



**ESPE**  
**UNIVERSIDAD DE LAS FUERZAS ARMADAS**  
**INNOVACIÓN PARA LA EXCELENCIA**

**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y  
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y  
CONTROL**

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE:**

**INGENIERO EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL**

**TEMA: “DESARROLLO DE UN SISTEMA DE CALIFICACIÓN PARA UN  
POLÍGONO VIRTUAL DE TIRO BASADO EN VISIÓN POR  
COMPUTADOR”**

**AUTOR: CASTRO SILVA, CRISTIAN PATRICIO**

**DIRECTOR: DR. AGUILAR CASTILLO, WILBERT G.**

**SANGOLQUÍ - 2018**



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y  
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA,  
AUTOMATIZACIÓN Y CONTROL**

**CERTIFICACIÓN**

Certifico que el trabajo de titulación, “**DESARROLLO DE UN SISTEMA DE CALIFICACIÓN PARA UN POLÍGONO VIRTUAL DE TIRO BASADO EN VISIÓN POR COMPUTADOR**” fue realizado por el señor **CASTRO SILVA CRISTIAN PATRICIO** el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, Noviembre de 2018



Ing. Wilbert Geovanny Aguilar Castillo PhD.

C.C: 0703844696



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y  
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA,  
AUTOMATIZACIÓN Y CONTROL**

**AUTORÍA DE RESPONSABILIDAD**

Yo, **CASTRO SILVA, CRISTIAN PATRICIO**, declaro que el contenido, ideas y criterios del trabajo de titulación: “**DESARROLLO DE UN SISTEMA DE CALIFICACIÓN PARA UN POLÍGONO VIRTUAL DE TIRO BASADO EN VISIÓN POR COMPUTADOR**” es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, Noviembre de 2018

A handwritten signature in blue ink, which appears to be 'C. CASTRO S', is written over a horizontal line. The signature is stylized and cursive.

Cristian Patricio Castro Silva

C.C. 1722046198



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y  
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA,  
AUTOMATIZACIÓN Y CONTROL**

**AUTORIZACIÓN**

Yo, **Castro Silva, Cristian Patricio** autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **DESARROLLO DE UN SISTEMA DE CALIFICACIÓN PARA UN POLÍGONO VIRTUAL DE TIRO BASADO EN VISIÓN POR COMPUTADOR** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, Noviembre de 2018



Cristian Patricio Castro Silva

C.C. 1722046198

## DEDICATORIA

Dedico este trabajo a mis padres, Jenny y German, que a lo largo de mi vida me han guiado con buenas enseñanzas e inculcado valores para formarme como una persona de bien. Por su apoyo constante en los buenos, pero sobre todo en los malos momentos, su paciencia y sobre todo el amor que siempre me han brindado.

Adicionalmente dedico este trabajo a mis hermanos, Andrés y Diana, quienes siempre han representado un ejemplo a seguir, me han respaldado a lo largo de mi vida y me guían con sus sabios consejos.

Familia, esto es para ustedes.

*Cristian Patricio Castro Silva.*

## **AGRADECIMIENTO**

Agradezco a mis padres por su arduo esfuerzo, para que, junto a mis hermanos, formarnos como profesionales y personas de bien. Por su apoyo incondicional en cada paso que doy y sobre todo por su paciencia hacia mí.

A mis hermanos, con quienes he compartido cada triunfo y derrota, pero siempre han sido una guía para mi formación tanto académica como profesional.

Al Centro de Investigación Científica y Tecnológica del Ejército (CICTE), por permitirme realizar mi trabajo en su institución, además por brindarme los conocimientos, materiales y personal capacitado para finalizar este proyecto.

Al Dr. Wilbert Aguilar, quien además de ser mi tutor, se ha convertido en un amigo y colega de investigación.

*Cristian Patricio Castro Silva.*

## ÍNDICE DE CONTENIDOS

**CARÁTULA**

**CERTIFICADO DEL DIRECTOR.....i**

**AUTORÍA DE RESPONSABILIDAD ..... ii**

**AUTORIZACIÓN..... iii**

**DEDICATORIA .....iv**

**AGRADECIMIENTO ..... v**

**ÍNDICE DE CONTENIDOS .....vi**

**ÍNDICE DE TABLAS..... x**

**ÍNDICE DE FIGURAS .....xi**

**RESUMEN.....xiv**

**ABSTRACT .....xv**

**CAPÍTULO I .....1**

**GENERALIDADES .....1**

1.1 Antecedentes ..... 1

1.2 Justificación e importancia..... 6

1.3 Alcance del proyecto..... 7

1.4 Objetivos ..... 10

1.4.1 Objetivo General..... 10

1.4.2 Objetivos Específicos .....	11
1.5 Descripción general del proyecto de investigación.....	11
<b>CAPÍTULO II.....</b>	<b>13</b>
<b>MARCO TEÓRICO .....</b>	<b>13</b>
2.1 Polígono de tiro .....	13
2.2 Obtención de imágenes .....	15
2.2.1 Modelo de Pin-Hole.....	16
2.2.2 Espacios de colores .....	19
2.3 Homografía .....	23
2.4 Caracterización de una imagen .....	24
2.4.1 Algoritmos de detección y descripción de puntos de interés.....	26
<b>CAPÍTULO III .....</b>	<b>31</b>
<b>GENERALIDADES DEL SISTEMA.....</b>	<b>31</b>
3.1 Especificaciones de hardware .....	31
3.1.1 Proyector de video .....	31
3.1.2 Cámara web .....	32
3.1.3 Disparador Laser.....	33
3.1.4 Estación de trabajo.....	33
3.2 Especificaciones de software .....	34
3.2.1 Instalación de librerías OpenCV en Unity.....	36



