texto correspondiente, ya sea en formato pdf o en algún editor de texto.

Escáneres de película (Negativos y Diapositivas)

Este tipo de escáner tiene una resolución de trabajo bastante alta hasta 4000 puntos por pulgada (DPI por sus siglas en inglés), pero posee una limitante que sólo puede trabajar con negativos y diapositivas, no es posible capturar fotografías en papel (Figura 5).



Figura 2.5: Escáner de película

Adicionalmente el hardware especializado es capaz de eliminar al momento del escaneo los rayones, raspaduras y polvo que se acumulan sobre las diapositivas y los negativos. Esto se logra iluminando la muestra desde diferentes ángulos, obteniendo de esta manera varios resultados por cada píxel. Luego analizando y comparando las imágenes obtenidas y la información supuestamente real, con todo esto se evita muchísimo trabajo de retoque, que podría llegar a complicarse.

También es posible obtener tonos reales, es decir obtener colores negros verdaderos en vez de negros con impurezas. Esto se logra escaneando varias veces el mismo negativo o diapositiva (de 4 a 16 veces, según la calidad que se desea obtener).

Por último, presenta un sistema de carga motorizado, lo que permite el procesado en lote de varias diapositivas en menos tiempo, o directamente tiras de películas de muchos fotogramas o hasta carretes completos sin cortar.

Cámaras digitales

Estos dispositivos son muy utilizados debido a sus costos accesibles, versatilidad y la alta calidad con la que capturan la “imagen real” sin tener que pasar por películas, procesos químicos, y escaneados, los cuales a su vez introducen errores e impurezas.



Figura 2.6.a: Cámara digital 1



Figura 2.6.b: Cámara digital 2

Una cámara digital posee la virtud de captar directamente “el objeto o paisaje original” por lo que se obtienen los mejores resultados al no existir procesos intermedios. La luz proveniente del objeto o paisaje original va directamente al sensor, pasando únicamente por las lentes del objetivo. Una vez capturada por el sensor, la imagen es representada en valores binarios. El siguiente paso para visualizarse en el monitor es una simple transmisión de datos que digitalmente está garantizada y verificada, es decir no existen errores, en caso de existir alguno es corregido automáticamente.

El principio de funcionamiento de una cámara digital es idéntico al de una cámara de película; consta de un objetivo compuesto por varias lentes que se encarga de dirigir la luz proveniente del objeto o paisaje a una zona en la que se pone un elemento de captura. En las cámaras tradicionales, en esa zona de captura hay una película sensible a la luz (dentro de un recinto oscuro), mientras que en las digitales hay un sensor electrónico. El sensor electrónico transforma la luz que le llega en números, que son almacenados en una memoria, y que permanecen allí hasta que el usuario decide que hacer con ellos.

Existen dispositivos capaces de proveer funcionalidades extras sobre las imágenes justo después de la captura como por ejemplo visualización, eliminación, o transferirlas a una impresora o computador para su posterior procesamiento. Por ello, para lograr imágenes digitales de buena calidad, las primeras etapas son muy importantes, es aquí donde los distintos componentes pueden afectar la calidad de la imagen obtenida, como son los lentes, y el sensor. El resto de los componentes harán que se pueda manipular en el sitio con mayor o menor facilidad las fotos obtenidas, cambiar fotos, grabar video, ver o imprimir las fotografías sin la necesidad de recurrir a un computador.

Teléfonos móviles con cámara

Actualmente, es muy común un teléfono móvil que incorpore una cámara digital (figura 7). Por lo general, la calidad de estas cámaras no es muy elevada, y están limitadas principalmente en la resolución y tipo de ópticas que emplean, las cuales suelen ser de plástico o vidrio, en vez de cristal. La falta de resolución también se sustenta en que la finalidad es transmitir la fotografía obtenida a través de una llamada telefónica. Al adquirir un teléfono móvil con cámara digital hay que tener en cuenta que no se está comprando una cámara digital de calidad, sino que será un teléfono móvil con prestaciones adicionales.



Figura 2.7: Teléfono móvil con cámara

Cámara de video

Además de las cámaras de fotografía digitales, existen cámaras de video digitales (figura 8) que permiten la captura de horas de movimiento de forma digital que luego pueden transmitir a un computador para su posterior utilización en este medio (las más modernas graban directamente en los DVD de 8cm de diámetro).



Figura 2.8.a: Cámara de video 1



Figura 2.8.b: Cámara de video 2

Como la tecnología de una cámara de video digital es similar a la de una cámara de fotos digital, las cámaras de fotos digitales incorporan la posibilidad de grabar video. Las limitaciones que existen cuando se captura un video por las cámaras de fotos digitales son la capacidad de almacenamiento reducida que tienen, o la velocidad del sensor para transmitir información captada al medio de almacenamiento. Existen cámaras digitales que limitan el tiempo al espacio de almacenamiento disponible, y otras que dan límites, por ejemplo: no más de tres minutos consecutivos, a pesar de que hay espacio de almacenamiento de más de 30 minutos.

En el caso de los videos, también la resolución de captura es importante, ya que a resoluciones más elevadas, mayor información será almacenada, y en consecuencia se obtiene menos tiempo de filmación, así como también se ve afectada cualquier aplicación en tiempo real.

Cámaras web o webcam

Se denominan webcams a dispositivos de captura digital de video de baja calidad, suelen estar destinados a las videoconferencias a través de Internet. Al momento de una video conferencia, es necesario transmitir video en tiempo real. En general, las imágenes deben ser de baja resolución, por lo que las webcams en la mayoría de los casos no ofrecen mucha calidad. Por ello son pequeñas y económicas. En la actualidad, con el avance de la tecnología ahora existen webcam de alta calidad.

Además de la representación de videoconferencia para la cual fueron diseñadas, muchas ofrecen la posibilidad de tomar fotografías digitales, e incluso otras permiten colocarles baterías y ser portátiles (figura 9).



Figura 2.9.a: Webcam



Figura 2.9.b: Webcam portátil

Equipos médicos

Otros dispositivos disponibles para la captura son los utilizados en el área médica. La captura en estos dispositivos viene dada por diversas técnicas que permiten la visualización de partes internas del organismo. Otra motivación que presentan estos dispositivos es que no sólo se limitan a la medicina sino que su aplicación puede ser extendidas a otras disciplinas como lo son: estudio del suelo, rocas, minerales, actividades sísmicas y propósitos de seguridad. A continuación, se explicarán brevemente varios dispositivos usados comúnmente [1] [21]:

Rayos X

Este dispositivo mediante la emisión de rayos X que atraviesan volúmenes o sólidos, genera una imagen bidimensional en función de la densidad del material (conocida como radiografía). Existe una variedad de usos de los rayos X: En la industria se utilizan para detectar problemas estructurales y grietas en metales que no pueden ser identificadas a simple vista. Comercialmente se utilizan en aviones y puentes para asegurar que no existan fracturas o grietas en los materiales. En el área de seguridad específicamente en el chequeo de equipaje y cargamentos se utilizan para identificar armas, bombas y materiales peligrosos. El uso más común es en medicina, se utilizan para examinar dentro del cuerpo en busca de anomalías, huesos rotos, crecimientos cancerígenos, entre otros.



Figura 2.10.a: Equipo de rayos X



Figura 2.10.b: Radiografía humana



Figura 2.10.c: Radiografía maletín



Figura 2.10.d: Radiografía

Resonancia Magnética

También conocida como resonancia magnética nuclear, mide la respuesta de los protones (partículas cargadas positivamente que se encuentran en el núcleo de todos los átomos) a un campo magnético. Crea imágenes blanco y negro o en color que reflejan la química de los tejidos. Esta técnica tiene una limitante, no puede ser empleada en personas con marcapasos o prótesis metálicas. Es muy útil para visualizar alteraciones en los tejidos, placas de ateroma, lesiones cerebrales, tumores entre otros.



Figura 2.11.a: Equipo de resonancia magnética

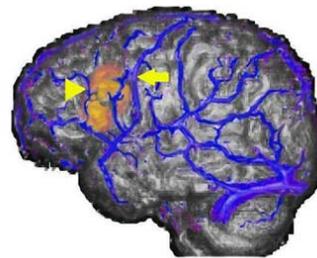


Figura 2.11.b: Imagen de resonancia magnética

Tomografía

La tomografía computarizada también conocida como tomografía axial computarizada (TAC) consiste en un as de rayos X que se desplaza alrededor del objeto, escaneado desde cientos de ángulos diferentes. El computador toma toda esta información, la agrupa y ensambla un modelo 3-D del objeto en cuestión. Esta técnica es muy útil en el área médica para la localización de tumores, inflamaciones, infecciones, cálculos renales y biliares, lesiones en los tejidos y malformaciones. Con esta técnica es posible capturar objetos del mundo real y representarlos dentro del computador en tres dimensiones.



Figura 2.12.a: Equipo de tomografía axial computarizada (TAC)



Figura 2.12.b: Tomografía pulmones

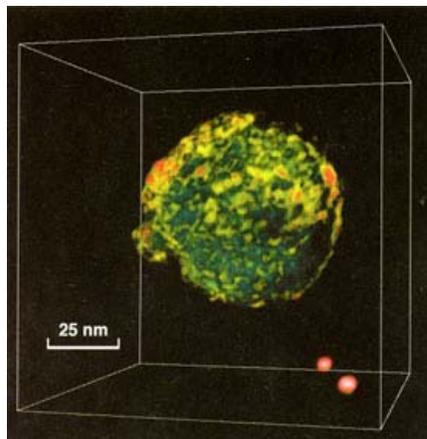


Figura 2.12.c: Tomografía de un virus

Tomografía de Emisión de Positrones

La tomografía de emisión de positrones consiste en inyectar a un cuerpo una sustancia radioactiva la cual emite partículas cargadas positivamente (positrones). Al chocar estas con los electrones negativos de los átomos de los tejidos, genera una radiación gamma que se visualiza en un monitor de video. La tomografía de emisión de positrones se utiliza para estudiar la función de algún órgano o los cambios metabólicos en algún tejido.



Figura 2.13.a: Equipo de Tomografías de emisión de positrones

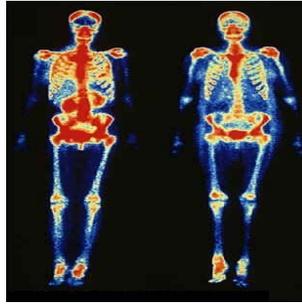


Figura 2.13.b: Tomografía de Emisión de Positrones Original

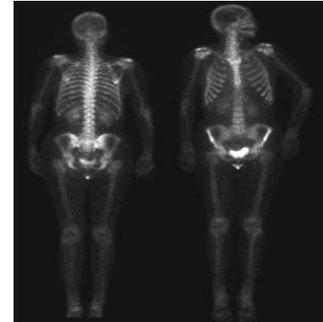


Figura 2.13.c: Tomografía de Emisión de Positrones Resultado

Angiografía de sustracción digital

Esta técnica consiste en utilizar un computador que se encarga de comparar una imagen de rayos X de una parte del cuerpo antes y después de inyectar un medio de contraste en un vaso sanguíneo. La sustracción en la primera imagen de los tejidos visualizados en la segunda imagen, permiten revelar donde se encuentra la obstrucción del vaso.



Figura 2.14.a: Angiografía de sustracción digital

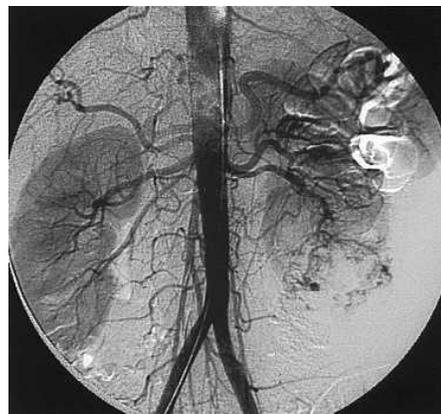


Figura 2.14.b: Resultado de una angiografía de sustracción digital

Ultrasonidos

Esta técnica consiste en generar ondas sonoras de alta frecuencia mediante un dispositivo manual, las ondas reflejadas por las diversas estructuras del cuerpo son recogidas por el dispositivo. Estas señales son enviadas a un monitor formando una imagen (monograma) que puede ser estática o en movimiento.



Figura 2.15.a: Ultrasonido



Figura 2.15.b: Ultrasonido cardiología

2.3 PROCESAMIENTO DE LA IMAGEN

Una vez obtenida la imagen a través de los dispositivos de captura es necesario realizar diversos tratamientos para mejorar la calidad, extraer características y/o segmentar cierta información. Esto con el fin de simplificar la complejidad de la imagen y centrarse en los factores relevantes e importantes. Todos estos tratamientos se realizan de manera digital.

El procesamiento digital de imágenes está dado por un conjunto de operaciones llevadas a cabo sobre las imágenes a fin de realizar mediciones cuantitativas que permitan describirlas; es decir, extraer ciertas características que permitan mejorar, perfeccionar o detallar la imagen [18].

Esta área está íntimamente relacionada con la Visión Artificial ya que contribuye con distintas técnicas a las etapas de preprocesamiento, extracción de características, segmentación de imágenes. Los resultados de la aplicación de estas técnicas será la información fundamental para tomar una decisión en un sistema de Visión Artificial.

Las distintas técnicas de procesamiento se pueden clasificar, debido a su efecto y propósito sobre la imagen, en tres categorías [5]:

- Preprocesamiento
- Extracción de características
- Segmentación

2.3.1 PREPROCESAMIENTO

La motivación de realizar un tratamiento previo a las imágenes se debe a la necesidad de corregir los errores producidos por los dispositivos de adquisición de imágenes o las condiciones del ambiente al momento de la captura.

En los siguientes párrafos se explicarán las distintas operaciones llevadas a cabo en la etapa de preprocesamiento utilizando la siguiente convención.

$IMG(x, y)$ = función que retorna el píxel (x, y) de la imagen original.

Operaciones Aritméticas

Consiste en tomar una imagen y aplicarle operaciones matemáticas fundamentales, tales como: suma, resta, multiplicación o división con respecto a una constante o a otra imagen.

- Una imagen y una constante:

$$\text{Suma: } SUM(x,y) = IMG1(x,y) + C$$

$$\text{Resta: } RES(x,y) = IMG1(x,y) - C$$

$$\text{Multiplicación: } MUL(x,y) = IMG1(x,y) * C$$

$$\text{División: } DIV(x,y) = IMG1(x,y) / C; \text{ con } C \neq 0$$

- Dos imágenes:

$$\text{Suma: } SUM(x,y) = IMG1(x,y) + IMG2(x,y)$$

$$\text{Resta: } RES(x,y) = IMG1(x,y) - IMG2(x,y)$$

$$\text{Multiplicación: } MUL(x,y) = IMG1(x,y) * IMG2(x,y)$$

$$\text{División: } DIV(x,y) = IMG1(x,y) / IMG2(x,y); \text{ con } IMG2(x,y) \neq 0$$

Operaciones Lógicas

Consiste en tomar una imagen original y aplicarle operaciones a nivel de bits (por ejemplo: AND, OR, XOR, etc), con respecto una constante lógica o a otra imagen.

- Una imagen:

$$\text{AND: } \mathbf{AND}(x,y) = \mathbf{IMG1}(x,y) \ \& \ C$$

$$\text{OR: } \mathbf{OR}(x,y) = \mathbf{IMG1}(x,y) \ / \ C$$

$$\text{XOR: } \mathbf{XOR}(x,y) = \mathbf{IMG1}(x,y) \ \wedge \ C$$

$$\text{NEG: } \mathbf{NEG}(x,y) = \mathbf{NOT}(\mathbf{IMG}(x,y))$$

- Dos imágenes:

$$\text{AND: } \mathbf{AND}(x,y) = \mathbf{IMG1}(x,y) \ \& \ \mathbf{IMG2}(x,y)$$

$$\text{OR: } \mathbf{OR}(x,y) = \mathbf{IMG1}(x,y) \ / \ \mathbf{IMG2}(x,y)$$

$$\text{XOR: } \mathbf{XOR}(x,y) = \mathbf{IMG1}(x,y) \ \wedge \ \mathbf{IMG2}(x,y)$$

Umbralización o thresholding

Consiste en eliminar (o cambiar a otro número) los valores superiores o inferiores a una variable que se llamará valor de umbralización.

Ejemplo: valor de Umbralización = 20

Si $(\mathbf{IMG}(x,y) < \text{valor de Umbralización})$ **Entonces** $\mathbf{UMB}(x,y) = 0;$

Sino $\mathbf{UMB}(x,y) = \text{valor nuevo};$

FSi

Binarización

Consiste en transformar una imagen a dos valores de intensidad distintos, es decir 0 ó 1. Es un caso particular de la umbralización en donde se colocan todos los píxeles que superen el valor del umbral en 1 y los que no en 0.

Escala de Grises

Consiste en tomar una imagen y transformarla en otra que está formada por varios tonos de blanco y negro.

Contraste

Consiste en modificar las diferencias de luminosidad y densidad entre las sombras y las luces altas en una imagen.

Brillo

Consiste en modificar la luminosidad u oscuridad presente en una imagen.

Traslación o Desplazamiento

Consiste en mover una imagen con respecto a un cierto origen, se toma la posición de cada píxel (x, y) de la imagen y se le adiciona la cantidad T_x a la primera coordenada y la cantidad T_y a la segunda coordenada, donde (T_x, T_y) representa el desplazamiento.

$$IMG(x,y) = IMG(x + T_x, y + T_y)$$

Rotación

Consiste en girar una imagen con respecto a un origen y en un ángulo β . Se toma cada píxel (x, y) de la imagen y se rota en un ángulo determinado β con respecto a un centro c_x, c_y .

Escalado o Zoom

Consiste en reducir o aumentar el tamaño de la imagen, dado un factor de escalamiento S_x para la primera coordenada y S_y para la segunda. Es decir:

$$IMG(x,y) = IMG(x * S_x, y * S_y)$$

Espejo

La técnica de espejo consiste en invertir las posiciones de cada píxel (x,y) con respecto a un eje determinado. Este eje puede ser horizontal, vertical o diagonal. Suponiendo que la imagen es de tamaño MxN.

Espejo Vertical:

$$\mathbf{IMG}(x,y) = \mathbf{IMG}(N-x, y)$$

Espejo Horizontal:

$$\mathbf{IMG}(x,y) = \mathbf{IMG}(x, M - y)$$

Espejo Diagonal:

$$\mathbf{IMG}(x,y) = \mathbf{IMG}(C-x, M - y)$$

Filtrado

El filtrado es una técnica que se utiliza para lograr distintos efectos visuales dependiendo de la necesidad que se presente. La convolución es una técnica de filtrado que consiste en la construcción y aplicación de un operador a la imagen, lo cual producirá una modificación en cada píxel de acuerdo a la información de sus vecinos. Este operador es llamado de convolución y está formado por matrices rectangulares de diferentes tamaños, en la práctica comúnmente se utilizan matrices cuadradas de tamaños 3x3, 5x5, 7x7.

Ejemplo:

Se tiene una máscara W de dimensión 3x3, una imagen F(6x6) y una imagen resultado G

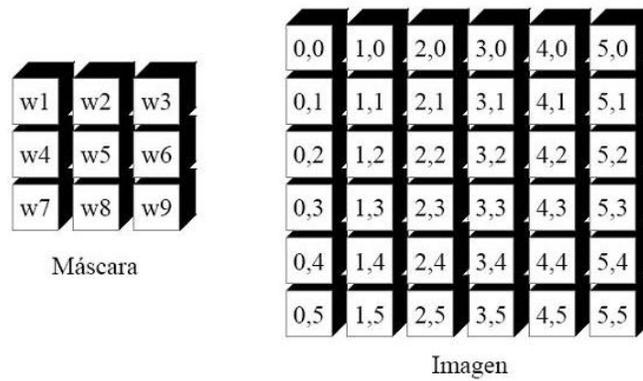


Figura 2.16.a: Máscara e imagen

Convolución

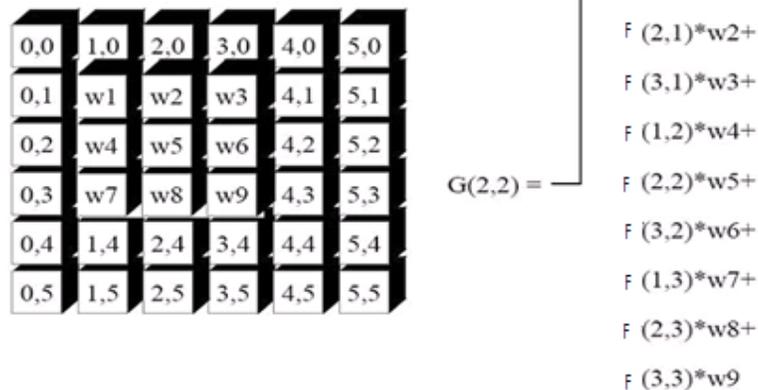


Figura 2.16.b: Proceso de convolución

Filtros de promedio

La técnica de promedio consiste en buscar el término medio del valor del píxel con respecto a su vecindad en el núcleo de convolución. El valor obtenido será el nuevo valor del píxel. Estos filtros se usan generalmente para eliminar el ruido, mientras más grande sea la máscara más píxeles serán tomados en cuenta y más ponderado será el resultado, pero mas información podría perderse. El efecto visual es el siguiente:

Definimos el símbolo (\oplus) para denotar el proceso de convolución

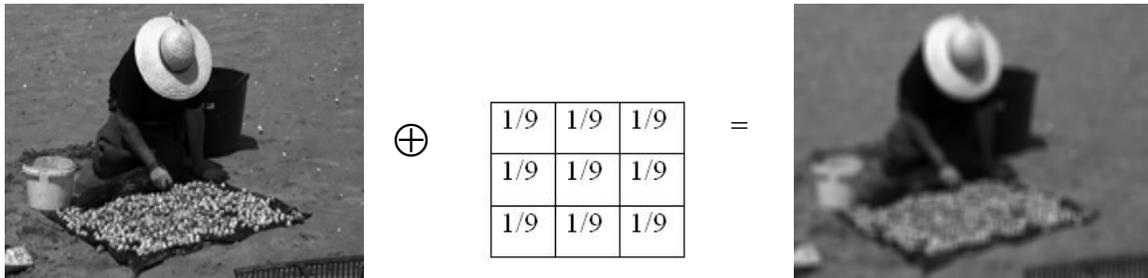


Figura 2.17.a: Imagen original (Promedio)

Figura 2.17.b: Máscara de promedio

Figura 2.17.c: Filtro promedio

Filtro Gaussiano

La idea de esta técnica es hacer el efecto de “Campana de gauss” a la imagen, lo cual significa dar más importancia al píxel central e ir disminuyendo la importancia a medida de que los valores estén más lejos del píxel original. El filtro gaussiano es usado comúnmente con el propósito de suavizado, remueve detalles y ruidos que puedan estar asociados a la imagen.