3.2.2 Extracción de imágenes de cámara web .....	37
3.3 Especificaciones físicas del sistema.....	39
3.4 Diagrama de bloques del sistema.....	41
<b>CAPÍTULO IV .....</b>	<b>43</b>
<b>CALIBRACIÓN DE LA CÁMARA.....</b>	<b>43</b>
4.1 Algoritmo para detección del área de disparo.....	43
4.2 Detección del área de disparo .....	45
4.2.1 Detección y descripción de los puntos de interés .....	46
4.2.2 Emparejamiento entre imágenes .....	48
4.2.3 Comparación de rendimiento de algoritmos implementados .....	51
<b>CAPÍTULO V.....</b>	<b>58</b>
<b>DETECCIÓN DEL DISPARO LÁSER .....</b>	<b>58</b>
5.1 Consideraciones .....	58
5.2 Algoritmo de detección de laser.....	60
5.3 Extracción del área de disparo .....	61
5.4 Conversión de espacio de colores .....	62
5.5 Detección de blobs .....	63
5.6 Virtualización del disparo .....	68
5.7 Pruebas de funcionamiento del algoritmo.....	69
<b>CAPÍTULO VI .....</b>	<b>73</b>

<b>DESARROLLO DEL SISTEMA DE CALIFICACIÓN.....</b>	<b>73</b>
6.1 Implementación de entornos virtuales .....	73
6.2 Implementación del software del polígono virtual de tiro .....	75
6.3 Consideraciones del uso del software del polígono virtual de tiro .....	80
<b>CAPÍTULO VII.....</b>	<b>82</b>
<b>CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>82</b>
7.1 Conclusiones .....	82
7.2 Recomendaciones.....	84
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>86</b>

## ÍNDICE DE TABLAS

<b>Tabla 1</b> <i>Características de la estación de trabajo</i> .....	<b>34</b>
<b>Tabla 2</b> <i>Comparación de rendimiento de algoritmos de detección de puntos de interés</i> .....	<b>47</b>
<b>Tabla 3</b> <i>Escenarios de iluminación propuestos</i> .....	<b>53</b>
<b>Tabla 4</b> <i>Resultados de algoritmos con la cámara se encuentra en la posición 1.</i> .....	<b>55</b>
<b>Tabla 5</b> <i>Resultados de algoritmos con la cámara se encuentra en la posición 2.</i> .....	<b>55</b>
<b>Tabla 6</b> <i>Resultados de algoritmos con la cámara se encuentra en la posición 3.</i> .....	<b>56</b>
<b>Tabla 7</b> <i>Parámetros para la detección de blobs</i> .....	<b>64</b>
<b>Tabla 8</b> <i>Parámetros seleccionados para detección del disparo</i> .....	<b>65</b>
<b>Tabla 9</b> <i>Resultados de pruebas de funcionamiento en iluminación baja</i> .....	<b>71</b>
<b>Tabla 10</b> <i>Resultados de pruebas de funcionamiento en iluminación media</i> .....	<b>71</b>
<b>Tabla 11</b> <i>Resultados de pruebas de funcionamiento en iluminación alta</i> .....	<b>72</b>
<b>Tabla 12</b> <i>Registro de calificaciones en Microsoft Excel</i> .....	<b>79</b>

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Esquema general del polígono virtual .....	<b>8</b>
<b>Figura 2.</b> Polígono de tiro en interiores.....	<b>14</b>
<b>Figura 3.</b> Polígono de tiro en exteriores .....	<b>14</b>
<b>Figura 4.</b> Representación de una imagen mediante una matriz.....	<b>15</b>
<b>Figura 5.</b> Modelo Pin-Hole .....	<b>17</b>
<b>Figura 6.</b> Funcionamiento de modelo Pin-hole .....	<b>18</b>
<b>Figura 7.</b> Representación del modelo RGB.....	<b>20</b>
<b>Figura 8.</b> Representación modelo HSV .....	<b>21</b>
<b>Figura 9.</b> Representación modelo HSI de colores .....	<b>22</b>
<b>Figura 10.</b> Representación gráfica del funcionamiento de la matriz de Homografía .....	<b>24</b>
<b>Figura 11.</b> Caracterización de una imagen de forma Global y Local.....	<b>25</b>
<b>Figura 12.</b> Proceso de detección de puntos de interés mediante el algoritmo SIFT .....	<b>27</b>
<b>Figura 13.</b> Proceso de detección de puntos de interés mediante el algoritmo SURF.....	<b>28</b>
<b>Figura 14.</b> Proceso de detección de puntos de interés mediante el algoritmo FAST.....	<b>30</b>
<b>Figura 15.</b> Proyector Epson LCD EB-X04 .....	<b>32</b>
<b>Figura 16.</b> Cámara web Logitech c920 HD .....	<b>32</b>
<b>Figura 17.</b> Logotipo del software Unity.....	<b>35</b>
<b>Figura 18.</b> Logotipo de OpenCV .....	<b>35</b>

<b>Figura 19.</b> Logotipo de librería OpenCV para Unity en Assets Store .....	<b>36</b>
<b>Figura 20.</b> Importación de OpenCV y archivos generados .....	<b>37</b>
<b>Figura 21.</b> Imagen extraída desde Unity mediante cámara web .....	<b>38</b>
<b>Figura 22.</b> Procesamiento de imagen en Unity .....	<b>39</b>
<b>Figura 23.</b> Plano del laboratorio de simulación .....	<b>40</b>
<b>Figura 24.</b> Diagrama de instalación del sistema.....	<b>41</b>
<b>Figura 25.</b> Diagrama de bloques del sistema .....	<b>42</b>
<b>Figura 26.</b> Algoritmo para detección de la zona de disparo.....	<b>44</b>
<b>Figura 27.</b> Imagen utilizada para la detección de la zona de interés (target) .....	<b>45</b>
<b>Figura 28.</b> Algoritmos de detección de características.....	<b>48</b>
<b>Figura 29.</b> Emparejamiento de imágenes .....	<b>50</b>
<b>Figura 30.</b> Detección del área de disparo .....	<b>51</b>
<b>Figura 31.</b> Escenarios para pruebas de funcionamiento de algoritmos implementados .....	<b>52</b>
<b>Figura 32.</b> Precisión de los algoritmos implementados .....	<b>57</b>
<b>Figura 33.</b> Ciclos de operación de la cámara y laser.....	<b>59</b>
<b>Figura 34.</b> Extracción del área de disparo.....	<b>62</b>
<b>Figura 35.</b> Espacio de colores HSV .....	<b>63</b>
<b>Figura 36.</b> Aplicación de filtros para detección de blobs.....	<b>67</b>
<b>Figura 37.</b> Reconocimiento del disparo laser .....	<b>68</b>
<b>Figura 38.</b> Virtualización del disparo laser .....	<b>69</b>

<b>Figura 39.</b> Imagen proyectada para pruebas de funcionamiento.....	<b>70</b>
<b>Figura 40.</b> Escenarios desarrollados para el polígono virtual de tiro .....	<b>73</b>
<b>Figura 41.</b> Siluetas desarrolladas para el polígono virtual de tiro.....	<b>74</b>
<b>Figura 42.</b> Menú de configuración de la cámara.....	<b>75</b>
<b>Figura 43.</b> Menú de configuración de la práctica de tiro .....	<b>76</b>
<b>Figura 44.</b> Selección de siluetas para escenarios virtuales.....	<b>77</b>
<b>Figura 45.</b> Calificación en tiempo real de disparos laser .....	<b>78</b>
<b>Figura 46.</b> Registro de calificación .....	<b>79</b>

## **RESUMEN**

El presente trabajo de investigación tiene como objetivo desarrollar un sistema de calificación para un polígono virtual de tiro basado en visión por computador. El principal recurso del sistema es una cámara para receptar toda la información visual, adicionalmente los disparos se realizarán mediante un disparador laser sobre una proyección con siluetas virtuales y se asignara una calificación dependiendo el objetivo alcanzado. Se realiza la detección, descripción y emparejamiento de puntos de interés con el propósito de estimar la ubicación de un target generado por un proyector y establecerlo como zona de disparo. Los algoritmos de SIFT, SURF, ORB y combinaciones entre ellos, se ponen a prueba para determinar la precisión de su funcionamiento. En la segunda etapa se realiza la virtualización del disparo, mediante el cambio en espacio de colores y la detección de blobs se determina las coordenadas del láser en las imágenes adquiridas por la cámara. Finalmente se aplica transformaciones geométricas para colocar el haz laser en la posición correcta dentro de un entorno virtual. El sistema ha sido sometido a pruebas con variación en la iluminación del ambiente controlado y cambio en la posición de la cámara, presentando un correcto desempeño a pesar de las perturbaciones.

### **PALABRAS CLAVE:**

- **PUNTOS DE INTERES**
- **DETECION DE BLOBS**
- **HOMOGRAFIA**
- **TRANSFORMACION GEOMETRICA**

## **ABSTRACT**

The objective of this research is to develop a qualification system for a virtual shooting range based on computer vision. The main resource of this system is a camera to receive all the visual information, in addition, the shots will be made by a laser trigger on a projection with virtual targets and a rating will be assigned depending on the objective achieved. The purpose of the detection, description and matching of points of interest is to estimate the location of a target generated by a projector and establish it as a firing zone. SIFT, SURF, ORB, ORB-SURF and SURF-ORB algorithms are tested to determine the accuracy of their operation. The second stage focuses on the virtualization of the shot, through the change in color space and the detection of blobs determines the coordinates of the laser in the images acquired by the camera. Finally, geometric transformations are applied to place the laser beam in the correct position within a virtual environment. The system has been subjected to tests with variation in the lighting of the controlled environment and change in the position of the camera, presenting a correct performance despite the disturbances.

### **KEYWORDS:**

- **FEATURE DETECTION**
- **FEATURE DESCRIPTION**
- **BLOB DETECTION**
- **HOMOGRAPHY**
- **GEOMETRIC TRANSFORMATION**



## CAPÍTULO I

En este capítulo se presenta una introducción sobre este proyecto de investigación, se analizan antecedentes relacionados con visión artificial, desarrollo de entornos virtuales y productos comerciales relacionados al tema; se presenta también el alcance del proyecto junto con las etapas de este y los objetivos que se han planteado cumplir.

### GENERALIDADES

#### 1.1 Antecedentes

Un polígono de tiro, o campo de tiro, es una instalación especializada y diseñada para practicar con armas de fuego o arquería, estos pueden estar ubicados en interiores o exteriores.