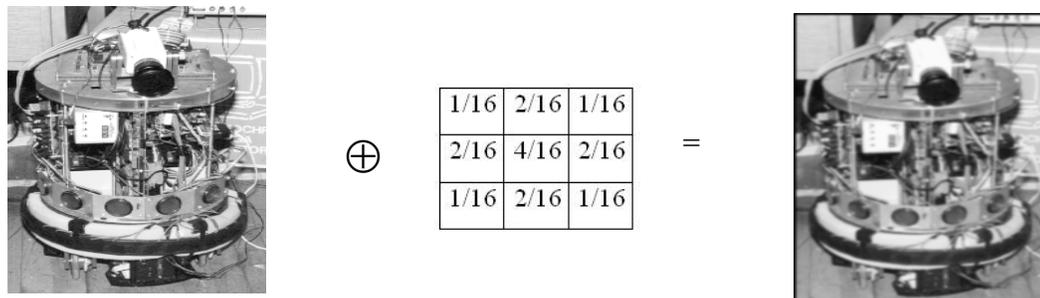


Figura 2.18.a: Imagen original (Gauss)

Figura 2.18.b: Máscara de Gauss

Figura 2.18.c: Filtro Gaussiano

Existen filtros capaces de detectar y resaltar los píxeles que presenten cambios de intensidades bruscos en una imagen. Esto se logra mediante la aplicación de operadores de convolución. Una característica es que son unidireccionales, aunque se pueden combinar varios para obtener diferentes resultados, este proceso es llamado convolución múltiple.

Filtro de Sobel

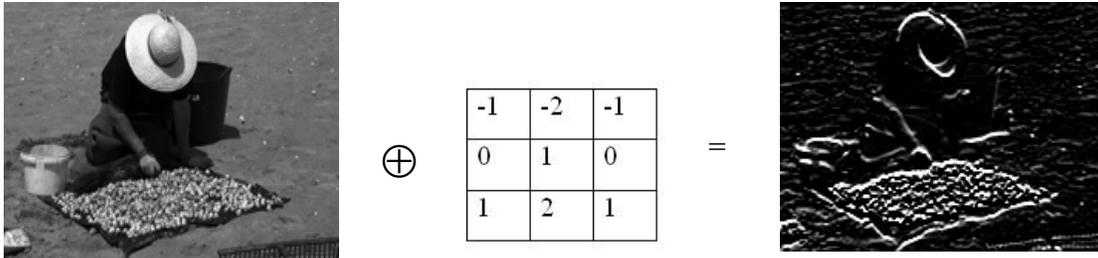


Figura 2.19.a: Imagen original (Sobel)

Figura 2.19.b: Máscara de Sobel(Norte)

Figura 2.19.c: Filtro Sobel (Norte)

Los filtros de prewitt y roberts son particulares ya que se aplican mínimo dos veces a la imagen original, una vez con la máscara horizontal y una vez con la máscara vertical, luego ambos resultados son sumados para obtener bordes bien definidos, este proceso se puede repetir para obtener mayor definición de los bordes.

Filtro de Prewitt

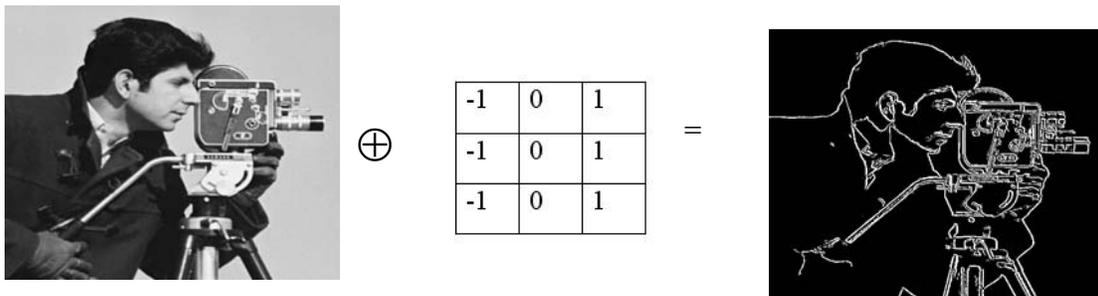


Figura 2.20.a: Imagen original (Prewitt)

Figura 2.20.b: Máscara de Prewitt (horizontal)

Figura 2.20.c: Filtro Prewitt

Operador de Roberts



\oplus

0	0	0
0	1	0
0	0	-1

=



Figura 2.21.a: Imagen original (Roberts)

Figura 2.21.b: Máscara de Roberts (horizontal)

Figura 2.21.c: Filtro Roberts

Los filtros de Prewitt y Sobel localizan bien los cambios de tono por fila y por columna, mientras que el operador de Roberts en este caso localiza de forma diagonal los bordes pero es más sensible al ruido. En el resultado visual no existe mucha diferencia entre las distintas técnicas.

Filtro Laplaciano

Esta técnica está basada en el cálculo de la velocidad de cambio de intensidades en la imagen. Se crea una máscara que destaque los píxeles (a través del aumento de sus niveles de intensidad) cuya variación, con respecto a su vecindad, sea significativa. Esto permite extraer los bordes de los objetos presentes. Su principal característica es ser no direccional e invariante a rotaciones. Es decir, el resultado no se ve afectado por la dirección del operador o la ubicación de la imagen.



\oplus

0	1	0
1	-4	1
0	1	0

=



Figura 2.22.a: Imagen original (Laplace)

Figura 2.22.b: Máscara de Laplace

Figura 2.22.c: Filtro Laplaciano

Histograma

El histograma de una imagen es una función discreta que describe de manera global la cantidad de intensidades que contiene una imagen. Un histograma generalmente se realiza a partir de una imagen en escala de grises, sin embargo, se puede hacer de una imagen en formato RGB creando un histograma por cada componente de color.

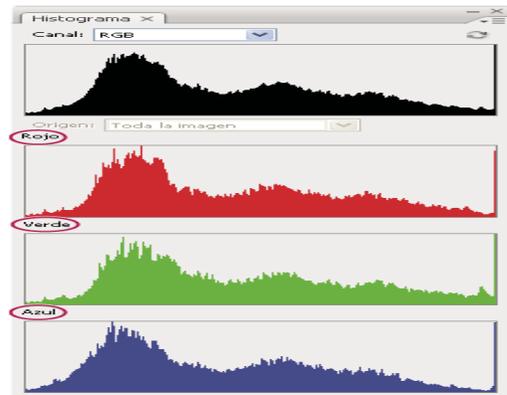


Figura 2.23: Histogramas

Ecualización

Se utiliza para mejorar el contraste de la imagen. Esta operación redistribuye los niveles de intensidades de una forma más equitativa, es decir que si existen picos en el histograma la ecualización redistribuirá estos picos para hacer un histograma resultante más suave.

Este procedimiento cuenta con ciertas características:

- Una mayor utilización de los recursos disponibles: al ecualizar el histograma, se puede ver como los tonos que antes estaban más agrupados, ahora se han separado, ocupando todo el rango de grises, por lo que la imagen se está enriqueciendo al tener niveles de gris distintos entre sí, mejorando, por tanto, la apariencia visual de la imagen.
- Un aumento del contraste: esta ventaja es consecuencia del punto anterior, ya que si el histograma de la imagen ocupa todo el rango de grises, se aumenta la distancia entre los diferentes tonos de la imagen, y en consecuencia, se aumenta el contraste.

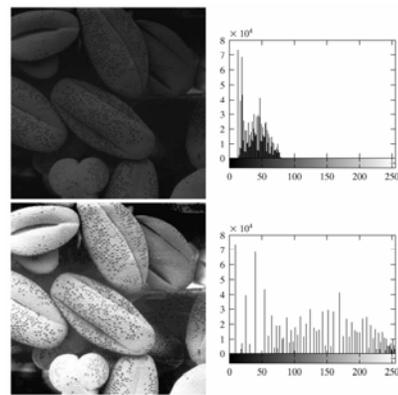


Figura 2.24: Ejemplo de ecualización

- Pérdida de información: puede ocurrir que a algunos píxeles que en la imagen original tenían distintos niveles de gris se les asigne, tras la ecualización global, el mismo nivel de gris. Por otro lado, hay casos en los que dos niveles de gris muy próximos se separen, dejando huecos en el histograma [5].

2.3.2 EXTRACCIÓN DE CARACTERÍSTICAS

Las características son atributos que identifican y diferencian elementos. Estos atributos se utilizan para tomar decisiones respecto a los objetos en la imagen. Existen atributos naturales como artificiales. Los naturales se definen mediante la apariencia visual de la imagen, mientras que los artificiales son el resultado de operaciones realizadas a la imagen.

Realizar mediciones sobre las imágenes generalmente requiere que las características estén bien definidas, los bordes bien delimitados y, el color y el brillo sean uniformes. El tipo de mediciones a realizar para cada característica específica es un factor importante para poder determinar los pasos apropiados para su procesamiento. Los procedimientos aplicados para el procesamiento de imágenes están orientados a las aplicaciones, ya que lo que puede ser adecuado para una aplicación puede no serlo para otra [20].

Las características a ser detectadas deben al menos cumplir con los siguientes requisitos básicos para obtener los mejores resultados:

- Discriminantes: que diferencien lo mejor posible los objetos de un grupo de otro.
- Independientes: los descriptores que definan a cada objeto no tiene que estar relacionados, de tal manera, si se cambia un descriptor los demás no sean afectados por éste.
- Suficientes: tienen que delimitar de forma suficiente la pertenencia de un objeto a una clase delimitada.

La búsqueda de características bien definidas en una imagen no es una tarea sencilla. Es por esto que surgen ciertos algoritmos capaces de facilitar las búsquedas en base a una cualidad en común. Búsqueda en profundidad (BFS) es un algoritmo de exploración de grafos muy usado por su sencillez y eficacia. El algoritmo consiste en visitar una zona del grafo; si cumple con características dadas se marca como visitado, y luego se repite el proceso con todos los vértices adyacentes recursivamente. El resultado final serían todas las zonas con una característica dada marcada de una forma específica. Este método es adecuado para agrupar zonas con características similares, lo cual es ideal para la etapa de extracción de características [22].

Existen varias características que pueden ser extraídas sobre regiones particulares de la imagen, como [9]:

Líneas

Una de las características principales que se puede extraer son las líneas. Esto se logra con la transformada de Hough [3]. La transformada de Hough es un algoritmo empleado en reconocimiento de patrones en imágenes que permite encontrar ciertas formas dentro de una imagen. La versión más simple consiste en encontrar líneas. Su modo de operación es principalmente estadístico y consiste en que, para cada punto que se desea averiguar si es parte de una línea se aplica una operación dentro de cierto rango, con lo que se averiguan las posibles líneas de las que puede ser parte el punto. Esto se continúa para todos los puntos en la imagen. Al final se determina qué líneas fueron las que más puntos posibles tuvieron y esas son las líneas en la imagen. El problema de este algoritmo es que el cálculo de la transformada es muy lento, debido a que tiene que desplazarse por todos los contornos encontrando las líneas ajustadas a estos.

Color

La percepción del humano del color en un objeto cambia dependiendo de la forma en que la luz incida. Debido a esto al momento de utilizar un color como característica primordial es necesario evaluar las condiciones de luz presentes. El color en una imagen es una característica muy importante. Un objeto de interés de un color conocido en una imagen puede ser extraído fácilmente. Computacionalmente el color puede ser representado con múltiples modelos como: RGB (rojo, verde y azul), CMY (cyan, magenta y amarillo), HSL (tono, saturación e iluminación) entre otros.

Iluminación

Existen varios tipos de luces que pueden afectar el color de un objeto [28].

- Ambiental: es la luz que esta esparcida en todo el entorno e ilumina a todos los objetos por igual, es imposible determinar su origen y cuando rebota en un objeto se esparce igualmente en todas direcciones.
- Difusa: es la luz que viene de una dirección particular y es reflejada equitativamente sobre una misma superficie, la luz del sol es un ejemplo de luz difusa.
- Especular: es aquella que tiene una dirección pero a diferencia de la luz difusa se refleja en una dirección particular creando ese efecto de brillo en metales y plásticos brillantes.

Centro del objeto

El centro de gravedad o centroide es la media de las coordenadas de los puntos del contorno.

Caja delimitadora del contorno

La región puede ser delimitada por un cuadrilátero.

Redondez

Indica la similitud de un objeto a una circunferencia perfecta. La redondez R estará entre los valores 0 y 1, donde 1 representa un círculo (perfecto) y 0 para una región que tenga altura y/o ancho igual a cero. La redondez viene dada por la siguiente fórmula:

$$R = \frac{4.A.\pi}{L^2}$$

Perímetro del objeto

Es el número de píxeles asociados al borde del objeto.

Área del objeto

Es el número de píxeles que contiene el objeto.

2.3.3 SEGMENTACIÓN

Proceso mediante el cual una imagen es dividida en grupos disjuntos (un píxel no puede pertenecer a dos grupos al mismo tiempo) con el propósito de separar las partes de interés de la imagen del resto, usualmente en esta etapa se identifica la existencia del objeto. A continuación diferentes técnicas de segmentación:

Transformaciones morfológicas

Según Ortiz [17], el objetivo de las transformaciones morfológicas es la extracción de estructuras geométricas en los conjuntos sobre los que se opera, mediante la utilización de otro conjunto de forma conocida denominado elemento estructurante. El tamaño y la forma de este elemento se escogen a priori, de acuerdo a la morfología del conjunto con que se va a interactuar y a la extracción de formas que se desean obtener.



Figura 2.25: Ejemplos básicos de elementos estructurantes utilizados en la práctica

Dilatación

Esta técnica consiste en tomar el elemento estructurante Y , desplazarlo por la imagen X revisando si Y contiene algún elemento del conjunto X obteniendo de esta manera un conjunto resultante que representa la dilatación de la imagen. Esta técnica es utilizada para aumentar el contorno, definición de los objetos y unir líneas discontinuas. Se puede extender esta definición a imágenes binarias o escala de grises.

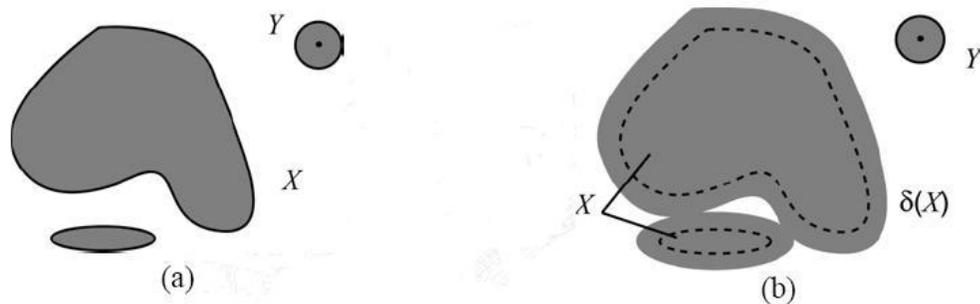


Figura 2.26: Dilatación de X por el elemento estructurante Y

En la figura 26 se puede notar que el conjunto X aumenta su definición, en la figura 27 se presenta el mismo proceso de dilatación pero de manera numérica, obteniendo un resultado similar.

0	1	0
1	1	1
0	1	0

Figura 2.27.a: Máscara de Dilatación Binaria

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Figura 2.27.b: Imagen original (Dilatación)

0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

Figura 2.27.c: Resultado Dilatación Binaria

Erosión

Esta técnica es la función dual de la dilatación, pero no inversa. Consiste en comprobar si el elemento estructurante Y está totalmente incluido dentro del conjunto X . Cuando esto no ocurre, el resultado de la erosión es el conjunto vacío. A diferencia de la dilatación que expande los bordes y contornos del objeto, la erosión reduce los contornos de los objetos y es utilizado para separar objetos que están unidos por una pequeña parte de sus contornos. Se puede extender esta definición a imágenes binarias o escala de grises.

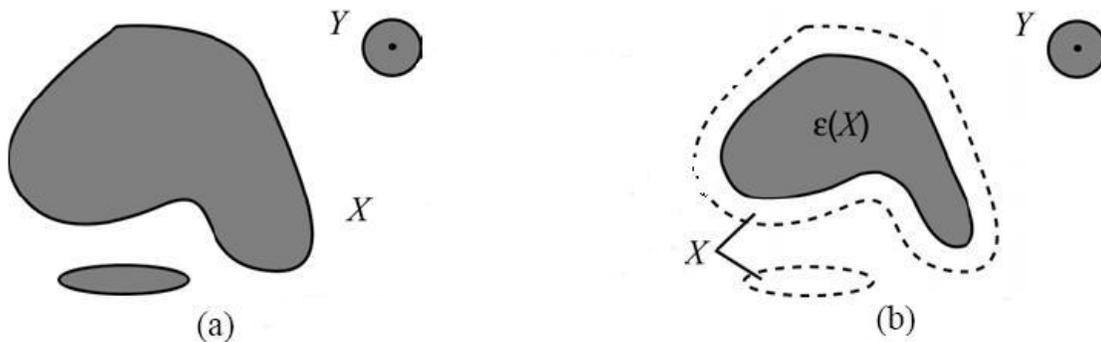


Figura 2.28: Erosión de X por el elemento estructurante Y

En la figura 28 se puede notar que los elementos conectados del conjunto X mas pequeños que Y son eliminados. En la figura 29 es el mismo proceso de erosión pero de manera numérica.

0	1	0
1	1	1
0	1	0

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0

Figura 2.29.a: Máscara de Erosión

Figura 2.29.b: Imagen original (Erosión)

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	0	1	1	1	0	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Figura 2.29.c: Resultado Erosión

Opening o Apertura

Esta técnica consiste en aplicarle a una imagen una erosión seguida de una dilatación. Esta técnica se utiliza para separar unas formas de otras, descomposición de objetos en elementos más simples, extracción de formas determinadas en un entorno con ruido, eliminar salientes estrechos y separar objetos que no están demasiado unidos.

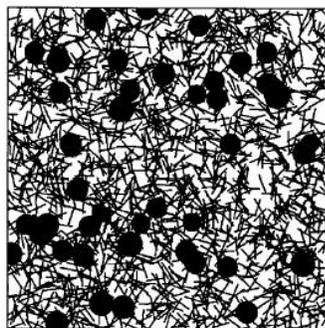


Figura 2.30.a: Imagen Original (Opening)

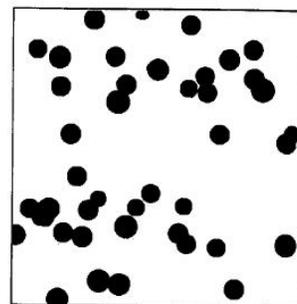


Figura 2.30.b: Resultado Opening

Cierre o Closing

Técnica que consiste en aplicar a una imagen una dilatación seguida de una erosión. Esta técnica se utiliza para ayudar a eliminar estructuras oscuras menores en tamaño al elemento estructurante. La dilatación maximiza los valores de que se atenúan los objetos oscuros (figura 31.a). La erosión minimiza la señal y solo los elementos no eliminados quedan presentes en la imagen final (figura 31.b).

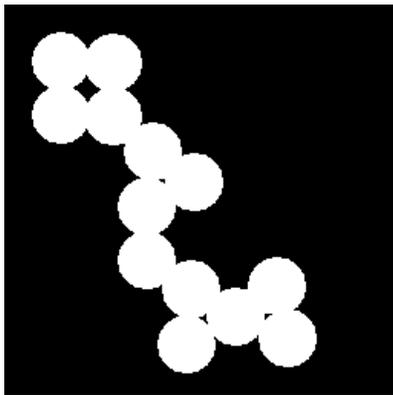


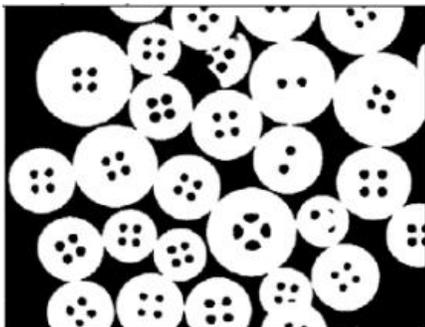
Figura 2.31.a: Imagen original (Closing)



Figura 2.31.b: Resultado Closing

Esqueletización

Técnica mediante la cual se toman los objetos de una imagen y se representan con un esqueleto de grosor de un píxel. En otras palabras, se toma un objeto cualquiera y se va reduciendo su grosor hasta que sea equivalente a un píxel. El esqueleto conseguido se puede utilizar como una idea para la forma inicial del objeto.