El polígono de tiro cuenta con los siguientes elementos básicos:

- Punto de disparo: lugar donde se coloca la persona para disparar
- Targets: siluetas de diferentes formas que son el objetivo a alcanzar por el disparo.
- Paredón: área situada detrás de los targets para contener los disparos.
- Banderas de viento: funcionan para estimar la velocidad del viento

Dentro de las Fuerzas Armadas y la Policía se emplean polígonos de tiro para capacitar a nuevo personal en el uso de armamento mediante targets creados bajo estándares propiamente militares actuales e históricos. Adicionalmente se han planteado ejercicios que permiten desarrollar destrezas como: tiro con una mano, tiro en movimiento, trabajo en equipo, entre otros.

Con el desarrollo computacional actual, y su aplicación en el ámbito de la seguridad y defensa (Andrea, Byron, Jorge, Inti, & Aguilar, 2018) (Basantes, y otros, 2018) (Jara-Olmedo A. , Medina-Pazmiño, Tozer, Aguilar, & Pardo, 2018) (Jara-Olmedo A. , y otros, 2018) (Pardo, Aguilar, & Toulkeridis, 2017) (Orbea, y otros, Math Model of UAV Multi Rotor Prototype with Fixed Wing Aerodynamic Structure for a Flight Simulator, 2017) (Orbea, y otros, Vertical take off and landing with fixed rotor, 2017), ha sido factible la implementación de polígonos virtuales de tiro que permiten simular una práctica de tiro dentro de un ambiente controlado. Sistemas de entrenamiento como VirTra y MiloRange cuentan con armas láser, sistema de proyección que muestra objetivos virtuales o en su defecto videos, cámaras, y otros elementos.

VirTra, empresa americana, presenta simuladores de entrenamiento para personal de la policía y guardias de seguridad. El valor de productos que presenta varía entre 150.000 y 300.000 dólares dependiendo de la cantidad de pantallas y elementos adicionales. MiloRange es otra empresa del sector, sus precios alcanzan hasta 150.000 dólares, sin embargo, sus productos no simulan un entorno virtual, proyectando videos sobre diferentes escenarios de crimen. Dentro del territorio ecuatoriano no se ha desarrollado un sistema comercial de este tipo, por lo que se ha visto la necesidad de recurrir a compra de productos extranjeros.

La realidad virtual se ha convertido en una herramienta útil para simulaciones de entrenamiento y otras aplicaciones (Aguilar & Morales, 3D Environment Mapping Using the Kinect V2 and Path Planning Based on RRT Algorithms, 2016) (Amaguaña, Collaguazo, Tituaña, & Aguilar, 2018) (Aguilar, Morales, Ruiz, & Abad, RRT\* GL Based Path Planning for Virtual Aerial Navigation, 2017) (Cabras, Rosell, Pérez, Aguilar, & Rosell, 2011) (Aguilar, Abad, Ruiz, Aguilar, & Aguilar-

Castillo, 2017) (Aguilar, Morales, Ruiz, & Abad, RRT\* GL Based Optimal Path Planning for Real-Time Navigation of UAVs, 2017). Se han empleado diferentes métodos para que una persona pueda interactuar directamente con los ambientes virtuales. El progreso de las tecnologías actuales ha permitido crear una conexión entre el mundo real y el mundo virtual (Leung, Chan, Tang, & Komura, 2011) (Xu, 2008).

Los motores de desarrollo de videojuegos se han transformado también en herramientas de simulación que ofrecen soportes como alta calidad en renderizado de gráficas, variedad de recursos para la construcción de ambientes virtuales, múltiples lenguajes de programación, compatibilidad con diferentes plataformas y dispositivos, entre otros. Unreal Engine y Unity son líderes en este mercado, y han permitido el desarrollo de cientos de juegos en diferentes plataformas entre móviles y de escritorio. Unreal permite trabajar en 2D como en 3D, su lenguaje de programación es en C++ o Blueprint (lenguaje creado por la misma compañía), cuenta con versiones libres, sin embargo, cuando se intenta comercializar el producto es necesario adquirir una licencia (Unity, 2018). Unity es un motor que trabaja de forma similar a Unreal, aunque su programación se la realiza en C# o Javascript, y presenta varias opciones de licencia, permitiendo trabajar también de forma gratuita (Unreal Engine, 2018).

La visión por computador es un campo de la ciencia que mediante la extracción y procesamiento de imágenes del mundo real permite que un sistema pueda comprender de forma similar a como un humano usa el sentido de la vista (Aguilar W. G., et al., Pedestrian Detection for UAVs Using Cascade Classifiers and Saliency Maps, 2017) (Aguilar W. G., et al., 2017) (Aguilar W. G., et al., 2017) (Aguilar W. G., et al., Real-Time Detection and Simulation of Abnormal Crowd Behavior,

2017) (Aguilar W. G., et al., Cascade Classifiers and Saliency Maps Based People Detection, 2017). Dentro de esta disciplina se han desarrollado a lo largo de la historia varios métodos para el tratamiento de imágenes tanto en tiempo real como en post procesamiento. La detección de puntos de interés es uno de los problemas más comunes en visión por computador, se refiere a identificar líneas, puntos, regiones, esquinas, bordes, etc., dentro de una imagen (Li, Wang, Tian, & Ding, 2015) (Zitova & Flusser, 2003), se han perfeccionado varios algoritmos para la detección de características pero los que más destacan han sido: Speeded-Up Robust Features (SURF) (Bay, Tuytelaars, & Van Gool, 2006), Scale Invariant Feature Transform (SIFT) (Lowe, 2004), ORB (Rubble, Rabaud, Konolige, & Bradski, 2011), Features from Accelerated Segment Test (FAST) (Rosten & Drummond, Machine learning for high-speed corner detection, 2006), Binary Robust Independent Elementary Features (BRIEF) (Calonder, Lepetit, Strecha, & Fua, 2010). Adicionalmente se han realizado comparaciones entre varios de los algoritmos mencionados para medir el rendimiento basado la detección de puntos de interés aplicado sobre imágenes sometidas a diferentes transformaciones o deformaciones como es el ruido, rotaciones, escalamientos u otras. Como se menciona en (Karami, Prasad, & Shehata, 2017) ORB es un algoritmo con menor tiempo de ejecución, además SIFT presenta un mejor desempeño, sin embargo, cuando la rotación de las imágenes es notable SURF y ORB son superiores. En cuanto al ruido ORB y SIFT brindan mejores.

Un blob o una mancha es también uno de los principales problemas en la detección de características en una imagen, debido a sus propiedades es incapaz de ser reconocido por un detector de esquinas o bordes, sin embargo, debido a sus características como su forma, brillo, área o convexidad puede ser encontrado en una imagen (Zhang, Wu, & Beeman, 2016). Un blob se

define como una región la cual difiere en brillo o color con las regiones que se encuentran cerca (Soliman, 2016).

En la literatura se registran proyectos de investigación del mismo tipo, en (MEŠKO & TOTH, 2013) se recreó un sistema de cámara-proyector el cual menciona sobre una compensación en los parámetros de la cámara, tanto en exposición como en ganancia para ayudar a que el haz de laser no se pierda dentro de la proyección. (Ryt & Sobel, 2014) inicia con la detección de blobs en la imagen, además de eliminar el ruido mediante una serie de filtros y realiza variación en el brillo de la imagen para evitar perturbaciones. En (Lapointe & Godin, 2005) se implementó un sistema para controlar el cursor de una computadora mediante el uso de un dispositivo laser, el uso de métodos de morfología de imágenes ayudo a distinguir las manchas generadas por el láser. También se han implementado estos algoritmos en entornos virtuales tal como se propone en este trabajo, (Domínguez, 2016) y (Rohit, 2006) han desarrollado ambientes de tiro simulados, en cada uno se crea un algoritmo propio para obtener los mejores resultados, sin embargo en ninguno de los trabajos mencionados consta la detección del área en la cual se reconocerá la acción del láser, por tanto este proyecto presenta un aporte y contribución para el desarrollo de sistemas similares que se puedan implementar en un futuro.

Tomando en cuenta estos antecedentes y con el propósito de brindar una solución completamente funcional se propone en este trabajo de investigación un sistema el cual permita reconocer los disparos laser y recreándolo en un ambiente totalmente virtual, todo esto basado en visión por computador.

## 1.2 Justificación e importancia

Las Fuerzas Armadas, así como la Policía Nacional, han sido las encargadas de la formación de los ciudadanos que prestan servicio a su país. Estas instituciones brindan una instrucción tanto física como académica, asegurando la excelencia de sus miembros. Una de las áreas más importantes dentro de la instrucción, es la preparación para el uso de armamento como pistolas, fusiles, etc.

Con el transcurso de los años se han creado varios métodos de entrenamiento para el uso de armas; el ejercicio más sobresaliente es el que se realiza al interior de un “Polígono de tiro”, que consta de un terreno aislado el cual posee varios objetivos de diferentes tamaños y a diferentes distancias, pero, sobre todo, es un espacio que requiere una seguridad específica que garantice que el entrenamiento no va a causar daños a terceros, por tanto, son espacios físicos no tan fáciles de encontrar. El tirador obtiene una evaluación de acuerdo con el sector del blanco donde impactó su disparo.

Pese a los buenos resultados que se han obtenido del Polígono de Tiro, la inversión en municiones va en incremento, representando una gran inversión económica para las Fuerzas Armadas. Por este motivo se plantea el uso de las nuevas tecnologías para el desarrollo de una aplicación en realidad virtual, la cual proporcione a los usuarios una experiencia lo más cercana a la realidad.

El Centro de Investigación Científica y Tecnológica del Ejército (CICTE) propone el desarrollo de un Polígono de Tiro Virtual de bajo costo, como una formación previa al Polígono de Tiro Real,

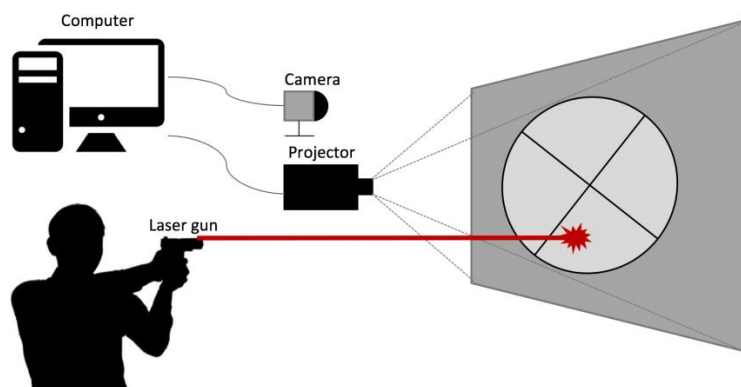
con el objetivo de disminuir gastos en municiones y se pueda brindar un mayor tiempo de entrenamiento a cada alumno, pero sobre todo de optimizar el uso de recursos a través de un entrenamiento más efectivo de sus miembros.