**Figura 2.32.a: Imagen original
(Esqueletización)**

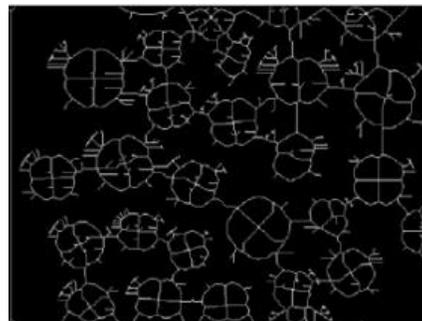


Figura 2.32.b: Resultado Esqueletización

Segmentación basada en umbralización

Es el método más básico para diferenciar un objeto del fondo, el cual se basa en la técnica explicada anteriormente de umbralización. De esta manera se puede segmentar la imagen en dos partes. La idea principal de esta técnica es encontrar el valor umbral más adecuado para realizar una binarización o umbralización y separar el objeto del fondo. Conseguir este valor umbral es el principal problema de esta técnica, ya que puede resultar sencillo como podría resultar complicado, dependiendo si las condiciones de iluminación y del fondo son homogéneas, constantes o variantes dependiendo de la situación.

Segmentación orientada a regiones

Si se conoce con anterioridad la estructura de la zona a segmentar, se crea un patrón de la estructura de la región a buscar basada en distintas imágenes de ésta, este patrón será comparado con una imagen, determinando si la zona a segmentar ha sido encontrada. La segmentación orientada a regiones intenta descubrir regiones que tengan características en común dentro de la imagen, esto se puede lograr ya sea agrupando píxeles en zonas o eligiendo regiones arbitrariamente y comparando con el patrón de la estructura a buscar.

Segmentación orientada a bordes

La idea de esta técnica proviene en gran parte de las técnicas de detección de bordes, y se trata del cálculo de un operador local de derivación basado en que un píxel pertenece a un borde si se produce un cambio brusco entre sus niveles de intensidad con sus vecinos. Mientras más brusco sea el cambio, más fácil es detectar el borde. Un problema a tener en cuenta es que en la búsqueda de los cambios bruscos para detectar los bordes, también se detectará, de manera indeseada, el ruido. Como se ha visto anteriormente, existen varios operadores que se utilizan para la extracción de bordes, pero estos operadores no son exactos y son afectados por el ruido, por lo tanto es necesario realizar otros pasos posteriores.

Las fronteras son bordes unidos que caracterizan la forma de un objeto. Por lo tanto, son útiles para calcular rasgos geométricos como tamaño u orientación. Existen varios métodos para determinar si un borde es frontera como la relajación de bordes (la cual se encarga de resaltar éstos y determinar si son fronteras), que lo logra tomando en cuenta los píxeles cercanos al borde, así como también se puede hacer un seguimiento de contorno que no es más que ubicar un borde e identificar su forma para determinar si engloba algún rasgo geométrico.

2.4 PROCESAMIENTO DE ALTO NIVEL

En esta última fase del proceso se toman todos los datos obtenidos de las fases anteriores. Luego la misma es interpretada, procesada y se generan los resultados. El procesamiento de alto nivel es el encargado de producir el resultado final, ya sea detectar patrones, tomar decisiones, realizar acciones entre otros.

Las técnicas aplicadas en esta etapa suelen estar basadas en Inteligencia Artificial. Sin embargo, en muchos casos también se usan heurísticas, modelos algorítmicos, matemáticos y/o geométricos para resolver el problema en cuestión.

Enfocados en el problema que se intenta resolver, a continuación se explicarán ciertas heurísticas utilizadas y una técnica geométrica que permite hallar la posición de un objeto en el espacio tridimensional en base a dos imágenes de dicho objeto, cada una tomada desde diferentes puntos de vista.

2.4.1 HEURÍSTICAS

Una heurística es un conjunto de criterios que permiten obtener una solución satisfactoria a un problema en específico, aunque no necesariamente la solución óptima [19]. Al momento de detectar la presencia de un objeto existen diversas maneras de eliminar falsos positivos y generar un resultado. Características como tamaño, forma, posición y color entre otros son importantes para tomar una decisión. Dependiendo del caso alguna información será más relevante que otra, lo importante es desarrollar criterios que sean capaces de tomar en cuenta las características más relevantes que identifiquen inequívocamente al objeto de interés. La intención general de este método es ganar rapidez y desempeño perdiendo precisión.

2.4.2 ESTIMACIÓN DE COORDENADAS TRIDIMENSIONALES

Existen diversas técnicas para la estimación de coordenadas tridimensionales en base a dos vistas diferentes de un punto en específico, una de ellas es la basada en la geometría epipolar, a continuación la explicación de esta geometría así como mecanismos capaces de estimar una posición en el espacio tridimensional.

Hartley y Zisserman [6] indican un método para estimar la posición en tres dimensiones de un objeto desde dos vistas diferentes, formando una geometría a partir de estas, la cual es llamada epipolar (figura 33).

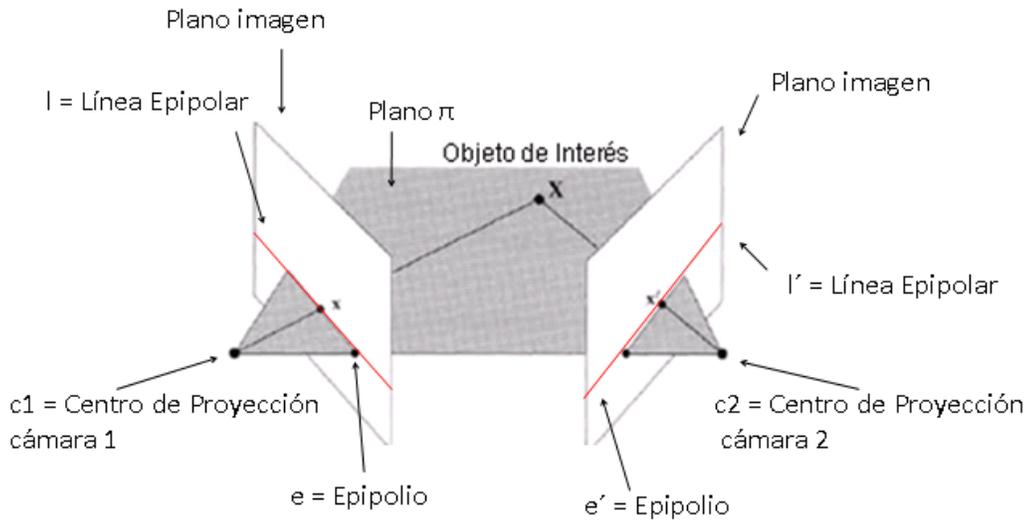


Figura 2.33: Geometría epipolar

Donde x y x' son un objeto en cada imagen respectivamente, X es la posición del objeto en tres dimensiones, π es el plano que intercepta al objeto en cada imagen y X , lo que indica que x , x' y X son coplanares. e es el punto de intersección entre el plano de la imagen y la línea que une las cámaras, l y l' es la intersección entre el plano de la imagen A y B respectivamente con π . $C1$ y $C2$ son los respectivos centros de proyección de cada cámara.

Triangulación lineal

Es un método para estimar la posición de un objeto en un espacio tridimensional a partir de dos imágenes, el cual está basado en la geometría explicada anteriormente. La idea principal es estimar un punto en tres dimensiones X el cual es visualizado por dos cámaras. Este punto debe satisfacer $x = P.X$, $x' = P'.X$, (donde P y P' son las matrices de proyección de cada una de las cámaras y x y x' el punto X en ambas imágenes de cada cámara). Sin embargo el método no cumple exactamente con las relaciones geométricas ya que al momento del cálculo de los puntos x y x' pueden existir imprecisiones, esto implica que no existirá un punto X tal que satisfaga exactamente $x = P.X$, $x' = P'.X$. Resultando en una aproximación que difiere ligeramente con respecto al resultado teórico esperado.

El método propuesto asume que las matrices de proyección de los dispositivos son provistas así como también los puntos de interés en cada imagen. Denotemos por P_1 la matriz de proyección de la primera cámara, P_2 la matriz de la segunda cámara, X el objeto en tres

dimensiones, x el objeto en la primera imagen, x' el objeto en la segunda imagen, α_i la distancia focal en x y β_i la distancia focal en y del dispositivo i .

$$P_1 = \begin{bmatrix} \alpha_1 & 0 & 0 \\ 0 & \beta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; P_2 = \begin{bmatrix} \alpha_2 & 0 & 0 \\ 0 & \beta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$X = \begin{bmatrix} a \\ b \\ c \end{bmatrix}; x = \begin{bmatrix} x_1 \\ y_1 \\ w_1 \end{bmatrix}; x' = \begin{bmatrix} x_1' \\ y_1' \\ w_1' \end{bmatrix}; x, x' \text{ expresados en coordenadas homogéneas.}$$

La relación entre un punto X en el espacio tridimensional y su proyección P se describe por las ecuaciones usuales de proyección perspectiva. Así, un punto x en una imagen de la primera vista se puede expresar como $x = P_1.X$ y homológamente $x' = P_2.X$ para la segunda vista.

$$P_1.X = \begin{bmatrix} \alpha_1 & 0 & 0 \\ 0 & \beta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \alpha_1.a \\ \beta_1.b \\ c \end{bmatrix}$$

$$P_2.X = \begin{bmatrix} \alpha_2 & 0 & 0 \\ 0 & \beta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \alpha_2.a \\ \beta_2.b \\ c \end{bmatrix}$$

Partiendo de las premisas $x = P_1.X$, $x' = P_2.X$ se procede a aplicar un producto cruz a la derecha en ambas ecuaciones donde se obtiene $x \times (P_1.X) = (P_1.X) \times (P_1.X)$, $x' \times (P_2.X) = (P_2.X) \times (P_2.X)$.

$$x \times (P_1.X) = \begin{bmatrix} x_1 \\ y_1 \\ w_1 \end{bmatrix} \times \begin{bmatrix} \alpha_1.a \\ \beta_1.b \\ c \end{bmatrix} = \begin{bmatrix} \beta_1.b.w_1 - y_1.c \\ \alpha_1.a.w_1 - x_1.c \\ \alpha_1.a.y_1 - \beta_1.b.x_1 \end{bmatrix} = \begin{bmatrix} 0.a & \beta_1.b.w_1 & -y_1.c \\ \alpha_1.a.w_1 & 0.b & -x_1.c \\ \alpha_1.a.y_1 & -\beta_1.x_1.b & 0.c \end{bmatrix}$$

$$x' \times (P_2.X) = \begin{bmatrix} x_1' \\ y_1' \\ w_1' \end{bmatrix} \times \begin{bmatrix} \alpha_2.a \\ \beta_2.b \\ c \end{bmatrix} = \begin{bmatrix} \beta_2.b.w_1' - y_1'.c \\ \alpha_2.a.w_1' - x_1'.c \\ \alpha_2.a.y_1' - \beta_2.b.x_1' \end{bmatrix} = \begin{bmatrix} 0.a & \beta_2.b.w_1' & -y_1'.c \\ \alpha_2.a.w_1' & 0.b & -x_1'.c \\ \alpha_2.a.y_1' & -\beta_2.x_1'.b & 0.c \end{bmatrix}$$

Las matrices asociadas a cada uno de estos sistemas tienen rango 2 y adicionalmente como el valor de w_1 y w_1' es 1, el sistema que relaciona las dos vistas con el objeto de interés lo podemos expresar como:

$$A.X = 0 \iff \begin{bmatrix} 0 & \beta_1 & -y_1 \\ \alpha_1 & 0 & -x_1 \\ 0 & \beta_2 & -y_1' \\ \alpha_1 & 0 & -x_1' \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

El sistema $A.X = 0$ planteado anteriormente tiene infinitas soluciones. La solución trivial $X = 0$ no es de interés para el problema en cuestión. A continuación una solución vía mínimos cuadrados que aproxima la solución al resultado real.

El sistema se resuelve encontrando el X que minimice $\|AX\|$ sujeto a la condición $\|X\|=1$. Sea $A=UDV^T$ (usando la descomposición en valor singular SVD por sus siglas en inglés) [4], es necesario minimizar $\|UDV^T X\|$. Sin embargo, $\|UDV^T X\|=\|DV^T X\|$ y $\|X\|=\|V^T X\|$. Por consiguiente, es necesario minimizar $\|DV^T X\|$ sujeto a la condición $\|V^T X\|=1$. Sustituyendo $y = V^T X$ y luego minimizando $\|Dy\|$ sujeto a $\|y\|=1$. Ahora, D es una matriz diagonal con sus entradas en orden descendente. Se deduce que la solución a este problema es $y = [0 \ 0 \ 1]^T$ teniendo una entrada distinta de cero, específicamente uno (1) en la última posición. Por último la solución de $X = Vy$ es el vector que contenga el mínimo valor singular, el cual es la última columna de V .

CAPÍTULO 3: COMPONENTES DE HARDWARE DEL SISTEMA PROPUESTO

El sistema de Visión Artificial propuesto consta de componentes físicos como dispositivos de captura, iluminación, objetos de interés y un ambiente especializado para su funcionamiento. A continuación se presentarán los componentes físicos utilizados en el sistema.

3.1 CÁMARA WEB

Es un componente de hardware esencial para el funcionamiento del sistema. Actualmente consta de dos (2) cámaras Web que deben estar colocadas de forma cruzada tal que su rango de visión abarque completamente el objeto de interés, es decir, las proyecciones de las cámaras no pueden ser paralelas (como se muestra en la figura 34). Aunque el sistema es capaz de reconocer cualquier cámara Web que sea manejada por la librería OpenCV (comentada más adelante) actualmente cuenta con dos (2) cámaras Logitech Quickcam Pro 9000®.

Requerimientos mínimos:

- Resolución de imagen (320x240).
- Tasa de refrescamiento 25 cuadros por segundo.
- Ajuste de foco manual.

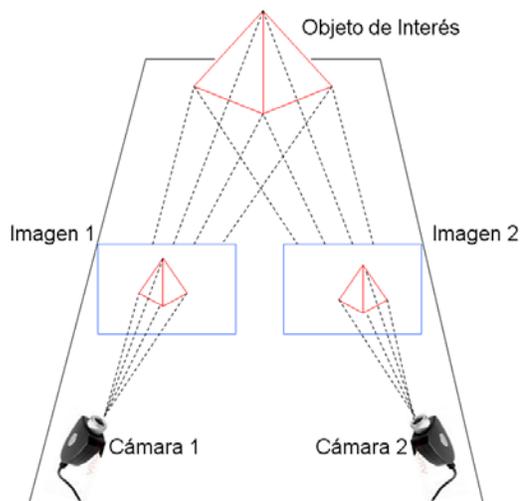


Figura 3.34.a: Posición de cámaras



Figura 3.34.b: Logitech Quickcam Pro 9000®

3.2 OBJETO DE INTERÉS

El sistema de Visión Artificial necesita un objeto para reconocer y representar su posición en un ambiente tridimensional. Este objeto debe tener un color característico que lo identifique, en caso contrario, marcas de colores pueden ser adheridas al objeto. El movimiento del objeto debe estar limitado al rango de visión de ambas cámaras Web, ya que si una de estas pierde de vista al objeto el sistema se detendrá momentáneamente hasta que el objeto de interés vuelva al rango de visión.



Figura 3.35: Diversos objetos de interés

3.3 ILUMINACIÓN

La luz toma un papel fundamental al momento de reconocer un color específico en una imagen. Cierta tipo de luz puede afectar a un color hasta el punto de tener otra tonalidad. Por lo tanto es necesario crear y ubicar iluminación que sea homogénea que no afecte al objeto de interés de forma tal que cambie su color. Esta luz es provista por una lámpara casera que consta de un interruptor de encendido o apagado, y un bombillo de 5 voltios (figura 36).



Figura 3.36: Lámpara casera

3.4 AMBIENTES DEL SISTEMA

El sistema de Visión Artificial necesita un entorno físico en el cual desenvolverse. A continuación se presentarán dos entornos, sus partes y características en los cuales se desenvolverá el sistema.

3.4.1 AMBIENTE CONTROLADO

A partir de la necesidad de crear un ambiente en el cual las condiciones de iluminación, color de fondo, espacio y disposición de los elementos a usar sean controlados y conocidos, se creó un ambiente el cual ofrece condiciones ideales que puedan garantizar el resultado óptimo del sistema. El ambiente controlado consta de una caja hecha de cartón piedra que no está

totalmente cerrada (figura 37.a), de dimensiones (67cm X 45,2cm X 47cm) la cual en su interior está totalmente pintada de color blanco. El color blanco fue escogido ya que éste refleja la luz, facilitando la reflexión y distribución. De esta manera una lámpara casera será capaz de iluminar el ambiente controlado. Este ambiente no debe permitir que ninguna fuente de luz externa lo afecte. Por lo tanto contiene una tapa de las mismas dimensiones de una de las caras que no permite la entrada de luces o ruido al sistema.

La disposición de los objetos en el ambiente controlado se observa en la (figura 37.b). Cabe recalcar que la luz debe estar posicionada del lado contrario al objeto (mirando a la pared opuesta). De esta manera se reduce la luz especular en el objeto.



Figura 3.37.a: Ambiente controlado

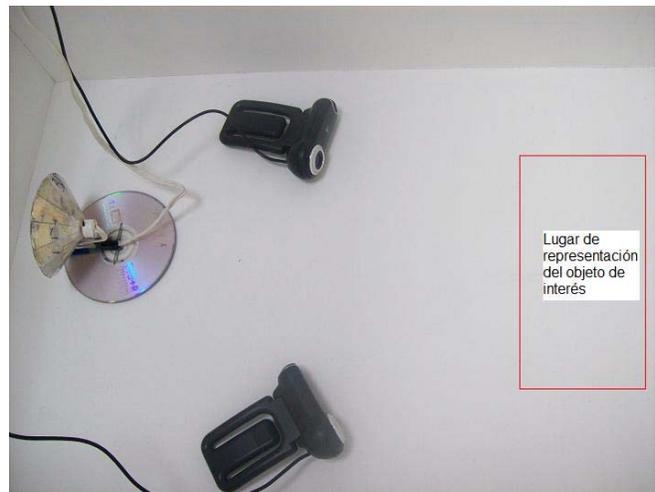


Figura 3.37.b: Disposición de objetos en el ambiente controlado

3.4.2 SIMULAP V-1

Es una estructura utilizada para entrenamientos de intervenciones laparoscópicas la cual asemeja la forma de un abdomen humano y en donde se insertan diversos instrumentos quirúrgicos que se utilizan durante dichas intervenciones (figura 38.a). El SIMULAP V-1 fue desarrollado por el INABIO.

Debido al material reflexivo de esta estructura y su color negro es necesario crear condiciones distintas a las del ambiente controlado. En este caso se cubrió totalmente con tela negra.

La disposición de los objetos en el SIMULAP v-1 se puede observar en la (figura 38.b).