Este trabajo se enfoca en el desarrollo de un sistema de evaluación para un polígono virtual de tiro basado en visión por computador para entrenamiento militar. Este simulador es una parte importante en el engranaje de todo el proyecto de polígono virtual de tiro, que ha iniciado a desarrollar el CICTE; por lo que su realización es una base sólida para continuar con la ejecución del proyecto. Además, es importante mencionar que un simulador de este tipo no existe en el mercado nacional, de acuerdo con la investigación del estado del arte, únicamente se han encontrado simuladores de fabricación extranjera, cuyos precios de venta son altos, provocando por tanto un encarecimiento en la implementación de un proyecto similar, abriendo la oportunidad de asumir el reto de hacer un simulador de este tipo, con talento e intelecto ecuatoriano. En ese contexto y procurando alcanzar la filosofía actual de desarrollo nacional, que busca disminuir la dependencia de agentes exógenos e incrementar la matriz productiva, es menester apoyar este tipo de proyectos.

### **1.3 Alcance del proyecto**

El presente proyecto de investigación tiene como objetivo combinar, en un polígono virtual de tiro, el uso de un disparador láser, escenarios virtuales proyectados y algoritmos de visión por computador para evaluar el tiro de forma automática y en tiempo real, a fin de dar una experiencia

de entrenamiento comparable a la de un polígono de tiro real. La Figura 1 presenta el esquema general del prototipo.



**Figura 1.** Esquema general del polígono virtual

Se han establecido los siguientes elementos del sistema:

*Disparador láser:* se empleará un elemento el cual genere un destello laser notorio capas de ser detectado por el sistema y poder replicar los disparos de un arma.

*Cámara:* la cámara seleccionada para la detección debe obtener imágenes en un periodo inferior al tiempo de emisión de luz del disparador láser. Por esto se ha considerado utilizar una cámara con un promedio de recolección entre 20 y 30 frames por segundo.

*Computadora:* se ha seleccionado el motor de videojuegos Unity para la implementación del proyecto, tomando en cuenta las necesidades del mismo se requiere un computador con las siguientes características mínimas: Windows 7 SP1 o superiores, Procesador de 64 bits, RAM de 8 GB y Tarjeta gráfica con DX9 o DX11



*Proyector:* con conexión HDMI o VGA

Adicionalmente se han planteado las etapas respectivas en el desarrollo del proyecto:

*Estudio del estado del arte:* se realizará un estudio del arte en el cual se recopile toda la información de investigaciones previas referentes al tema a desarrollarse, tanto en la aplicación militar de polígonos de tiro y polígonos de tiro virtuales existentes en el mercado, visión por computadora, calibración de cámaras, transformaciones geométricas, reconocimiento de láser y recreación de ambientes mediante realidad virtual.

*Detección del área de disparo:* se implementará un algoritmo capaz de detectar la zona de disparo proyectada sobre una superficie plana. Se plantean dos métodos:

- Calibración manual, en la cual el usuario seleccionará el área de trabajo a través de puntos de interés.
- Calibración automática, basado en visión por computador, en la cual se pretende implementar algoritmos detectores, descriptores y desestimadores de correspondencia (ORB, SURF, SIFT, BRIEF), incluso combinar estos algoritmos, para poder reconocer automáticamente la proyección en todo el campo visual de la cámara. Posteriormente realizar una comparación del rendimiento para determinar el método más efectivo que cumple con las expectativas.

En ambos casos se trabajará adicionalmente con transformaciones geométricas y homografía que permitan variar la perspectiva de la imagen y simplificar su posterior procesamiento.

*Detección y virtualización del disparo láser:* Consiste en el desarrollo de un algoritmo para la detección del láser dentro del área de trabajo. Mediante el procesamiento de imágenes se trabajará con reconocimiento de colores, detección de bordes, centro de masa, y detección de blobs, necesario para discriminar el contorno del láser y su centro dentro de la imagen recibida por la cámara. A continuación, se realizará la digitalización del disparo buscando recrearlo dentro de un ambiente virtual en la plataforma Unity.

*Creación de escenarios virtuales:* Se crearán los escenarios virtuales en Unity, que incluyen varios tipos de objetivos, los mismos que serán modelados para asignar una calificación dependiendo del lugar alcanzado por el láser, dando un mayor realismo a la práctica de tiro. Se desarrollan los menús de instructor en el cual se podrá seleccionar el tipo de entrenamiento y el monitoreo de la práctica de tiro en tiempo real, así como el respectivo sistema de calificación automático.

*Evaluación del sistema:* En esta etapa se realizarán las pruebas de funcionamiento mediante el uso del disparador láser y el apoyo de personal militar que evaluara el desempeño del proyecto. Se realiza corrección de errores tanto en algoritmos y ambientes virtuales, análisis de la precisión del reconocimiento del disparo y tiempo de respuesta del sistema.

## **1.4 Objetivos**

### **1.4.1 Objetivo General**

Desarrollar un sistema de evaluación para un polígono virtual de tiro basado en visión por computador.

### **1.4.2 Objetivos Específicos**

- Analizar el estado del arte sobre la caracterización de imágenes mediante la detección de puntos de interés.
- Diseñar algoritmos que permitan la detección de la zona de disparo mediante una calibración de cámaras basado en transformaciones geométricas y puntos de interés.
- Definir los parámetros necesarios bajo los cuales el sistema presenta un alto rendimiento y el mínimo error en la reconstrucción del disparo laser.
- Aplicar fundamentos de visión por computador para desarrollar algoritmos que permitan detección de láser en tiempo real.
- Evaluar el desempeño de algoritmo de detección de láser, basándose en la precisión de la reconstrucción del disparo dentro del ambiente virtual

### **1.5 Descripción general del proyecto de investigación**

El proyecto de investigación consiste en la elaboración de dos algoritmos que basados en la detección de puntos de interés permiten detectar la proyección de un escenario virtual proyectado sobre una superficie y los disparos laser que se realizan sobre las misma, de esta manera crear un sistema de calificación dependiendo las siluetas alcanzadas por los disparos realizados.