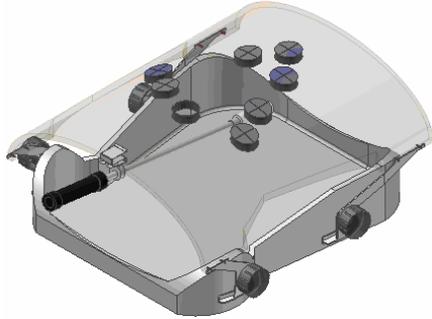


Figura 3.38.a: SIMULAP V-1



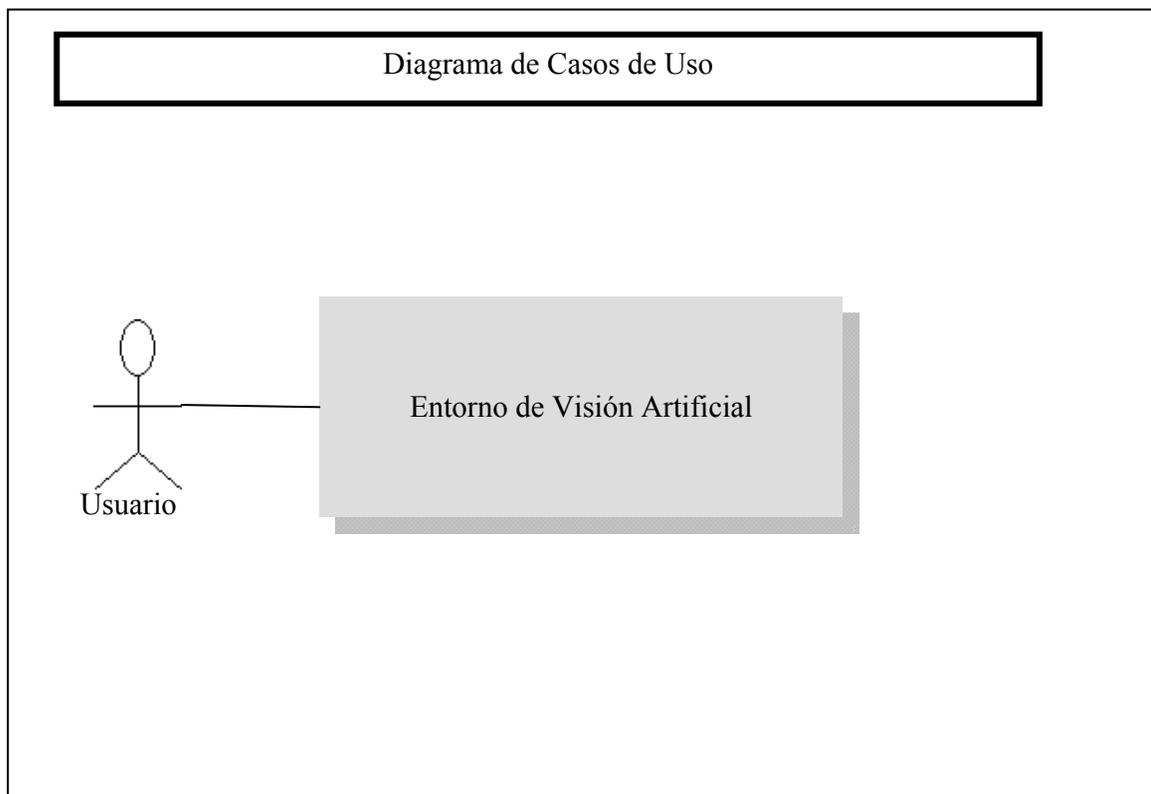
Figura 3.38.b: Disposición de objetos en el SIMULAP V-1

CAPÍTULO 4: DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN

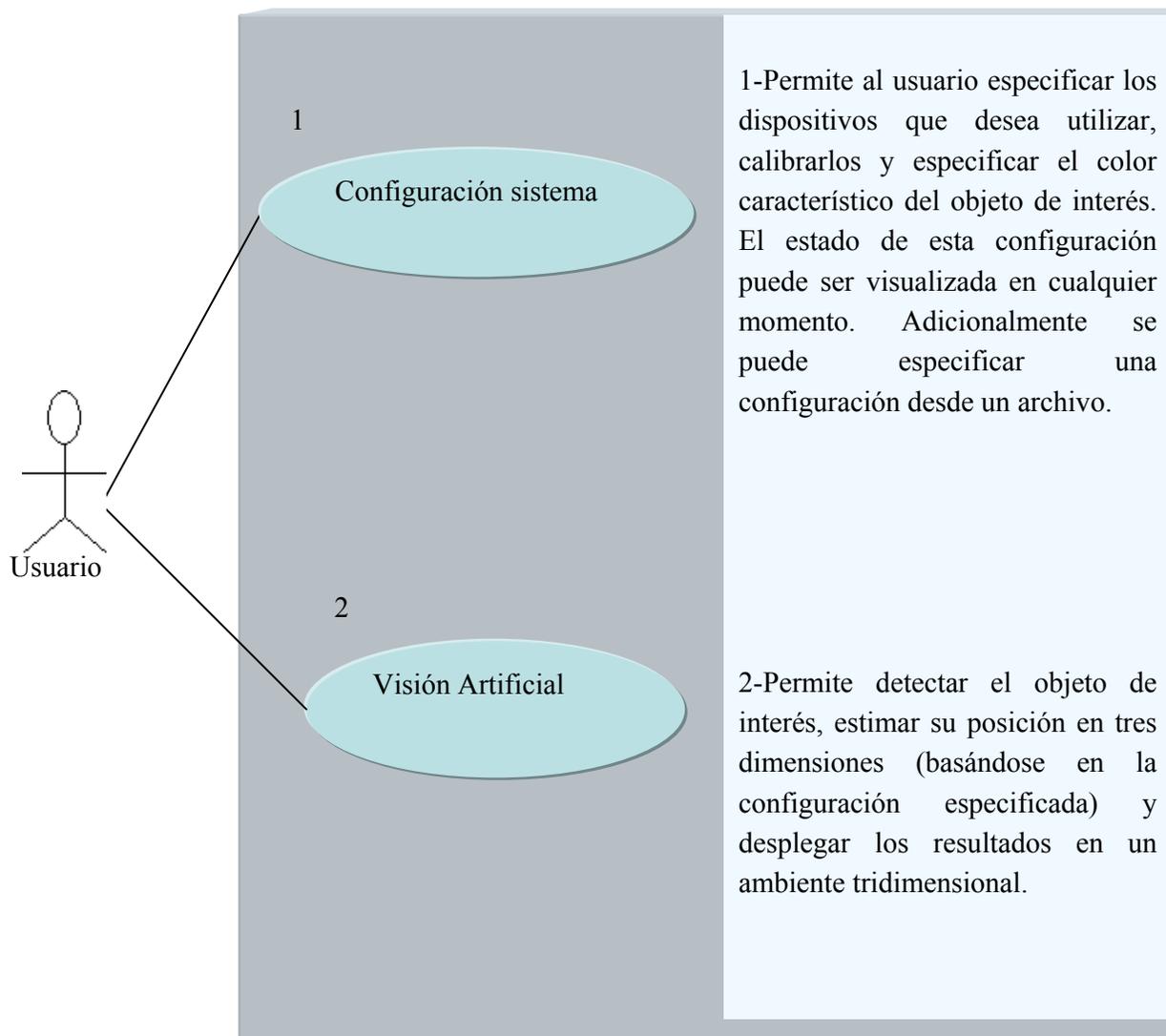
Una vez fijados los objetivos y elegidos los componentes físicos es necesario verificar la interacción del usuario con el sistema así como también la creación, diseño e implementación de sus módulos principales.

4.1 CASOS DE USO

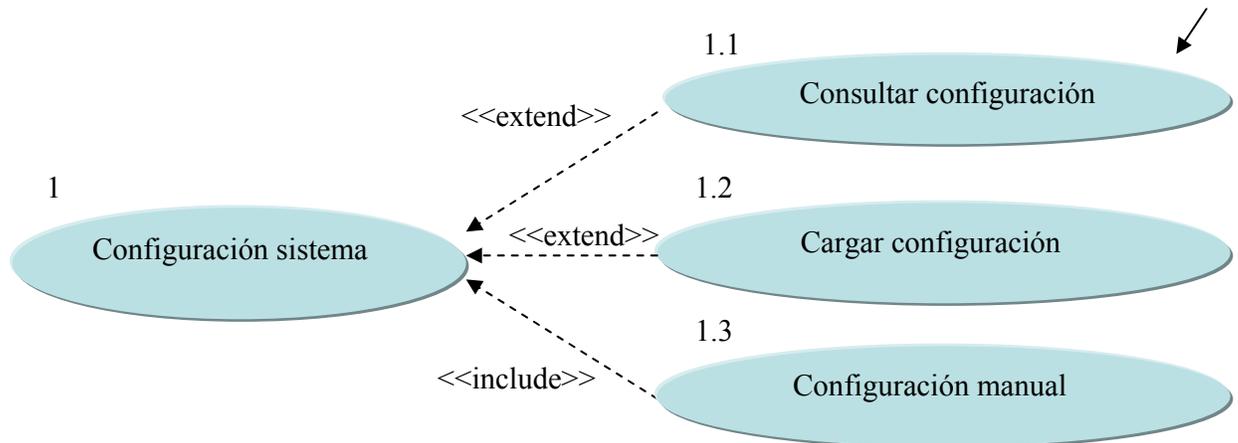
4.1.1 NIVEL 0: DESCRIPCIÓN DE ACTORES Y SU INTERACCIÓN CON EL SISTEMA



4.1.2 NIVEL 1: CASOS DE USO PRINCIPALES



4.1.3 NIVEL 2: REFINAMIENTO DE LOS CASOS DE USO PRINCIPALES Y SUS RELACIONES



Nombre: Consultar configuración.

Precondición: Ninguna.

Actores relacionados: Usuario.

Actores principales: Usuario.

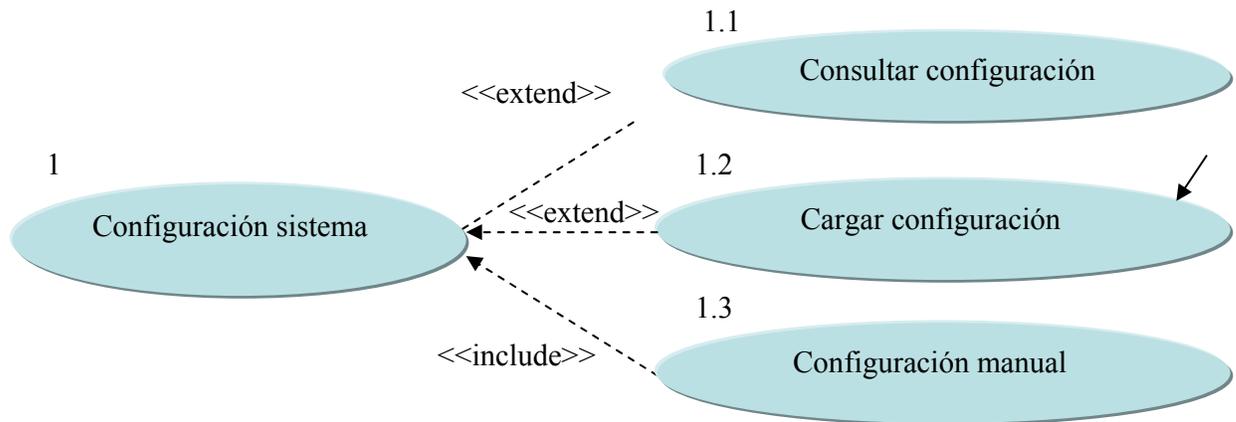
Flujo o Pasos:

Sistema	Usuario
1.- El sistema toma la información hasta el momento de la configuración actual y la despliega en una ventana. La información consiste en nombre del dispositivo, matriz asociada y rangos representativos del objeto de interés. Estos datos se despliegan dos (2) veces uno por cada dispositivo.	

Descripción: Funcionalidad que le permite a un usuario visualizar el estado global de la configuración del sistema.

Poscondición: Ninguna.

Dependencia con otros Casos de Uso: Configuración sistema.



Nombre: Cargar configuración.

Precondición: Ninguna.

Actores relacionados: Usuario.

Actores principales: Usuario.

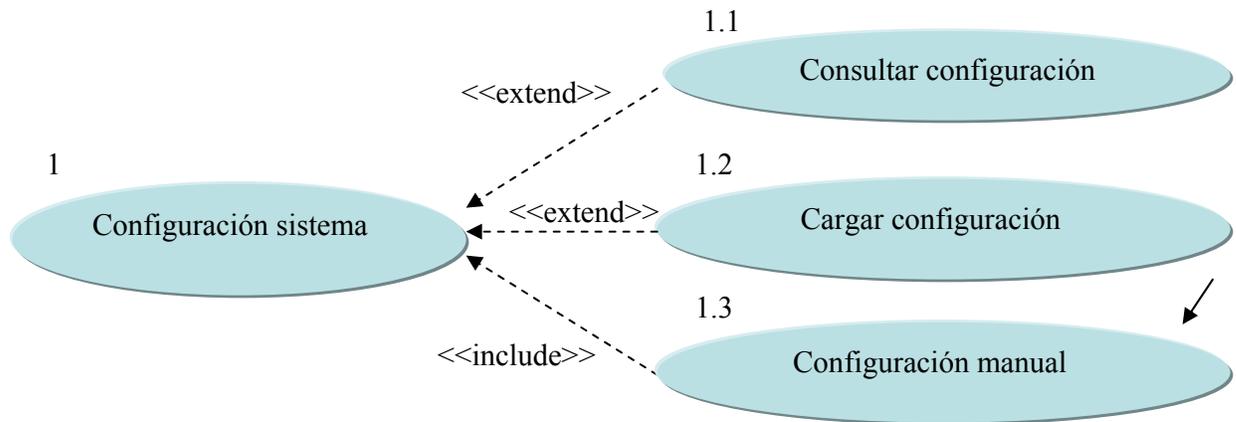
Flujo o Pasos:

Sistema	Usuario
	1.- El usuario selecciona el archivo en donde se encuentra la configuración.
2.- El sistema toma la ruta del archivo seleccionado extrae el contenido (nombre del primer dispositivo, matriz y valores del color característico; lo mismo sucede para el segundo dispositivo). Luego se despliega un mensaje de confirmación donde se pregunta si se desea sobrescribir la configuración actual.	
	3.- El usuario acepta o rechaza la sobrescritura de la configuración.
4.- Si el usuario acepta la configuración actual será sobrescrita por la información contenida en el archivo, sino la configuración actual se preserva.	

Descripción: Funcionalidad que le permite a un usuario configurar la aplicación desde un archivo sin necesidad de realizar todos los pasos de configuración.

Poscondición: Ninguna.

Dependencia con otros Casos de Uso: Configuración sistema.



Nombre: Configuración manual.

Precondición: Ninguna.

Actores relacionados: Usuario.

Actores principales: Usuario.

Flujo o Pasos:

Sistema	Usuario
	1.- El usuario selecciona los dispositivos a utilizar.
2.- El sistema guarda la selección.	
	3.- El usuario calibra los dispositivos seleccionados.
4.- El sistema guarda los resultados de la calibración.	
	5.- El usuario selecciona el color del objeto de interés.
6.- El sistema guarda el color característico del objeto de interés.	

Descripción: Funcionalidad que le permite a un usuario seleccionar los dispositivos a utilizar, calibrarlos y especificar el color característico del objeto de interés.

Poscondición: Ninguna.

Dependencia con otros Casos de Uso: Configuración sistema.

Visión Artificial

Nombre: Visión Artificial.

Precondición: Configuración sistema.

Actores relacionados: Usuario.

Actores principales: Usuario.

Flujo o Pasos:

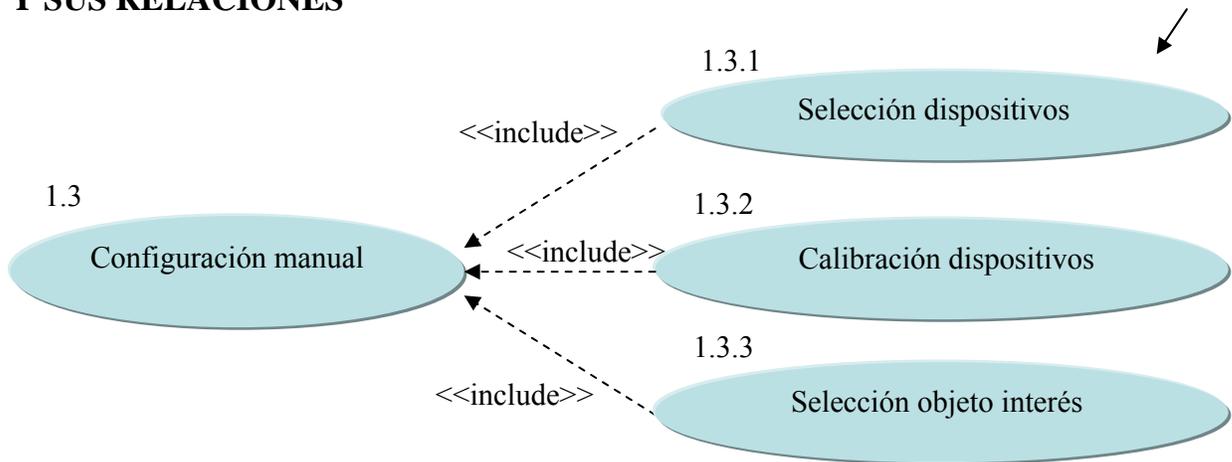
Sistema	Usuario
	1.- El usuario comienza el video en ambos dispositivos.
2.- El sistema por cada imagen procesada (de ambos dispositivos) realiza un preprocesamiento para eliminar ruido y arreglar detalles. El siguiente paso es realizar una segmentación en base al color del objeto de interés (configurado previamente). Luego al resultado de la segmentación se le calcula su caja delimitadora y su centro relativo a la imagen (donde esta contenido el posible objeto de interés). Por último se eliminan los falsos positivos de la etapa anterior y se selecciona uno sólo. Una vez obtenido un sólo resultado por cada una de las imágenes del primer dispositivo y del segundo, se utiliza el centro y la matriz de cada dispositivo para calcular la posición tridimensional del objeto utilizando geometría epipolar. Para finalizar se toma dicha coordenada y se coloca en un ambiente simulado donde se puede ver el movimiento en tres dimensiones y en tiempo real.	

Descripción: Funcionalidad que le permite a un usuario capturar el objeto de interés y representarlo en un ambiente en tres dimensiones simulado.

Poscondición: Ninguna.

Dependencia con otros Casos de Uso: Ninguna.

4.1.4 NIVEL 3: REFINAMIENTO DE LOS CASOS DE USO SECUNDARIOS Y SUS RELACIONES



Nombre: Selección de dispositivos

Precondición: Disponibilidad física de los dispositivos de captura.

Actores relacionados: Usuario.

Actores principales: Usuario.

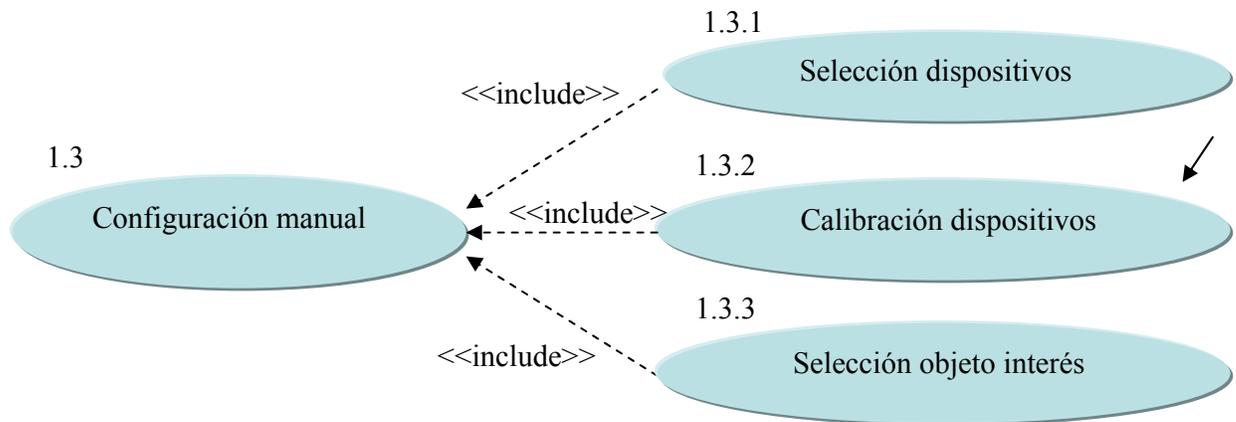
Flujo o Pasos:

Sistema	Usuario
1.- El sistema despliega una lista con todos los dispositivos disponibles.	
	2.- El usuario selecciona el primer dispositivo de captura.
3.- El sistema despliega otra lista con todos los dispositivos disponibles, exceptuando el seleccionado en el paso anterior.	
	4.- El usuario selecciona el segundo dispositivo de captura y acepta.
5.- El sistema guarda los dispositivos seleccionados.	

Descripción: Funcionalidad que le permite a un usuario seleccionar los dispositivos que desea utilizar en el proceso.

Poscondición: Ninguna.

Dependencia con otros Casos de Uso: Configuración manual.



Nombre: Calibración de dispositivos

Precondición: Selección previa de los dispositivos de captura.

Actores relacionados: Usuario.

Actores principales: Usuario.

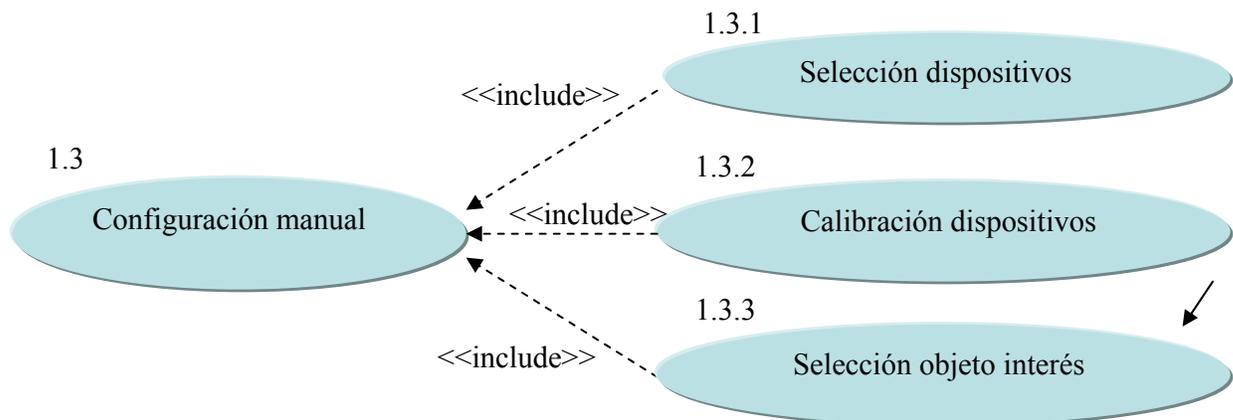
Flujo o Pasos:

Sistema	Usuario
1.- El sistema despliega los dispositivos elegidos previamente.	
	2.- El usuario selecciona el dispositivo que desea calibrar y especifica el conjunto de imágenes que contienen el patrón para la calibración.
3.- El sistema toma el conjunto de imágenes y las procesa una a una para obtener una matriz asociada al dispositivo en cuestión.	
	4.- Si el usuario está de acuerdo con los valores obtenidos en la matriz la configuración es guardada.
5.- El sistema guarda dicha matriz y se la asocia al dispositivo seleccionado (este proceso debe hacerse por cada dispositivo de captura).	

Descripción: Funcionalidad que le permite a un usuario calibrar los dispositivos seleccionados previamente.

Poscondición: Ninguna

Dependencia con otros Casos de Uso: Configuración manual, selección dispositivos.



Nombre: Selección objeto de interés.

Precondición: Selección previa de los dispositivos de captura y calibración de ambos.

Actores relacionados: Usuario.

Actores principales: Usuario.

Flujo o Pasos:

Sistema	Usuario
1.- El sistema despliega los dispositivos seleccionados previamente.	
	2.- El usuario selecciona un dispositivo y presiona el botón de video.
3.- El sistema prepara el dispositivo seleccionado y comienza el video.	
	4.-Se coloca el objeto de interés frente a la cámara y se toma una foto.
5.-El sistema despliega la foto.	
	6.-El usuario selecciona el área donde se encuentra el color del objeto de interés y procede a calcular su valor respectivo.
7.-El sistema calcula los valores representados en el área de selección y los despliega al usuario.	
	8.-El usuario revisa los valores obtenidos y si esta de acuerdo guarda los resultados.
9- El sistema guarda los valores y se los asocia al dispositivo seleccionado (este proceso debe hacerse por cada dispositivo de captura).	