El primer algoritmo es el encargado de reconocer el área de proyección, a partir de una imagen creada para cumplir con este propósito, se realizará una comparación con los algoritmos de

caracterización de imágenes más robustos: SURF, ORB y SIFT. De esta manera se podrá determinar cuál de estos posee un mayor rendimiento y cubre con las exigencias planteadas.

Para el reconocimiento del disparo laser se implementará otro algoritmo, el cual, basado en la manipulación de espacios de colores, morfología de imágenes y detección de blobs se pretende determinar el centro del haz laser para insertarlo en un entorno virtual en 3D.

El desarrollo del software se lo realizara en el motor de videojuegos Unity, aquí se recrearán siluetas y escenarios virtuales, además de la implementación de los algoritmos en lenguaje de programación C# empleando las librerías de visión artificial OpenCV. Adicionalmente se generarán los reportes correspondientes de prácticas de tiro realizados.

## CAPÍTULO II

### MARCO TEÓRICO

Este capítulo presenta los fundamentos teóricos correspondiente con polígonos de tiro, además una revisión general de visión por computador y los métodos para la caracterización de imágenes, temas fundamentales para la implementación del presente proyecto.

#### **2.1 Polígono de tiro**

Un polígono de tiro es un área adecuada para la práctica y entrenamiento de disparo con armas de fuego o arquería. Generalmente son operados por agentes militares o miembros de la policía, ellos son los responsables de guiar las prácticas de tiro y brindar las normas de seguridad respectivas sobre el uso de armas. La principal función de los polígonos de tiro es la capacitación de personal en el uso de armamento y afinar sus destrezas para alcanzar el objetivo. Los polígonos de tiro se los puede encontrar en interiores y exteriores dependiendo del tipo de arma de fuego que se puede usar en cada uno de estos ambientes y de los impactos que se generan.

Los polígonos de tiro en interiores pueden ser construcciones aisladas o ser parte de edificaciones más grandes. Los componentes básicos que contienen consiste en carriles de tiro, siluetas y trampa de balas, la responsable de detener las balas, además se presentan elementos como protecciones de seguridad para el usuario, aislamiento acústico e iluminación apropiada (General Services Administration, 2011) (Kardous & Murphy, 2010) . En Figura 2 se puede apreciar una práctica de tiro en este ambiente.



**Figura 2.** Polígono de tiro en interiores

Fuente: (Double Action Indoor Shooting Center, 2018)

Por el otro lado, los polígonos de tiro en exteriores son amplios campos que alcanzan distancias de hasta 1000 metros o más, para el uso de armas de largo alcance. Las siluetas y muros de retención de balas son elementos básicos en estos establecimientos, además se debe considerar los factores climáticos como la iluminación y el viento debido a que pueden influir en la dirección de las municiones. Las banderas son elementos igualmente importantes ya que muestran la dirección del viento, que debe ser considerada por la persona que realiza el disparo (CDC, 2013). Los rifles, escopetas, fusiles y pistolas son usados normalmente en estos escenarios. En Figura 3 se muestra un polígono de tiro exterior.



**Figura 3.** Polígono de tiro en exteriores

Fuente: (Las Vegas Outdoor Shooting Range, 2018)

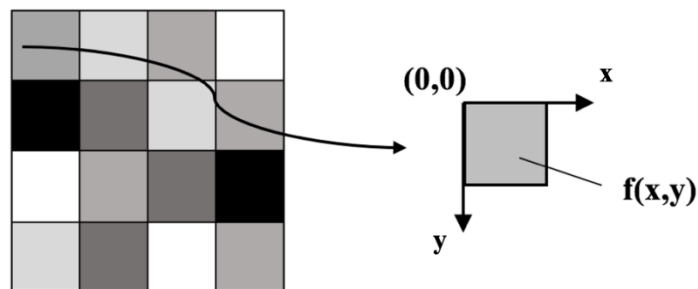
La selección del tipo de polígono de tiro depende de las necesidades del usuario, sin embargo, es de carácter obligatorio mantener las debidas normas de seguridad.

## 2.2 Obtención de imágenes

El desarrollo del proyecto se basa principalmente en adquisición de imágenes digitales y su respectivo procesamiento para encontrar la información necesaria. La adquisición de imágenes se lo realiza a mediante cámaras digitales, cámaras web, teléfonos inteligentes u otros dispositivos digitales. Una imagen es una matriz bidimensional de valores que representan la intensidad de la luz (Ma, Soatto, Košecká, & Sastry, 2003), por tanto, dicha imagen se encuentra en función de la intensidad de la luz:

$$v = f(x, y) \quad (1)$$

donde  $x$  y  $y$  corresponden a la coordenada del píxel dentro de la imagen y sus valores están representados por números reales que definen el nivel de gris asociado al punto  $p(x, y)$ , como se puede observar en Figura 3.



**Figura 4.** Representación de una imagen mediante una matriz.

La matriz formada por la imagen mantiene el valor de sus coordenadas en números enteros, de igual forma el valor de la función (1) también se representa como entero.

Para una imagen digital, el dominio está constituido por:

$$R = \{(x, y), 0 \leq x \leq x_m - 1, 0 \leq y \leq y_m - 1\} \quad (2)$$

Donde  $x_m$  y  $y_m$  corresponden al número de columnas y filas respectivamente.