Descripción: Funcionalidad que le permite a un usuario especificar y calcular el color del objeto de interés.

Poscondición: Ninguna

Dependencia con otros Casos de Uso: Configurar sistema, selección dispositivos, calibración dispositivos.

4.2 DIAGRAMA DE CLASES

El sistema de Visión Artificial fue estructura bajo el patrón de diseño Modelo Vista Controlador (MVC), donde los elementos de interfaz son parte de la vista, el modelo es el encargado de las operaciones funcionales del sistema y el controlador es el encargado de intercambiar mensajes entre la vista y el modelo. Existen controladores por cada interfaz utilizada y un controlador principal que gestiona a los demás. Esta metodología permite separar los elementos de interfaz de la lógica del programa, gracias a esto es posible la creación del Modelo que es una librería centralizada en donde se encuentran todas las herramientas de visión artificial, calibración y gestión de dispositivos. A continuación un extracto del diagrama de clases con los elementos más relevantes figura 39.

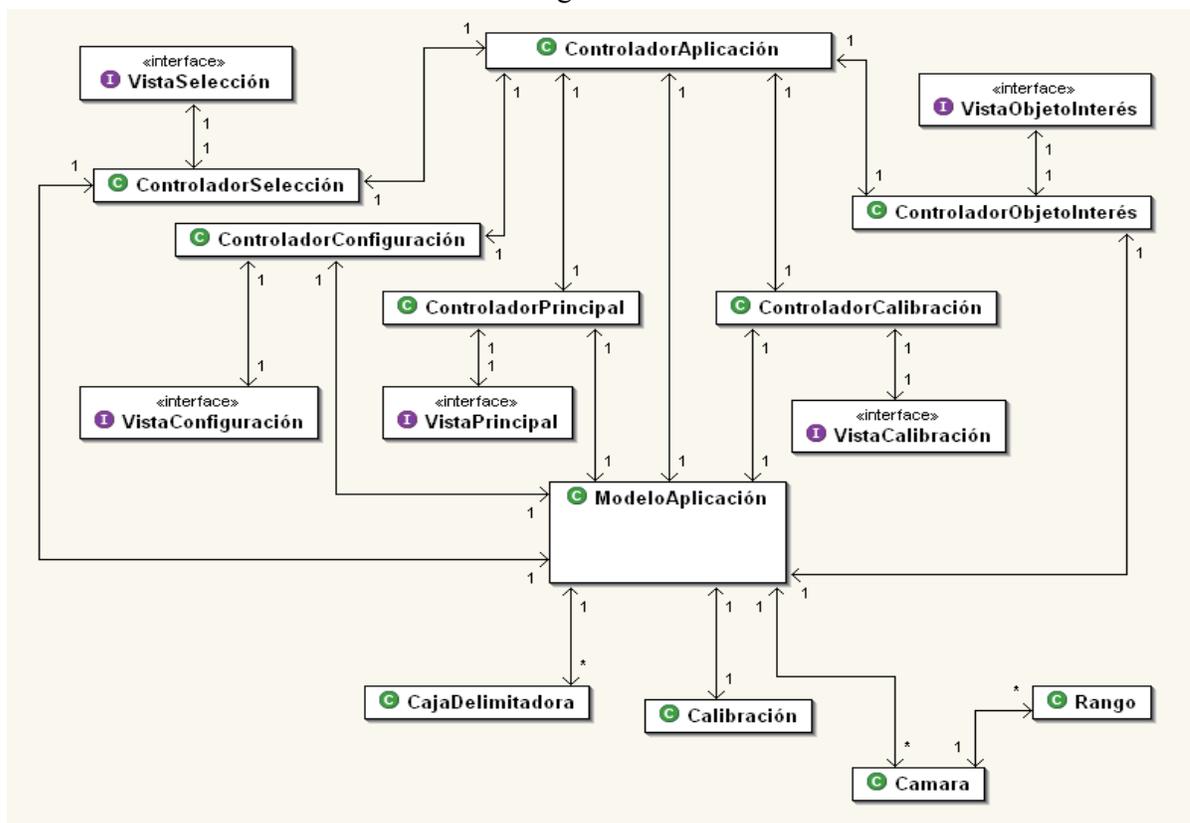


Figura 4.39: Extracto de diagrama de clases

4.2.1 ESTRUCTURA DE LAS CLASES MÁS RELEVANTES DEL SISTEMA

La clase calibración es de gran importancia ya que es posible estimar la matriz de proyección de cualquier dispositivo de captura, a través de un conjunto de imágenes que contengan un patrón de calibración específico. La calibración se logra utilizando la librería OpenCV.

Calibracion
+requerimientoFoto : vector<int>
+Calibracion() +~Calibracion() +calibrar(in nombres : vector<char *>, in * extension : char) : Matrix3x3 -dibujarCirculos() : void -initCorners3d() : void -inicializar() : void +actualizar(in valor : int) : void

Tabla 4.1: Clase Calibracion

La clase cámara es necesaria ya que posee toda la información de un dispositivo de captura como: matriz de proyección, color del objeto de interés a detectar e información descriptiva del dispositivo. Esta estructura permite manejar toda la información referente al dispositivo de captura fácilmente.

Camara
-identificador : int -descripcionCamara : CameraDescription -proyeccionCamara : Matrix3x3 +intervalo : vector<Rangos>
+Camara() +Camara(in identificador : int) +~Camara() +getNombre() : char +setIdentificador(in identificador : int) : bool +getIdentificador() : int +setProyeccion(in distanciaFocalX : double, in distanciaFocalY : double) : void +getProyeccion() : Matrix3x3 +esValida() : bool +calibrada() : bool +hayIntervalo() : bool

Tabla 4.2: Clase Camara

El modelo es un componente de software para Visión artificial centralizado que es posible adaptarlo a cualquier sistema. Utiliza las primitivas de OpenCV para el tratamiento digital de imágenes y el paquete LAPACK para la resolución de sistemas de ecuaciones lineales en el cálculo de la geometría epipolar.

ModeloAplicacion
<pre> +ancho : int +alto : int +anterior1 : CajaDelimitadora +anterior2 : CajaDelimitadora +foto : IplImage +coloresDetectar : vector<Rangos> +camara1 : Camara +camara2 : Camara +coordenada : Vector3 +deteccion : bool -calibracion : Calibracion +setPixel(inout origen : IplImage, in x : int, in y : int, in color : Vector3) : void +getPixel(inout origen : IplImage, in x : int, in y : int) : Vector3 +calcularRangos(inout origen : IplImage, in primero : Vector2, in ultimo : Vector2) : Rango +umbralizacion24(in origen : IplImage, out destino : IplImage, in valores : Rango, in positivo : Vector3, in negativo : Vector3) : void +detectarColor(inout origen : IplImage, in inicio : Vector2, in colorObjetivo : Vector3, in colorDestino : Vector3) : CajaDelimitadora +calcularBB(inout origen : IplImage, in colorObjetivo : Vector4, in colorDestino : Vector4) : vector<CajaDelimitadora> +openingIplImage(inout imagen : IplImage, in tamanoErosion : int, in formaErosion : int, in tamanoDilatacion : int, in formaDilatacion : int) : void +closingIplImage(inout imagen : IplImage, in tamanoErosion : int, in formaErosion : int, in tamanoDilatacion : int, in formaDilatacion : int) : void +erosionIplImage(inout imagen : IplImage, in tamano : int, in forma : int) : void +dilatacionIplImage(inout imagen : IplImage, in tamano : int, in forma : int) : void +convertirIplImage_QImage(in imagen : IplImage) : QImage +dgesvd(in **A : double, in m : int, in n : int, out *S : double, out **U : double, out **VT : double) : void +calculoEpipolar(in camara1 : Camara, in camara2 : Camara, in p1 : Vector2, in p2 : Vector2) : Vector3 +preprocesamiento(inout imagen : IplImage) : void +segmentacion(inout imagen : IplImage, in coloresDetectar, in positivo : Vector3, in negativo : Vector3) : void +extraccionCaracteristicas(inout imagen : IplImage, in positivo : Vector3, in destino : Vector3) : vector<CajaDelimitadora> +procesamientoAltoNivel(in imagen1, in imagen2, in camara1, in camara2, in b1, in b2) : Vector3 +cargarArchivo(in * ruta : char) : bool +suavizarMovimiento(in centro : Vector2, in anterior : CajaDelimitadora) : CajaDelimitadora +eliminarFalsosPositivos(in caja : CajaDelimitadora) : void +liberarDispositivos() : void +getCamarasDisponibles() : void +comenzarCaptura() : void +detenerCaptura() : void +iniciarCaptura() : void +terminarCaptura() : void +getFotoIplImage() : IplImage +callback(inout imagen : IplImage) : void +stereoCallback(inout imagen1 : IplImage, inout imagen2 : IplImage) : void +setUnaCamara(in camara : Camara, in elementoDespliegue : cvcamWindow) : bool +setDosCamaras(in camara1 : Camara, in elementoDespliegue1, in camara2 : Camara, in elementoDespliegue2 : cvcamWindow) : bool +calibrarCamara(in listalimagenes, in * extension : char, inout camara : Camara) : int </pre>

Tabla 4.3: Clase ModeloAplicacion

4.3 PLATAFORMA DE DESARROLLO

El Sistema de Visión Artificial fue desarrollado en el lenguaje de programación C++. Luego para el tratamiento de las imágenes y gestión de los dispositivos se utilizó OpenCV [16]. La librería LAPACK [13] permitió resolver sistemas de ecuaciones homogéneas de tipo $AX = 0$ por el método de Valor Singular de Descomposición (SVD por sus siglas en inglés). Las Interfaces gráficas de usuario fueron creadas con ayuda de la librería QT [24] y por ultimo se utilizo OpenGL® [28] para el despliegue en tres dimensiones de la posición del objeto de interés.

4.3.1 LENGUAJE DE PROGRAMACIÓN C++

El lenguaje de programación C++ ofrece las características sintácticas y funcionales del lenguaje C y agrega sus propios elementos, como lo es la programación orientada a objetos. Este último nos permite realizar programas bajo el paradigma orientado a objetos permitiéndonos Abstracción, Encapsulamiento, Herencia y Polimorfismo. De esta forma es posible crear aplicaciones modulares, portables, escalables, reutilizar código, además de su alta compatibilidad con otras aplicaciones, velocidad de procesamiento y ejecución. Un programa en C++ esta conformado fundamentalmente por una serie de archivos de cabecera, declaraciones y definiciones, así como una función principal denominada main. En base a todo lo expresado anteriormente el lenguaje de programación C++ es ideal para el desarrollo del sistema ya que es compatible con las tecnologías requeridas.

4.3.2 OPENCV

Es una librería de código abierto de Visión Artificial desarrollada por INTEL® con una amplia variedad de herramientas para la interpretación de imágenes digitales. Es compatible con la Librería de Procesamiento de Imágenes (IPL, por sus siglas en inglés) de INTEL®, la cual implementa operaciones de bajo nivel en imágenes digitales. A pesar de contar con primitivas como binarización, filtrado, estadísticas sobre la imagen, pirámides; OpenCV es en su mayoría librerías de alto nivel en donde implementan algoritmos para técnicas de calibración (Calibración de cámaras), detección de características y rastreo, análisis de formas (Geometría, procesamiento de contornos), análisis de movimiento (Plantillas de movimiento, estimadores), reconstrucción 3D, segmentación de objetos y reconocimiento. La característica esencial de la librería, aparte de funcionalidad y calidad, es su desempeño. Los algoritmos están basados en estructuras de datos flexibles (Estructuras de datos dinámicas) unidas con estructuras de datos IPL; más de la mitad de las funciones han sido optimizadas en lenguaje ensamblador aprovechando así las ventajas de la arquitectura INTEL®.

La librería OpenCV es una manera de establecer una “comunidad de código abierto para la visión” la cual permite ampliar los desarrollos y conocimientos mas recientes y aplicarlos a la computación visual en el área computacional. El software provee un conjunto de funciones para el procesamiento de imágenes así como también funciones de análisis para imágenes y patrones. Esta librería posee una interfaz independiente de la plataforma y un conjunto de códigos fuentes escritos en el lenguaje de programación C.

4.3.3 LAPACK

Es una librería que consta de un conjunto de rutinas escritas en Fortran77 que tienen como finalidad resolver sistemas simultáneos de ecuaciones lineales, solucionar sistemas de ecuaciones lineales utilizando mínimos cuadrados, problemas de autovalores y problemas de valor singular. Permite adicionalmente calcular factorizaciones de matrices (LU, Cholesky, QR, SVD entre otras). Es capaz de manejar matrices densas y condicionadas, pero no puede manejar matrices generales esparcidas. Los valores provistos son para matrices reales y complejas, permite simple y doble precisión.

Para este caso en específico fue necesario utilizar un conversor que permite traducir código escrito en Fortran77 a C++ y viceversa. Esta librería se utilizó para resolver el sistema de ecuaciones $AX = 0$ utilizando SVD.

4.3.4 QT

Es un marco de trabajo para desarrollo para aplicaciones independiente de la plataforma. Es usado ampliamente para el desarrollo de aplicaciones con Interfaces Gráficas de Usuario (GUI, por sus siglas en inglés), así como también para el desarrollo de aplicaciones de consola y servidores.

Qt utiliza C++ junto con una gran cantidad de extensiones no estándar implementadas por un preprocesamiento adicional que genera código C++ estándar antes de la compilación. También puede ser utilizado con muchos otros lenguajes de programación como por ejemplo C#, Java, Pascal, Perl, PHP, Ruby y Python. Es compatible con las plataformas más comerciales, y posee soporte internacionalizado extensible. Adicionalmente incluye funcionalidades SQL para acceso a base de datos, XML para tratar archivos, utiliza manejo de hilos, posee soporte de red y cuenta con un API unificado e interoperable para el manejo de los archivos.

4.3.5 OPENGL®

Es una interfaz de software para tarjetas gráficas. La interfaz consiste en un conjunto de procedimientos y funciones que le permiten al programador especificar los objetos y operaciones involucradas en la producción de imágenes gráficas de alta calidad, específicamente imágenes a color de objetos tridimensionales.

La mayor parte de las funciones de OpenGL® requieren que la tarjeta gráfica contenga un framebuffer. Esta interfaz se caracteriza en el despliegue de objetos como puntos, líneas, polígonos, mapas de bits y todo tipo de figuras en tres dimensiones de forma eficiente ya que están implementadas a nivel de hardware.

4.4 ENTORNO DE VISIÓN ARTIFICIAL

La aplicación desarrollada en este Trabajo Especial de Grado lleva como nombre Entorno de Visión Artificial. En los siguientes párrafos se comentará el diseño e implementación del sistema, así como también sus características y funcionalidades.

La aplicación consta de tres módulos principales (figura 40): configurar sistema, comenzar Visión Artificial y despliegue tridimensional. Previo a la simulación es necesario realizar la configuración pertinente. Esta consta en seleccionar dos (2) dispositivos de captura, calibrarlos y elegir el patrón de colores que representa al objeto de interés. Adicionalmente se puede consultar la configuración actual la cual ofrece el estado global del sistema, que incluye sus 3 pasos, elección de dispositivos, calibración y selección de patrón de colores del objeto de interés. Es posible especificar en un archivo una configuración determinada y luego cargarla para ser usada.

El módulo de Visión Artificial se encarga de extraer, filtrar, caracterizar, clasificar la información mediante procesamiento de imágenes y luego se calcula la posición en tres dimensiones del objeto de interés. Por último el despliegue tridimensional obtiene estas coordenadas y las representa visualmente al usuario en un ambiente simulado.

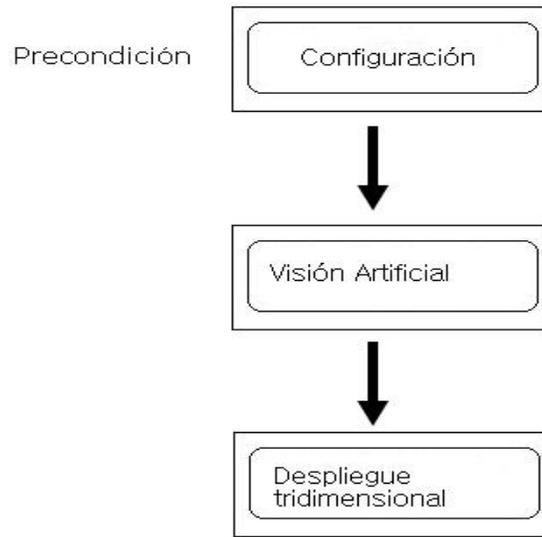


Figura 4.40: Esquema global del sistema

4.4.1 CONFIGURACIÓN DE LA APLICACIÓN

Previa a la utilización del sistema es condición necesaria especificar una configuración. Esto se puede lograr de dos maneras manualmente o especificada desde un archivo. La configuración consta en ambos casos de selección de dispositivos, calibración y especificación de colores del objeto de interés. Adicionalmente se puede visualizar el estado de la configuración en cualquier momento (figura 41).

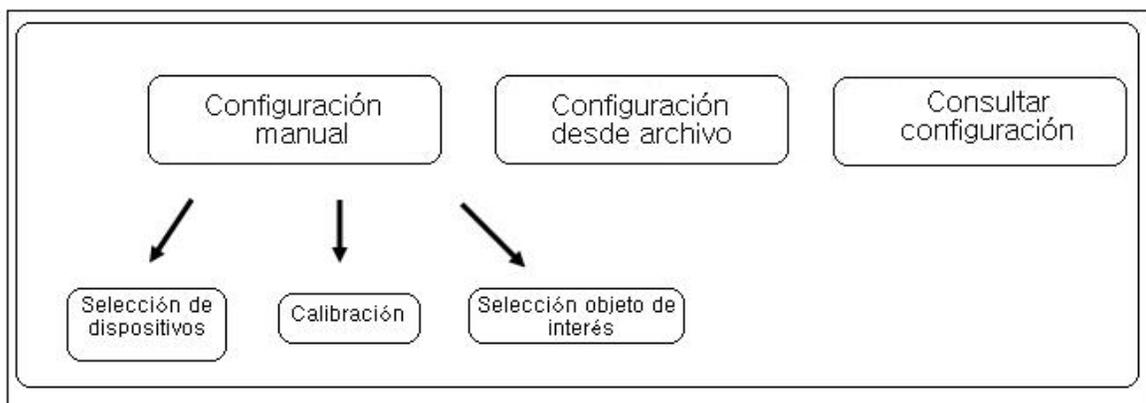


Figura 4.41: Módulo de configuración

Configuración Manual

Este método de configuración le permite al usuario especificar cada uno de los elementos del sistema manera manual (dispositivos, calibración y objeto de interés), permitiéndole visualizar las diferentes etapas y seguir los pasos para poder adaptar el sistema a su caso específico. Este modo de configuración es el más indicado para principiantes. Para poder configurar el sistema de manera manual es necesario completar tres (3) etapas: selección de dispositivos, calibración de dispositivos, objeto de interés.

Seleccionar Dispositivos

El primer paso de la configuración es escoger los dispositivos de captura que se desean utilizar. Para esto es necesario validar la existencia de al menos dos de ellos. Este módulo se encarga de hacer transparente la gestión y la disponibilidad de estos dispositivos. Utilizando la librería OpenCV es posible solicitar al sistema operativo una lista de los dispositivos de captura conectados físicamente al computador y listos para ser utilizados, obteniendo por cada uno de ellos su identificador y nombre. Luego esa lista se despliega en el sistema para que el usuario seleccione el primer dispositivo que desea utilizar. En base a la selección anterior se genera otra lista con todos los dispositivos menos el escogido, para que el usuario seleccione el segundo a utilizar. Una vez seleccionados ambos dispositivos, el sistema guarda ambos como dos instancias del objeto cámara con su nombre e identificador respectivo.

Calibración de los dispositivos seleccionados

Una vez realizada la selección de dos (2) dispositivos de captura (como se especificó en la etapa anterior) se procede a calibrarlos con ayuda de la librería OpenCV. La calibración es uno de los pasos más importantes del proceso ya que gracias a ella se puede deducir y calcular la matriz de proyección del dispositivo. Gracias a esta matriz es posible realizar la transformación del plano objeto (mundo real, coordenadas 3D) al plano imagen (imagen digital, coordenadas en 2D). Para este caso en específico es necesario obtener dicha matriz ya que el cálculo de la geometría epipolar involucra las matrices de proyección de cada dispositivo.

El proceso de calibración de los dispositivos se realiza en base a un conjunto de imágenes capturadas desde el dispositivo seleccionado las cuales llamaremos imágenes de calibración. Estas imágenes deben tener ciertas condiciones especiales [26] (para obtener las imágenes de calibración ideales, ver anexos). Una vez obtenidas las de imágenes de calibración, se selecciona la cantidad a utilizar (es recomendable utilizar al menos 25 imágenes) especificando la ruta de cada una de ellas.

El primer paso es especificar el patrón de calibración y las dimensiones del mismo. Para este caso se utilizó un tablero de ajedrez estándar (8x8) que cuenta con siete filas y siete columnas internas. Luego por cada imagen en la lista se determina si posee un patrón válido (función `cvFindChessBoardCornerGuesses`) y se localizan las esquinas del tablero de ajedrez (función `cvFindCornerSubPix`). De no poseerlo esta imagen no será tomada en cuenta. Luego en base a la posición del tablero y sus esquinas es posible calcular las esquinas internas del tablero de manera tal que un punto esté rodeado diagonalmente por dos cuadros blancos y dos negros (figura 42) [27].

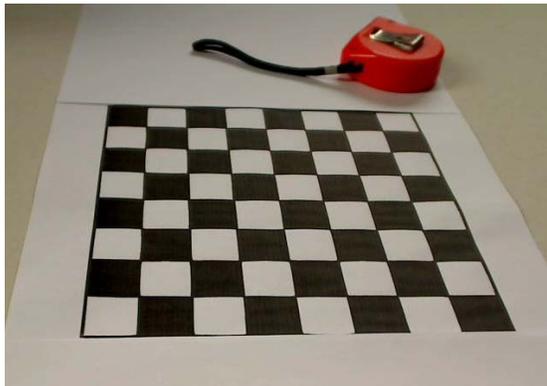


Figura 4.42.a: Patrón de calibración

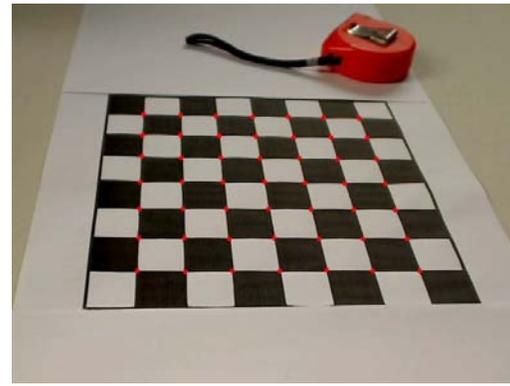


Figura 4.42.b: Resultado de la calibración

Este proceso se realiza una vez por cada una de las imágenes guardando en cada iteración la información de las esquinas internas. Por último se procede a calcular la matriz de la cámara basándose en la cantidad de imágenes utilizadas, las esquinas encontradas en todas las imágenes y la distorsión de la cámara (función `cvCalibrateCamera`). El reconocimiento del patrón, cálculo de las esquinas internas y cálculo de la matriz de la cámara fueron realizados con ayuda de un conjunto de rutinas de la librería OpenCV.

El resultado es desplegado al usuario en la interfaz gráfica. Si el usuario está de acuerdo con los valores obtenidos en el proceso, los valores serán guardados en una matriz que se le asociará al dispositivo que se está calibrando actualmente. Este proceso es necesario realizarlo por cada uno de los dispositivos que se deseen utilizar en el sistema.

Selección objeto de interés

El último paso de la configuración es especificar la codificación de colores que representa al objeto de interés. Para lograr esto es necesario haber realizado las etapas de selección de dispositivos y calibración, así como también haber codificado el objeto de interés con los colores respectivos. En primera instancia se despliegan los dos (2) dispositivos seleccionados en etapas anteriores. El usuario debe especificar el dispositivo con el cual desea

trabajar. Luego se chequea la disponibilidad del dispositivo y se prepara para ser utilizado, permitiendo de esta manera utilizar las opciones de video que considere necesarias (comenzar video, detener video, tomar foto).

Una vez que el dispositivo comience a reproducir el video se coloca el objeto de interés (previamente configurado con la marca de colores) dentro del rango de visión del dispositivo. Cuando el usuario considere que el objeto está en una posición adecuada se debe proceder a tomar una foto. En base a esta foto tomada (la cual contiene al objeto de interés) se le permite al usuario especificar una caja delimitadora que marque el área donde está ubicado el color representativo del objeto de interés. Si el usuario está de acuerdo con la zona escogida, se procede a calcular la intensidad de rojo, verde y azul (RGB, por sus siglas en inglés) de la caja delimitadora.

Este calculo se implementó con la función `calcularRangos` la cual es capaz de recorrer toda la caja delimitadora almacenando los valores de los píxeles y calculando en forma de rangos los valores mínimos y máximo de cada canal. Esto se realizó con el fin de flexibilizar el rango de colores tomados ya que estos se pueden malinterpretar al momento de la detección debido a factores externos.

Luego el resultado del cálculo es desplegado al usuario por medio de una interfaz gráfica, mostrando un rango por cada intensidad de color (rojo, verde y azul). Si el usuario está de acuerdo con los valores obtenidos dicho rango es guardado y asociado al dispositivo seleccionado. Este proceso es necesario realizarlo por cada uno de los dispositivos que se desee utilizar en el sistema.

Configuración cargada desde un archivo

Este método de configuración le permite al usuario cargar una configuración específica desde un archivo en formato XML, en donde se especificará cada uno de los elementos del sistema. La estructura interna del archivo se muestra en la figura 43.

```

<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>
<Configuracion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Camara nombre = "">
    <Calibracion>
      <DistanciaFocalX></DistanciaFocalX>
      <DistanciaFocalY></DistanciaFocalY>
    </Calibracion>
    <Intervalos>
      <Rango>
        <Rojo minimo="" maximo=""></Rojo>
        <Verde minimo="" maximo=""></Verde>
        <Azul minimo="" maximo=""></Azul>
      </Rango>
    </Intervalos>
  </Camara>
  <Camara nombre = "">
    <Calibracion>
      <DistanciaFocalX></DistanciaFocalX>
      <DistanciaFocalY></DistanciaFocalY>
    </Calibracion>
    <Intervalos>
      <Rango>
        <Rojo minimo="" maximo=""></Rojo>
        <Verde minimo="" maximo=""></Verde>
        <Azul minimo="" maximo=""></Azul>
      </Rango>
    </Intervalos>
  </Camara>
</Configuracion>

```

Figura 4.43: Estructura archivo configuración

Para el caso específico de esta aplicación es necesario configurar dos (2) dispositivos de captura, la información necesaria para completar la configuración es nombre del dispositivo, matriz de la cámara (específicamente distancia focal X e Y) y especificar el rango de valores por cada intensidad de rojo, verde y azul (mínimo y máximo respectivamente).

El nombre del dispositivo es condición necesaria ya que el sistema valida la disponibilidad y existencia del mismo. Si alguno de estos falla los demás datos no serán tomados en cuenta; los valores de calibración y los intervalos no necesitan especificarse. El sistema es capaz de cargar una configuración incompleta es decir se puede especificar el nombre y los valores de la calibración pero no especificar los rangos. También permite a partir de una configuración incompleta cargar los datos especificados y completar el proceso la vía manual.

En base a la información especificada se crean dos instancias de los dispositivos y se rellenan con la información del archivo. Este módulo permite acelerar el proceso de configuración cuando se conoce de antemano la información de configuración. Este método es indicado para usuarios avanzados.

A continuación una plantilla modelo para detección de un objeto de color negro (figura 44).

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>
<Configuracion xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Camara nombre = "Logitech QuickCam Pro 9000">
    <Calibracion>
      <DistanciaFocalX>785.772</DistanciaFocalX>
      <DistanciaFocalY>782.344</DistanciaFocalY>
    </Calibracion>
    <Intervalos>
      <Rango>
        <Rojo minimo = "0" maximo = "32"></Rojo>
        <Verde minimo = "0" maximo = "32"></Verde>
        <Azul minimo = "0" maximo = "19"></Azul>
      </Rango>
    </Intervalos>
  </Camara>
  <Camara nombre = "Logitech QuickCam Pro 9000">
    <Calibracion>
      <DistanciaFocalX>785.772</DistanciaFocalX>
      <DistanciaFocalY>782.344</DistanciaFocalY>
    </Calibracion>
    <Intervalos>
      <Rango>
        <Rojo minimo = "0" maximo = "22"></Rojo>
        <Verde minimo = "0" maximo = "20"></Verde>
        <Azul minimo = "0" maximo = "11"></Azul>
      </Rango>
    </Intervalos>
  </Camara>
</Configuracion>
```

Figura 4.44: Ejemplo de configuración

Consultar configuración

Debido a las diferentes etapas de la configuración es difícil llevar la traza del estado global de los dispositivos, así como también verificar los valores obtenidos en etapas ya realizadas. A causa de esto es necesario consultar la información en cualquier momento.

El sistema presenta una interfaz gráfica dividida en dos (2) secciones (una para cada dispositivo) cada sección presenta la información obtenida en cada etapa de configuración, nombre completo del dispositivo, matriz de la cámara (matriz 3x3) y por último el rango de intensidades de rojo, verde y azul que representan el objeto de interés. Todos estos valores que se visualizan, son sólo de lectura; no es posible editarlos. Para realizar ediciones es necesario realizar la etapa de configuración respectiva y sobrescribir los valores obtenidos en el intento anterior.

4.4.2 COMENZAR VISIÓN ARTIFICIAL

Una vez realizado el proceso de configuración el sistema está listo para procesar las imágenes de los dispositivos escogidos y estimar la posición del objeto de interés en tres dimensiones.

El proceso de Visión Artificial consiste en la realización de cinco (5) etapas las cuales son: adquisición de la imagen, preprocesamiento, segmentación, extracción de características y procesamiento de alto nivel (figura 45).

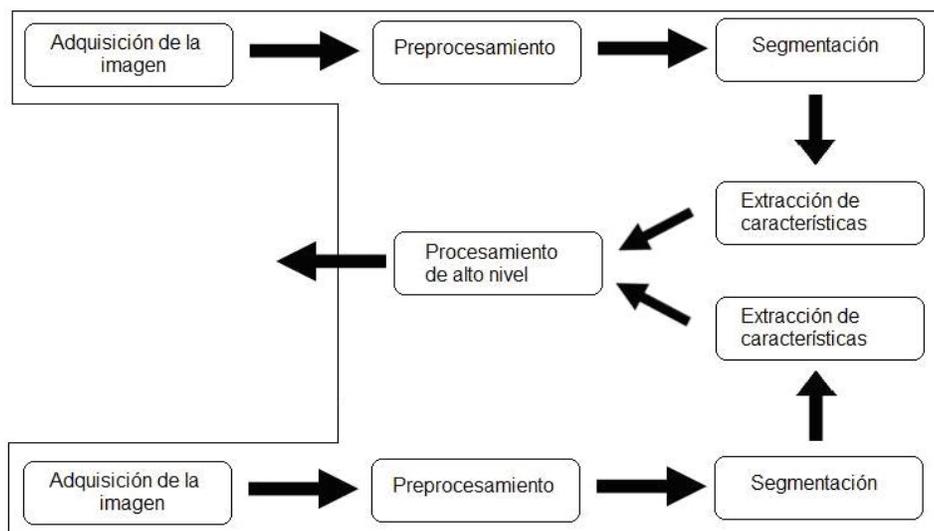


Figura 4.45: Módulo de Visión Artificial

Adquisición de la imagen

En primera instancia el sistema verifica la existencia y disponibilidad de los dispositivos seleccionados, de manera que estén preparados para utilizarse. Una vez realizada la verificación, las opciones de video son habilitadas permitiéndole al usuario comenzar el proceso cuando lo desee. En esta etapa del proceso es necesario verificar que los dispositivos aún estén disponibles para luego poder obtener las instancias de ambas imágenes con ayuda de la librería OpenCV. La cantidad de imágenes que se pueden procesar por iteración están atadas a la capacidad que presente el dispositivo de captura. Esta etapa sólo se encarga de validar disponibilidad y obtener las imágenes simultáneamente.

Preprocesamiento

Es necesario realizar un tratamiento previo a la imagen. Este consiste en corregir errores introducidos por los dispositivos de captura y el ambiente. Este procesamiento es realizado debido a que la calidad de la imagen está atada al desempeño del dispositivo y a las condiciones ambientales como por ejemplo el ruido, iluminación inadecuada, poca definición, entre otros. Debido a la diversidad de dispositivos y los factores ambientales es necesario corregir estos defectos para facilitar el procesamiento en las etapas posteriores y filtrar la información de la imagen. Esto se logra aplicándole a las imágenes un filtro gaussiano (provisto por la librería OpenCV) el cual permite suavizarla, distribuir uniformemente los colores y eliminar ruido, dando un efecto visual borroso y permitiendo disminuir el brillo especular introducido por la luz (figura 46).



Figura 4.46.a: Imagen de entrada



Figura 4.46.b: Resultado de preprocesamiento

Segmentación

En esta etapa se toma el resultado del preprocesamiento y se detecta la presencia o ausencia del posible objeto de interés basándose en el rango especificado en la configuración previa. El resultado es una descomposición en objetos más simples resaltados con un color que se denominará como positivo y el resto serán cambiados a otro color que se denominará como negativo. Esto se logra realizando la segmentación en dos (2) pasos, primero se detecta el posible objeto de interés y luego se resalta.

La detección se realiza con implementación de una segmentación por umbral (especificado en los rangos configurados previamente) en donde se verifica el valor de cada píxel en la imagen. Si dicho píxel está contenido en el rango especificado entonces es marcado como positivo, de no ser así será negativo. La ausencia del objeto de interés se detecta si el resultado de la segmentación por umbral es una imagen que sólo posee color negativo, en consecuencia se detiene el proceso. Es condición necesaria que el objeto de interés se encuentre en ambas imágenes y sea detectado en la etapa de segmentación para que el proceso pueda continuar.

Una vez detectado la presencia del objeto de interés y marcado como positivo, se procede a escalar el tamaño del objeto aplicando una erosión con respecto al fondo utilizando la librería OpenCV. Esto se realiza con el fin de ampliar el tamaño representativo del mismo ya que en algunos casos es posible que la muestra sea muy pequeña y se preste a confusiones o simplemente no se detecte (figura 47).

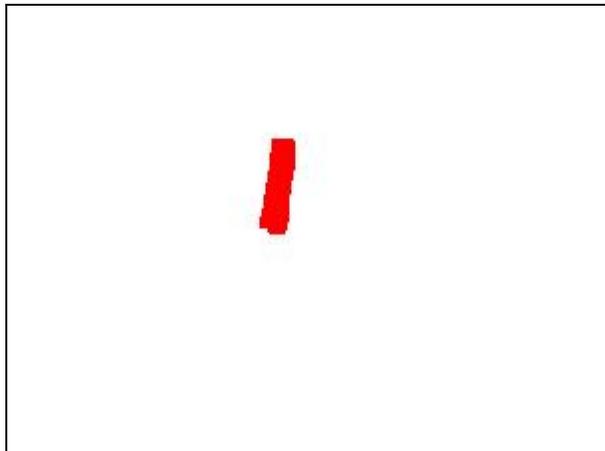


Figura 4.47: Resultado de la segmentación

Extracción de características

Luego de la segmentación se procede a obtener la información más relevante de cada objeto detectado. Las características más relevantes para este caso en específico es el área donde esta contenido el objeto, el centro del mismo y la cantidad de píxeles positivos encontrados (figura 48).

El procedimiento para obtener las características de cada objeto detectado fue implementado en la función calcularBB el cual se encarga de realizar un recorrido por toda la imagen en busca de colores positivos. Para el recorrido de la imagen se utilizó una implementación de búsqueda en profundidad. Primero se recorre la imagen buscando el color marcado como positivo en la etapa de segmentación. Luego, una vez encontrado un píxel de color positivo, se marca nuevamente y se visitan todos los píxeles adyacentes a éste. A medida que se hace el recorrido se aprovecha para calcular el tamaño, límites, cantidad de píxeles y centro del objeto enmarcado en la zona encontrada. La función implementada se realizo con una cola.

Por cada iteración del algoritmo calcula la cantidad de píxeles, máximo y mínimo valor en el eje X e Y, esto con el fin de obtener los puntos límite donde esta contenido el objeto (el cual llamaremos caja delimitadora). Por último se calcula el centro en base a la caja delimitadora. Durante la ejecución del algoritmo los píxeles que ya fueron revisados serán marcados, en consecuencia el objeto también, esto con el fin de no revisar un mismo objeto varias veces.

Para cada objeto se crea un vector de características el cual contendrá toda la información relevante encontrada en esta etapa de todos los posibles objetos encontrados.

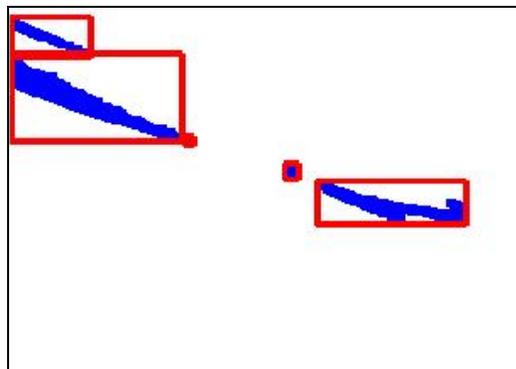


Figura 4.48: Ejemplo de varios objetos resultado de la extracción de características

Procesamiento de alto nivel

Las primeras cuatro (4) etapas mencionadas anteriormente es necesario realizarlas una vez por cada dispositivo. Es decir, es necesario adquirir la imagen del primer dispositivo y del segundo, luego aplicarle a ambas un preprocesamiento, una segmentación por umbral y extraer los vectores de características respectivos asociado a cada una de las imágenes.

En procesamiento de alto nivel se utilizan los vectores de características obtenidos de ambas imágenes y se procesan con diversas heurísticas y mecanismos matemáticos y/o geométricos para identificar una solución única por cada imagen. Luego con la ayuda de la geometría epipolar se estima la posición tridimensional del objeto de interés.

Primero se procede a eliminar los falsos positivos. Esto se implementó contando la cantidad de píxeles del color de interés de cada posible zona, escogiendo la zona más poblada (mayor cantidad de color). Luego tomando en cuenta la posición anterior del objeto (en caso de existir), se calcula la nueva posición en base a la diferencia entre la posición anterior y la actual, todo esto para lograr que el movimiento sea fluido. Es importante recalcar que este cálculo se hace por cada imagen logrando una posición única por cada una de ellas.

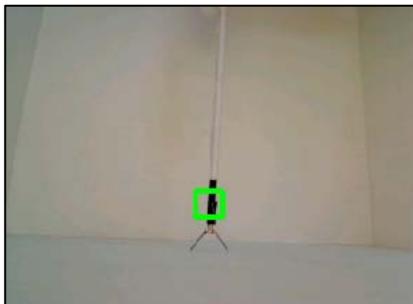


Figura 4.49: Objeto de interés identificado

Esta posición será utilizada para mostrar al usuario el estado de la detección del objeto de interés en cada una de las cámaras. También es usada para el cálculo de geometría epipolar, teniendo como resultado un sistema de ecuaciones homogéneas, el cual será resuelto por el método SVD el cual es provisto por la librería LAPACK, obteniendo así la estimación de la posición del objeto de interés en forma de una coordenada tridimensional única.

4.4.3 DESPLIEGUE TRIDIMENSIONAL

La última etapa del Sistema de Visión Artificial es el despliegue de resultados en un ambiente tridimensional simulado, donde se obtienen las coordenadas en tres dimensiones calculadas en el paso anterior y se colocan en una escena tridimensional. Este módulo se encarga de realizar una transformación entre las coordenadas producto de la geometría epipolar y la escala utilizada en la simulación de la escena.

La escena fue desarrollada en OpenGL® y consta de un mallado el cual será el área de representación del objeto, la ubicación del objeto está representada por un cubo de color verde como se muestra en la figura 50.

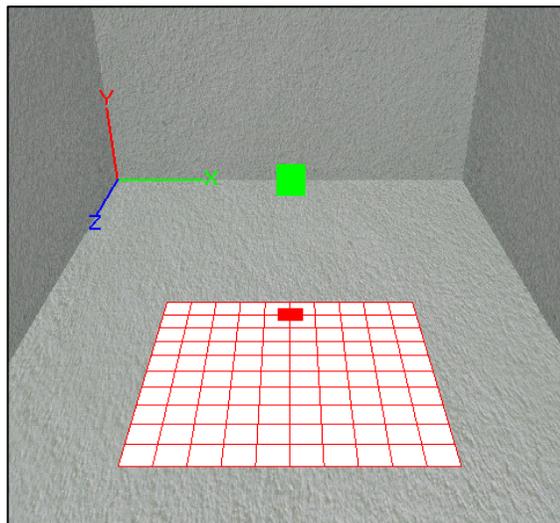


Figura 4.50: Escena tridimensional

CAPÍTULO 5: PRUEBAS Y RESULTADOS

Finalizada la etapa de desarrollo es necesario poner a prueba el sistema para evaluar rendimiento, eficiencia y precisión. En este capítulo se presentarán las pruebas realizadas en los dos ambientes antes mencionados y los resultados obtenidos.

5.1 PRUEBAS REALIZADAS

Las pruebas realizadas involucran la detección del objeto de interés en la cual se coloca el objeto en la escena y confirmar que el sistema logre su detección. Luego pruebas de movimiento, donde se coloca el objeto en la escena y se traslada, corroborando que el sistema determine el movimiento y despliegue una respuesta visual. Todas las pruebas realizadas involucran la presencia y ausencia del objeto de interés.

Se utilizaron dos (2) objetos con características distintas. Para cada ambiente se realizaron las mismas pruebas con los mismos objetos para poder compararlos. A continuación las características de los objetos utilizados así como también las pruebas realizadas.

5.1.1 PINZA LAPAROSCÓPICA

El primer objeto sobre el cual se realizaron pruebas con el sistema fue una pinza laparoscópica. Esta pinza tiene dimensiones de 30 cm de largo y 0.5 cm de diámetro (aproximadamente) con distintas marcas de colores de dimensiones 3 cm de largo y 0,5 cm de diámetro (aproximadamente) las cuales representan e identifican al objeto.

Como se muestra en (figura 51) el tamaño de la marca de color del objeto de interés es considerablemente pequeño.



Figura 5.51: Pinza laparoscópica con marca de color

5.1.2 CUBO

El segundo objeto utilizado fue un cubo de colores de dimensiones 4 cm de lado representado un objeto estándar a ser detectado (figura 52). Debido a que el volumen del color característico en el cubo es mayor con respecto al de la pinza es ideal al momento de realizar pruebas, debido a su diferencia de tamaños.

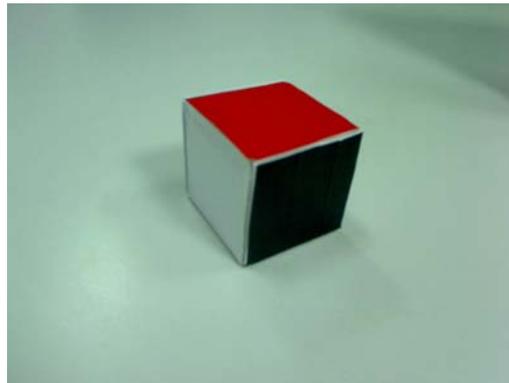


Figura 5.52: Cubo

5.2 RESULTADOS AMBIENTE CONTROLADO

El ambiente controlado resultó ideal para las pruebas realizadas, logrando una iluminación homogénea, reduciendo considerablemente la aparición de luz especular y obteniendo una representación del color estable.

La detección de ambos objetos de interés resultó exitosa, tanto en movimiento como estático, lográndose detectar de modo preciso el objeto en las imágenes capturadas por los dispositivos. En la mayoría de los casos la representación en la escena tridimensional del objeto de interés tiene correspondencia con los movimientos realizados, pero en algunas ocasiones la simulación no parece un movimiento natural.

5.3 RESULTADOS SIMULAP V-1

Utilizando una configuración similar de iluminación aplicada al ambiente controlado resultó que la luz especular dentro de la estructura es muy intensa debido al material reflexivo que la compone, en consecuencia fue necesario utilizar marcas de color blanco en los objetos para las pruebas, de forma tal que no sea afectado por la luz.

Al momento de las pruebas resultó que el SIMULAP posee un espacio reducido para la colocación de los dispositivos de captura y la iluminación, pero es tolerable permitiendo realizar las pruebas.

La detección de ambos objetos de interés resultó exitosa siempre y cuando se utilice el color blanco, en caso contrario el color calculado será objeto de alteraciones debido a las condiciones de luz cambiantes, resultando la detección solo en ciertas posiciones del objeto.

Al igual que en ambiente controlado en la mayoría de los casos la representación en la escena tridimensional del objeto de interés tiene correspondencia con los movimientos realizados, pero existen ocasiones en las cuales la simulación no parece un movimiento natural.

5.4 TIEMPO DE RESPUESTA DEL SISTEMA

Una de las métricas más importantes para el desarrollo de este trabajo es el tiempo de respuesta. Esto se debe a que es necesario, en la mayoría de los casos, realizar el proceso en tiempo real. El tiempo de respuesta (expresado en segundos y milésimas de segundo) se calculó como la suma del tiempo de realización del Proceso de Visión Artificial más el tiempo de despliegue en OpenGL®, por cada iteración.

5.4.1 TIEMPO DE REALIZACIÓN DEL PROCESO VISIÓN ARTIFICIAL

El proceso de Visión Artificial es el que amerita mayor procesamiento y tiempo, en consecuencia es necesario evaluar y analizar su tiempo de realización. Esto se logro tomando en cuenta mil (1000) iteraciones del proceso. Para fines ilustrativos, las tablas y gráficos tendrán un tamaño de veinte (20) iteraciones.

Pinza laparoscópica

Los resultados obtenidos para este caso involucran a la pinza laparoscópica y se muestra el tiempo que le toma al sistema realizar el proceso de Visión Artificial cuando la pinza está presente y cuando no.

El siguiente es el caso en el que la pinza de laparoscópica esta presente en algunas ocasiones y en otras no. La medición fue realizada en segundos y milésimas de segundo (Tabla de valores y gráfico).

Iteración	Tiempo en segundos	Milésimas de segundo
1	0.0430	43
2	0.0420	42
3	0.0410	41
4	0.0420	42
5	0.0460	46
6	0.0430	43
7	0.0430	43
8	0.0430	43
9	0.0410	41
10	0.0430	43
11	0.0370	37
12	0.0370	37
13	0.0390	39
14	0.0380	38
15	0.0390	39
16	0.0380	38
17	0.0380	38
18	0.0400	40
19	0.0400	40
20	0.0410	41

Tabla 5.4: Tiempo con pinza presente y ausente

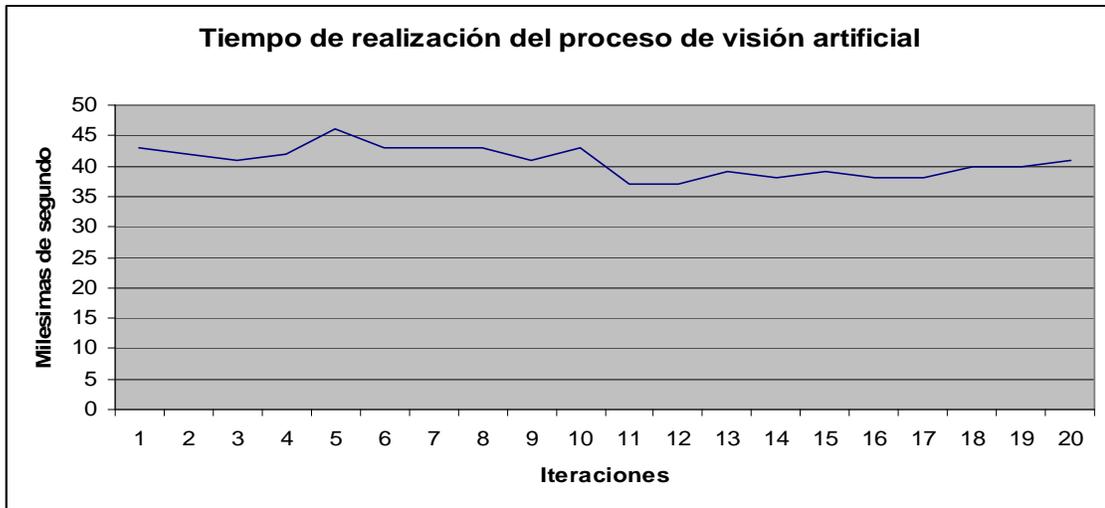


Gráfico 5.1: Tiempo de Visión Artificial con pinza presente y ausente

Tiempo promedio pinza laparoscópica

Para el cálculo del tiempo promedio, máximo y mínimo se tomaron en cuenta las mil (1000) iteraciones.

	<i>Tiempo en segundos</i>	<i>Tiempo en milésimas de segundos</i>
<i>Promedio</i>	<i>0.03785148</i>	<i>37.8514797</i>
<i>Máximo</i>	<i>0.38600000</i>	<i>386</i>
<i>Mínimo</i>	<i>0.02600000</i>	<i>26</i>

Tabla 5.5: Tiempo promedio de pinza

Cubo

Los resultados obtenidos para este caso involucran al cubo utilizado y se muestra el tiempo que le toma al sistema realizar el proceso de visión artificial cuando el cubo está presente y cuando no.

El siguiente es el caso en el cubo está presente en algunas ocasiones y en otras no. La medición fue realizada en segundos y milésimas de segundo (Tabla de valores y gráfico).

Iteración	Tiempo en segundos	Milésimas de segundo
1	0.0860	86
2	0.0860	86
3	0.0900	90
4	0.0950	95
5	0.0990	99
6	0.1040	104
7	0.1000	100
8	0.1070	107
9	0.1160	116
10	0.1220	122
11	0.1280	128
12	0.1370	137
13	0.1470	147
14	0.1520	152
15	0.1560	156
16	0.1800	180
17	0.1910	191
18	0.2000	200
19	0.1950	195
20	0.1860	186

Tabla 5.6: Tiempo con cubo presente y ausente

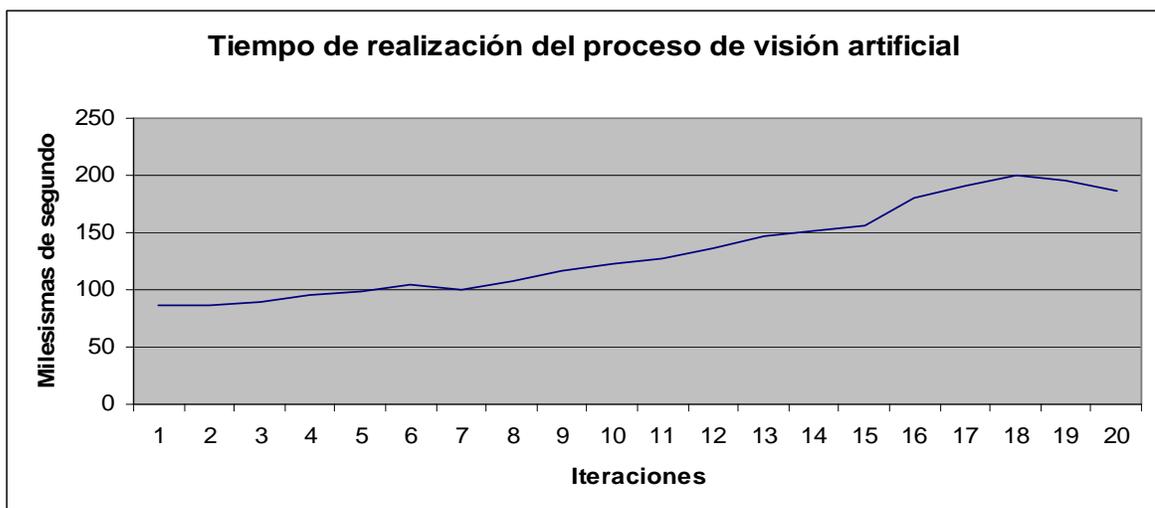


Gráfico 5.2: Tiempo de Visión Artificial con cubo presente y ausente

Tiempo promedio cubo

Para el cálculo del tiempo promedio, máximo y mínimo se tomaron en cuenta las mil (1000) iteraciones.

	<i>Tiempo en segundos</i>	<i>Tiempo en milésimas de segundos</i>
<i>Promedio</i>	<i>0.08301733</i>	<i>83.01733</i>
<i>Máximo</i>	<i>0.41000000</i>	<i>410</i>
<i>Mínimo</i>	<i>0.06800000</i>	<i>68</i>

Tabla 5.7: Tiempos del cubo

Una vez culminado el cálculo de los tiempos para ambos objetos de interés podemos notar que el tiempo de realización para el cubo es mayor al de la pinza laparoscópica, esto se debe a que el área que conforma el color característico del cubo es mas amplio que el de la pinza. En consecuencia existe más procesamiento en cada una de las etapas del proceso de Visión Artificial creando la diferencia de tiempos.

5.4.2 TIEMPO DE DESPLIEGUE TRIDIMENSIONAL

Una vez realizado el proceso de Visión Artificial es necesario desplegar los resultados en una escena tridimensional, en consecuencia el tiempo de despliegue debe ser considerado al momento de calcular el tiempo de respuesta del sistema.

En esta sección se tomaron en cuenta mil (1000) iteraciones del despliegue. Para fines ilustrativos, las tablas y gráficos tendrán un tamaño de veinte (20) iteraciones.

Iteración	Tiempo en segundos	Milésimas de segundo
1	0.0040	4
2	0.0100	10
3	0.0040	4
4	0.0130	13
5	0.0050	5
6	0.0120	12
7	0.0050	5
8	0.0000	0
9	0.0050	5
10	0.0110	11
11	0.0060	6
12	0.0130	13
13	0.0060	6
14	0.0030	3
15	0.0060	6
16	0.0120	12
17	0.0080	8
18	0.0140	14
19	0.0080	8
20	0.0000	0

Tabla 5.8: Tiempo de despliegue

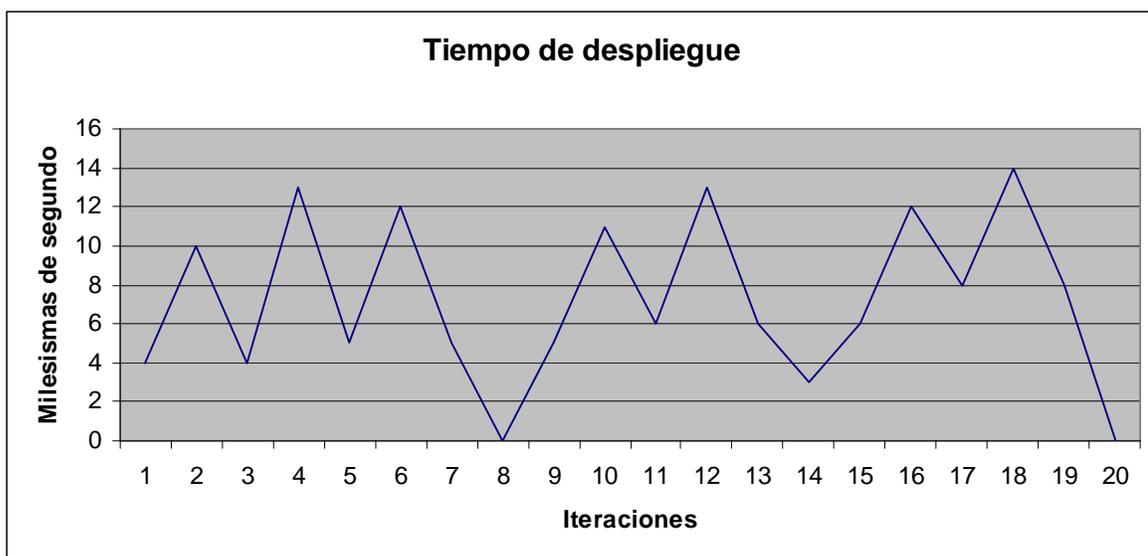


Gráfico 5.3: Tiempo de despliegue

Tiempo promedio de despliegue

Para el cálculo del tiempo promedio se tomaron mil (1000) iteraciones de ejecuciones, y se promediaron. Adicionalmente se calculó el valor máximo y mínimo entre ambos casos.

	<i>Tiempo en segundos</i>	<i>Tiempo en milésimas de segundos</i>
<i>Promedio</i>	0.00634353	6.34352765
<i>Máximo</i>	0.04900000	49
<i>Mínimo</i>	0.00000000	0

Tabla 5.9: Tiempos de despliegue

Evaluando el tiempo de despliegue podemos notar que es considerablemente menor con respecto al proceso de Visión Artificial. El tiempo promedio de despliegue es ínfimo, lo cual es ideal ya que no va a afectar considerablemente el tiempo de respuesta del sistema. Adicionalmente el mínimo tiempo de despliegue es cero ya que en ocasiones no hay detección en el proceso de Visión artificial y no es necesario actualizar el lugar de representación.

5.4.3 TIEMPO PROMEDIO DE RESPUESTA DEL SISTEMA

En base a los resultados anteriores calculamos el tiempo promedio de respuesta sumándole el tiempo promedio de realización del proceso de Visión Artificial al tiempo promedio de despliegue.

	<i>Tiempo en segundos</i>	<i>Tiempo en milésimas de segundos</i>
<i>Promedio</i>	0.06677794	66.777935
<i>Máximo</i>	0.43500000	435
<i>Mínimo</i>	0.01400000	14

Tabla 5.10: Tiempos de respuesta del sistema

El sistema tiene como datos de entrada dos imágenes (una por cada dispositivo de captura), es notable que la cantidad de imágenes que se pueden procesar están relacionadas con el desempeño del sistema. Analizando el tiempo respuesta, en el mejor de los casos es posible procesar 71 imágenes por segundo, en el peor de los casos 2 y en promedio 15.

CONCLUSIONES

Durante el desarrollo del Trabajo Especial de Grado fue posible implementar un sistema de Visión Artificial que cumple con los objetivos planteados, demostrando así que es posible detectar y ubicar en un espacio tridimensional un objeto en base a su color o marcas de color.

Específicamente, al evaluar y analizar los resultados obtenidos se obtuvo lo siguiente:

- La detección y seguimiento del objeto en ambos ambientes se logra de manera satisfactoria, permitiendo obtener una buena ubicación de cada imagen.
- Mediante la geometría epipolar se logró estimar la posición en tres dimensiones del objeto de interés.
- El módulo de calibración creado es capaz de deducir y calcular la matriz de proyección de cualquier dispositivo de captura logrando que la aplicación sea independiente del dispositivo a utilizar.
- El software realizado se desarrolló de forma modular permitiendo que sus partes sean reutilizables, así como también cualquier tipo de adición o mejora se puede realizar sin tener que modificar la estructura del mismo.
- Debido al diseño y funcionamiento del sistema es posible adaptarlo para detectar múltiples objetos de interés.
- Para evitar corregir errores introducidos por el dispositivo de captura en la etapa de preprocesamiento es necesario que el dispositivo utilizado provea imágenes de alta calidad y buena resolución.
- El color del objeto de interés resultó ser de vital importancia para el correcto funcionamiento del sistema, por lo tanto es necesario que al momento de la selección del color del objeto de interés obtener la mayor área de muestra posible que caracterice el color a detectar para evitar que algunas tonalidades no sean tomadas en cuenta.
- La selección del color característico del objeto de interés debe realizarse en base a la escena en donde se desarrolle el sistema, preferiblemente el color debe tener de poca o nula ocurrencia en el mismo y de color opuesto al fondo.

- Debido a los movimientos realizados por el objeto de interés dentro de la escena el dispositivo de captura tiende a desenfocar y enfocar constantemente, es recomendable ajustar el foco de la cámara de manera manual para evitar distorsiones en la imagen que afecten su procesamiento.
- En general en ambos ambientes, los movimientos rápidos no son detectados ya que las cámaras tienden a desenfocarse distorsionando la imagen. Es recomendable hacer movimientos lentos del objeto de interés para evitar desenfocos de las cámaras que distorsionen las imágenes y se pierda la detección del objeto.

TRABAJOS FUTUROS

- Utilizar un método alternativo a la geometría epipolar en caso de necesitarse mayor precisión en el momento de la simulación.
- Existen movimientos naturales del objeto de interés como las rotaciones o de apertura y cierre (en el caso de tijeras o pinzas) que son imperceptibles por el entorno de Visión Artificial, es recomendable la adición de métodos capaces de detectar estos movimientos y representarlos en el ambiente tridimensional.
- Desarrollar un método de detección más robusto el cual permita detectar un objeto de interés sin necesidad de usar marca de colores.
- Determinar la orientación del objeto de interés ubicado en la escena.
- Aplicar técnicas de procesamiento digital de imágenes que permitan sustituir la iluminación de la escena.

REFERENCIAS

- [1] *Babylon online dictionary*. Consultado el 08 de marzo de 2008 de la World Wide Web: http://www.babylon.com/definition/Im%C3%A1genes_m%C3%A9dicas/Spanish.
- [2] Climent, J. y Marés, P. (2004) *Sistema de Seguimiento en Tiempo Real para Intervenciones Quirúrgicas Asistidas*. Universidad Politécnica de Catalunya.
- [3] Duda, R. y Hart, P. (1972, enero). *Use of the Hough Transformation to Detect Lines and Curves in Pictures*. Publicado en el Comm. ACM, Vol 15, No. pp. 11-15.
- [4] Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations*. 3rd ed., Johns Hopkins University Press, Baltimore. ISBN 0-8018-5414-8.
- [5] González, A., Martínez, F., Pernía, A., Alba, F., Castejón, M., Odieres, J. y Vergara, E. (2006). *Técnicas y Algoritmos Básicos de Visión Artificial*. Universidad de la Rioja, Servicio de publicaciones.
- [6] Hartley, R. y Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press.
- [7] Holden, E-J. y Owens, R., (2001). *Visual Sign Language Recognition*. Lecture Notes in Computer Science. Australia.
- [8] Martínez, J., Arrue, B., Ollero, A., Merino, L. y Gómez, F. (2004, enero). *Computer Vision Techniques for Forest Fire Perception*. Elsevier. Image and Vision Computing 26.
- [9] Mery, D. (2002, enero). *Visión Artificial*. Tesis de Magister de Ingeniería Informática, Universidad de Santiago de Chile.
- [10] *Microsoft Corporation*. Consultado el 23 de octubre de 2008 de la World Wide Web: [http://office.microsoft.com/en-us/word/default.aspx.\(2007\)](http://office.microsoft.com/en-us/word/default.aspx.(2007)).
- [11] Muñoz, V., Gómez, G., Fernández, J., García, I., Perez, C. y Azouaghe, M. (2001). *Diseño y Control de un Asistente Robótico para Cirugía Laparoscópica*. Instituto de Automática y Robótica Avanzada de Andalucía, Universidad de Málaga.
- [12] Nalwa, V. S. (1993) *A Guided Tour to Computer Vision*. Addison – Wesley.
- [13] *LAPACK*. Consultado el 23 de octubre de 2008 de la World Wide Web: <http://www.netlib.org/lapack/>.
- [14] Olague, G. (2007). *Evolutionary Computer Vision*. GECCO'07, London, England, United Kingdom. ACM 978-1-59593-698-1.
- [15] *OpenOffice*. Consultado el 23 de octubre de 2008 de la World Wide Web: <http://www.openoffice.org/index.html>. (2002).
- [16] *OpenCV*. Consultado el 23 de octubre de 2008 de la World Wide Web: <http://sourceforge.net/projects/opencvlibrary/>. (2006).

- [17] Ortiz, F. G. (2002). Procesamiento morfológico de imágenes en color: Aplicación a la reconstrucción geodésica. Alicante: Biblioteca Virtual Miguel de Cervantes.
- [18] Pearson, D. (1991). *Image Processing*. Gran Bretaña: MacGraw-Hill.
- [19] Pearl, J. (1983). *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. New York, Addison-Wesley.
- [20] Russ, J. C. (1998). *The Image Processing Handbook*. CRC Press 3rd edition.
- [21] *School For Champions*. Consultado el 15 de marzo de 2008 de la World Wide Web: <http://www.school-for-champions.com>.
- [22] Sedgewick, R. (2002). *Algorithms in c++*. Part 1-5. Addison Wesley.
- [23] Strizinec, G. (2005). *Fotografía Digital*. Alfaomega - Ra-ma Editorial.
- [24] *Trolltech*. Consultado el 23 de octubre de 2008 de la World Wide Web: <http://trolltech.com/products/qt/>.(2008).
- [25] Urbina, B., Coto, E., Rodríguez, O., Miquilarena, R. y Cerrolaza, M. (2005, Septiembre). *Laparos: Computer Assisted Laparoscopic Surgery Training*. Universidad Central de Venezuela.
- [26] Vezhnevets, V., Velizhev, A. (2005). *GML C++ Camera Calibration Toolbox*. Graphics & media lab, The laboratory of computer graphics at dep. Michigan State University.
- [27] *Walking Dead Project*. Consultado el 22 de septiembre 2008 de la World Wide Web: http://wiki.vislab.usyd.edu.au/moin.cgi/Walking_Dead/
- [28] Woo, M., Neider, J., Davis, T. (1997). *OpenGL® Programming Guide*. Addison-Wesley Professional. 3ra edición. 1999.

ANEXOS

ANEXO A: RECOMENDACIONES PARA LA CREACIÓN DE LAS IMÁGENES DE CALIBRACIÓN

Las imágenes tomadas deben poseer características especiales las cuales permitan al Entorno de Visión Artificial detectar el patrón y en base a este deducir los valores de la matriz de proyección del dispositivo de captura utilizado. Para este caso se utilizó un tablero de ajedrez estándar 8x8, con el color de los cuadros blancos y negros.

- Deben existir espacios en blanco de separación entre el borde de la imagen y el borde del objeto de interés (figura 53).

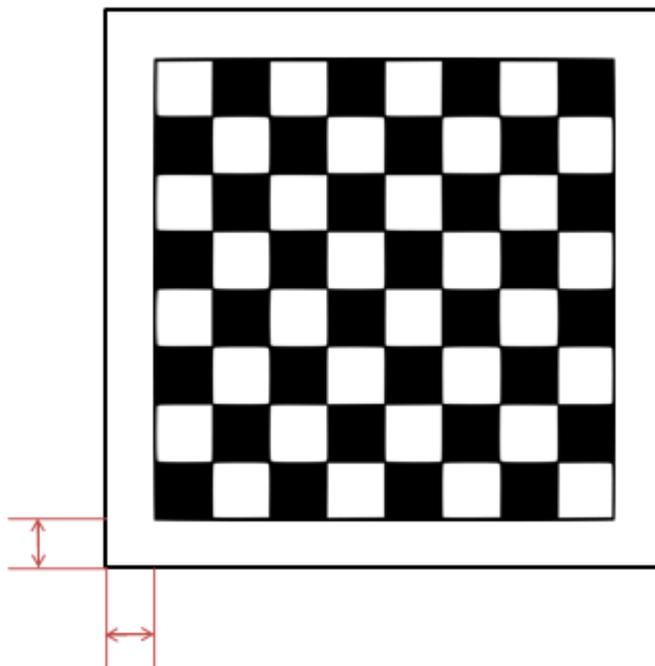


Figura 53: Patrón de calibración

- Todos los cuadros deben ser claramente visibles (no pueden estar ocluidos).
- Es recomendable el uso de un trípode, para evitar distorsiones en la imagen.
- Se deben utilizar al menos 25 o más imágenes para obtener los mejores resultados.
- El tamaño recomendado de los cuadros deber se entre tres y cinco centímetros (3-5 cm).

- El patrón de calibración (tablero de ajedrez) debe estar localizado en todos los lugares posibles donde se estime que el objeto de interés estará presente (figura 54).

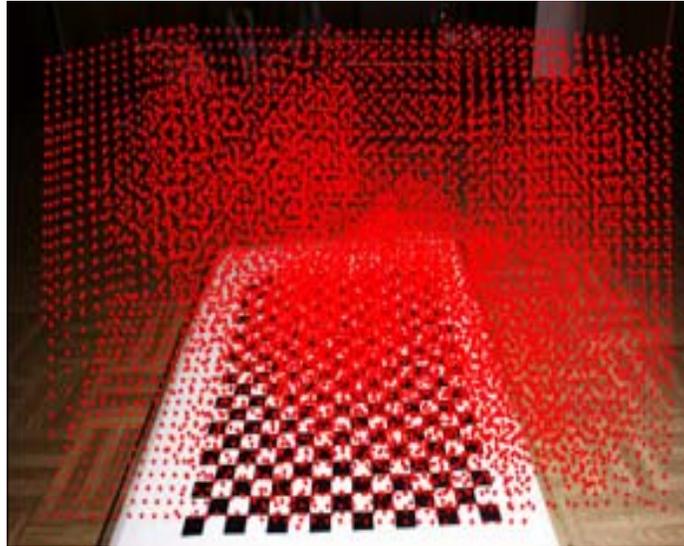


Figura 54: Posibles ubicaciones del tablero (múltiples tomas)

- El tablero de ajedrez debe ser plano.
- Se deben tomar las fotos desde las posiciones norte, noreste, este, sureste, sur, suroeste, oeste, noroeste y adicionalmente desde la parte superior del tablero (figura 55).

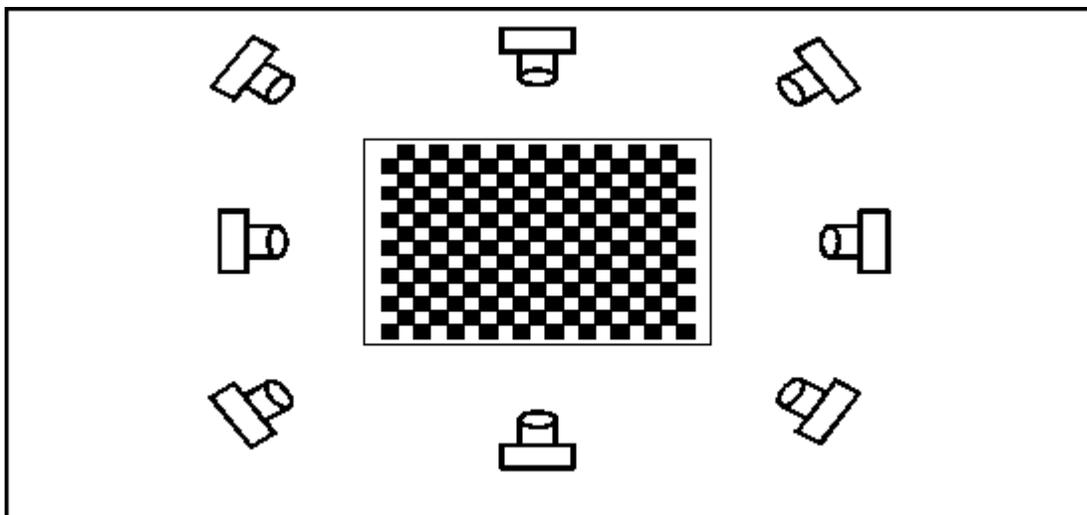


Figura 55: Ubicaciones de la cámara

- El ángulo de inclinación de la cámara debe ser constante (figura 56).

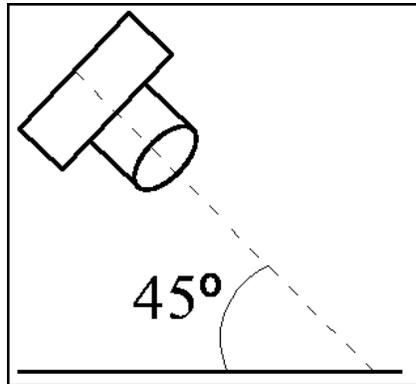


Figura 56: Ángulo de inclinación (ejemplo de 45°)

- Las fotos deben ser capturas desde las posiciones especificadas anteriormente y rotando la cámara de la manera especificada a continuación (figura 57).

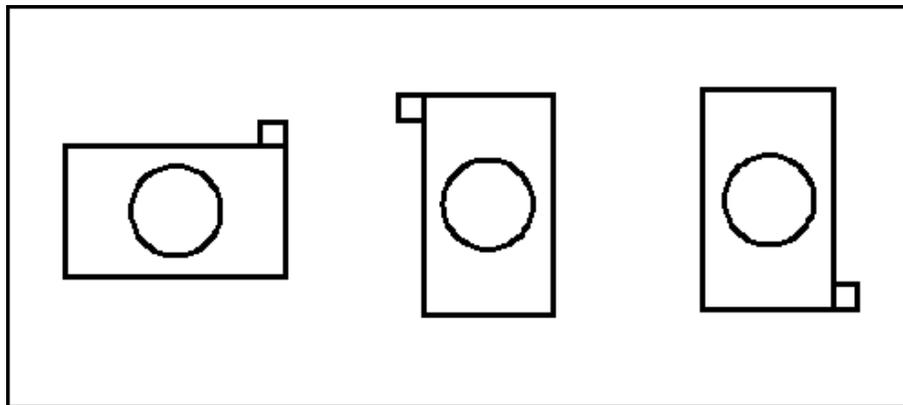


Figura 57: Rotaciones de la cámara

ANEXO B: INSTRUCTIVO DE INTERFACES

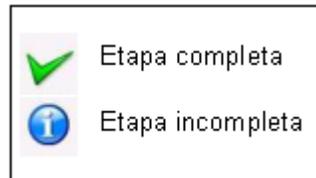
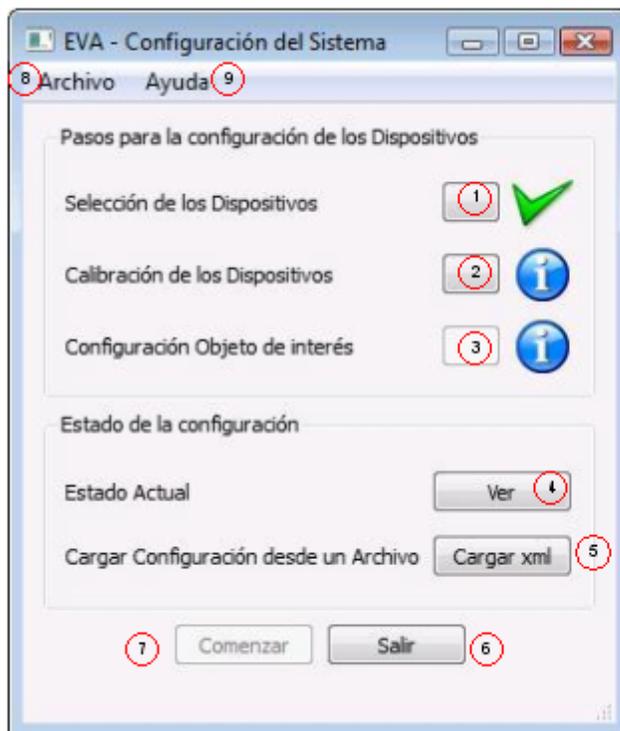
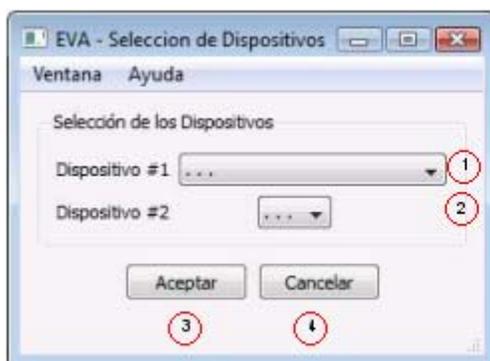


Figura 58: Notación



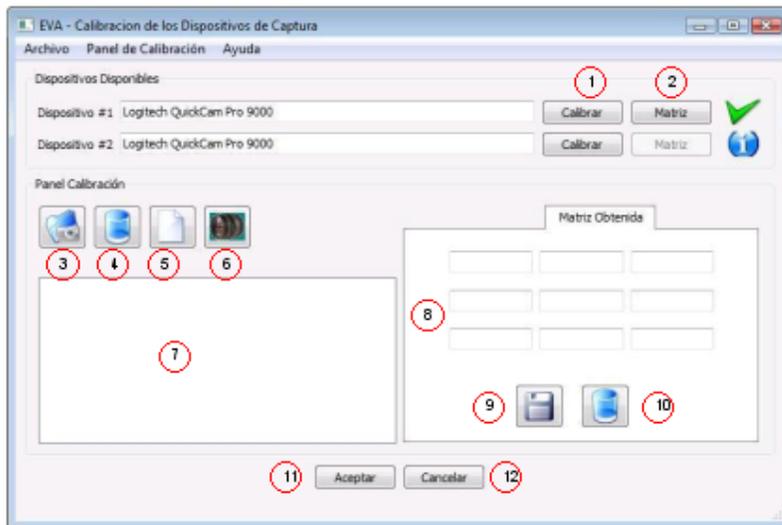
- 1.- Seleccionar Dispositivos de Captura.
- 2.- Calibrar los dispositivos escogidos.
- 3.- Configuración de los colores del objeto a detectar.
- 4.- Ver configuración actual.
- 5.- Cargar configuración desde un archivo XML.
- 6.- Salir del Sistema.
- 7.- Comenzar Visión Artificial (Sólo disponible cuando las tres etapas de configuración estén completas).
- 8.- Menú (Se puede acceder a todas las funcionalidades).
- 9.- Ayuda para guiar al usuario.

Figura 59: Ventana de configuración



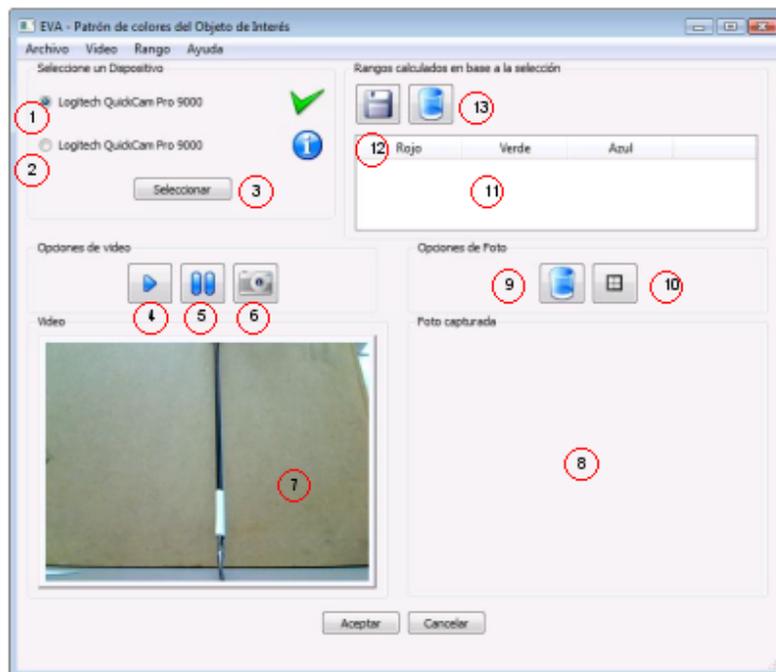
- 1.- Lista de dispositivos de captura disponibles #1.
- 2.- Lista de dispositivos de captura disponibles #2.
- 3.- Aceptar selección.
- 4.- Cancelar selección.

Figura 60: Ventana de selección de dispositivos



- 1.- Calibrar dispositivo.
- 2.- Visualizar matriz del dispositivo.
- 3.- Buscar imágenes de calibración.
- 4.- Borrar imagen de calibración seleccionada.
- 5.- Borrar todas las imágenes de calibración.
- 6.- Calibrar.
- 7.- Lista de imágenes.
- 8.- Matriz del dispositivo obtenida.
- 9.- Guardar matriz obtenida.
- 10.- Borrar matriz obtenida.
- 11.- Aceptar matrices obtenidas.
- 12.- Cancelar matrices obtenidas.

Figura 61: Ventana de calibración



- 1.- Dispositivo habilitado.
- 2.- Dispositivo deshabilitado.
- 3.- Preparar dispositivo habilitado.
- 4.- Comenzar video del dispositivo habilitado.
- 5.- Detener video del dispositivo habilitado.
- 6.- Tomar foto del video.
- 7.- Espacio para video.
- 8.- Espacio para foto.
- 9.- Borrar foto.
- 10.- Calcular rangos del color.
- 11.- Rangos calculados.
- 12.- Guardar rango.
- 13.- Borrar rango.

Figura 62: Ventana de objeto de interés

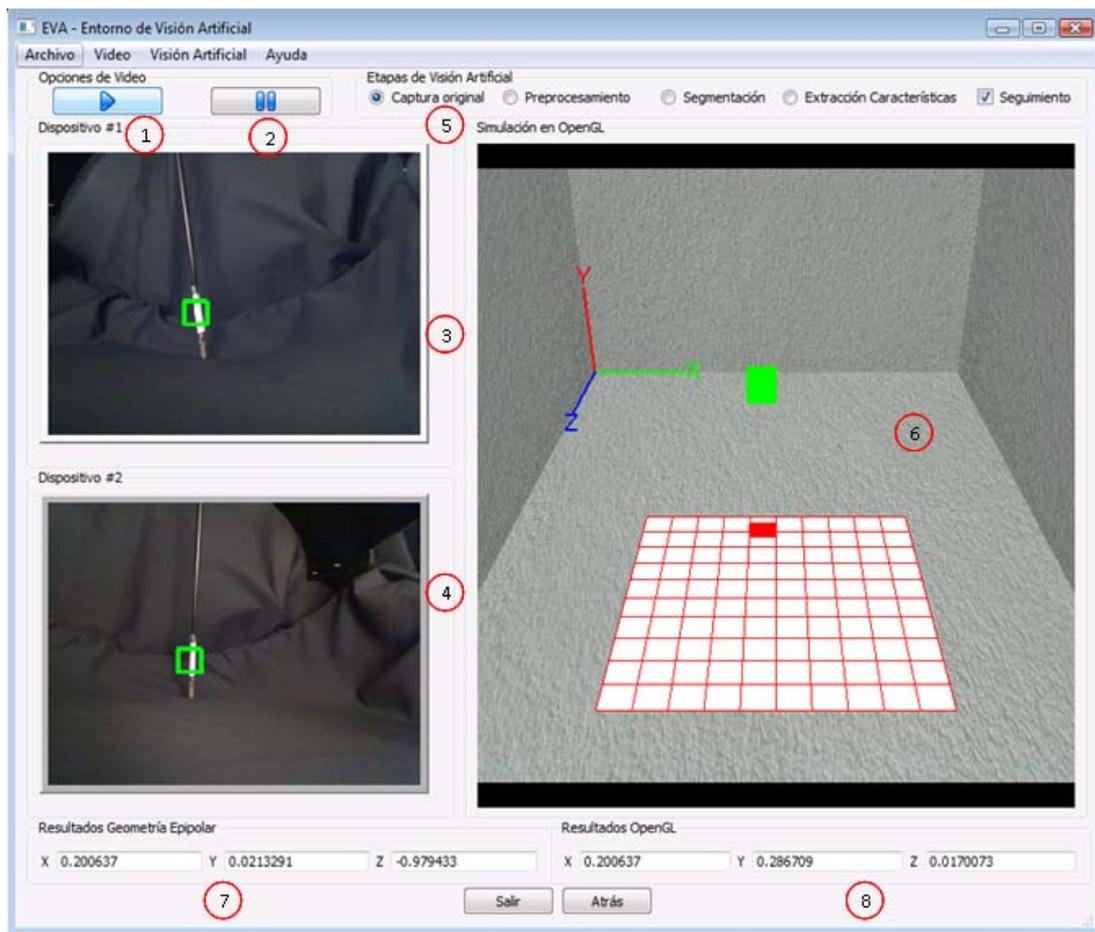
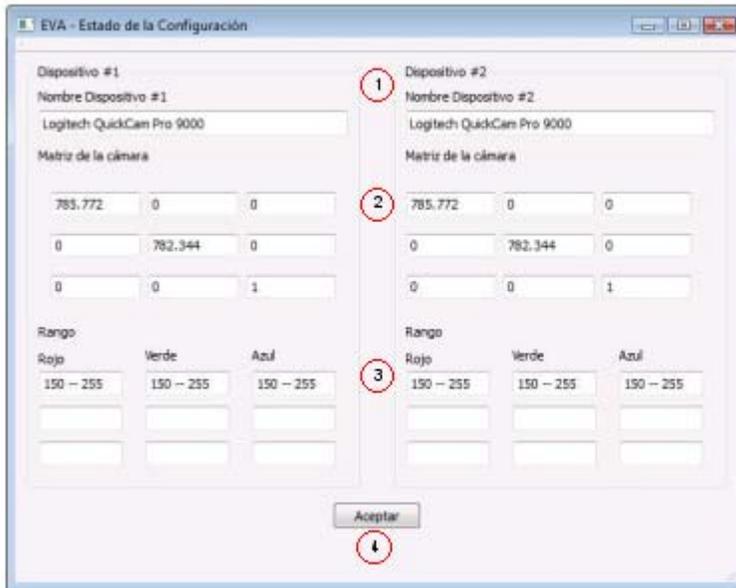


Figura 63: Ventana consultar configuración actual

- 1.- Comenzar el video.
- 2.- Detener video.
- 3.- Visual de la primera cámara.
- 4.- Visual de la segunda cámara.
- 5.- Panel de opciones para el video.
- 6.- Representación en tres dimensiones.
- 7.- Resultados de la geometría epipolar.
- 8.- Resultados del despliegue.



- 1.- Dispositivos de captura seleccionados.
- 2.- Matrices de los dispositivos.
- 3.- Rango de colores a detectar.
- 4.- Volver.

Figura 64: Ventana consultar configuración actual