Y el rango de sus valores de la función es:

$$0 \leq f(x, y) \leq g - 1 \quad (3)$$

Donde  $g$  es el número de niveles de gris.

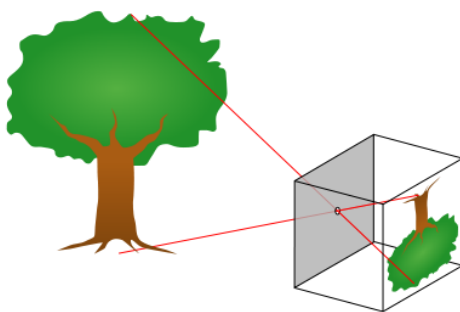
Los valores en  $x$  que representan las filas de la matriz inician en 0 e incrementan su valor de izquierda a derecha, en el caso de las columnas representadas por  $y$  e incrementan desde arriba hacia abajo.

### 2.2.1 Modelo de Pin-Hole

El modelo Pin-hole permite determinar la relación geométrica que existe entre un punto, en tres dimensiones, perteneciente al mundo real, y su respectiva proyección sobre una imagen en dos dimensiones (Aguilar, Casaliglla, & Pólit, Obstacle Avoidance for Low-Cost UAVs, 2017) (Aguilar, y otros, Real-Time 3D Modeling with a RGB-D Camera and On-Board Processing, 2017) (Aguilar, Casaliglla, & Pólit, Obstacle Avoidance Based-Visual Navigation for Micro Aerial Vehicles, 2017) (Aguilar, y otros, On-Board Visual SLAM on a UGV Using a RGB-D Camera,

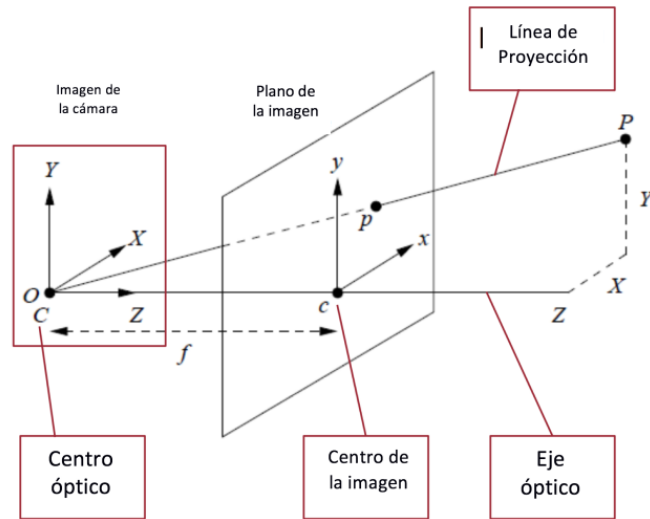


2017) (Aguilar, Casaliglla, Pólit, Abad, & Ruiz, Obstacle Avoidance for Flight Safety on Unmanned Aerial Vehicles, 2017) (Aguilar, y otros, Visual SLAM with a RGB-D Camera on a Quadrotor UAV Using on-Board Processing, 2017). En Figura 4. Se puede observar el funcionamiento de este modelo para la obtención de imágenes.



**Figura 5.** Modelo Pin-Hole

El proceso de mapeo geométrico de 3D a 2D es también conocido como proyección en perspectiva. El centro de la proyección en perspectiva, donde se intersecan todos los rayos, lleva el nombre de centro óptico o centro de la cámara; atravesando del centro óptico perpendicularmente hacia el plano de la imagen se encuentra una línea, conocida como eje óptico (Sturm, 2016). En Figura 5 se puede apreciar de forma visual estos conceptos.



**Figura 6.** Funcionamiento de modelo Pin-hole

Entonces un punto P en el espacio se denota por:

$$P = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (4)$$

y proyecta sobre una imagen 2D un punto p:

$$p = \begin{pmatrix} x \\ y \end{pmatrix} \quad (5)$$

Por simetría se obtiene el modelo de Pin-hole:

$$\frac{X}{Z} = \frac{x}{f}, \frac{Y}{Z} = \frac{y}{f} \quad \rightarrow \quad x = f \frac{X}{Z}, y = f \frac{Y}{Z} \quad (6)$$

Donde:

$x, y$ : coordenadas (píxeles)

$f$ : distancia focal (m)

Se puede mencionar entonces que para un punto P real de coordenadas (X, Y, Z) se asigna un punto p en una imagen (x, y). Usando la representación homogénea de los puntos, se obtiene una ecuación de proyección lineal como:

$$\begin{bmatrix} fx \\ fy \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & f & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (7)$$

Dado M una matriz que representa una imagen se obtiene la matriz de rotación y traslación de la siguiente manera:

$$M' = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} M = \begin{bmatrix} \mathbf{R} & t \\ 0 & 1 \end{bmatrix} M \quad (8)$$

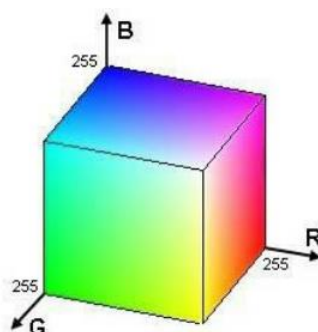
Con  $\mathbf{R}$  como la matriz de rotación y  $t = [t_x t_y t_z]^T$  como un vector de traslación.

### 2.2.2 Espacios de colores

Una maquina no puede percibir los colores como un ser humano lo hace, por esto se han establecido diversos espacios de colores, que se basa en una aproximación matemática para representar la intensidad de cada color como lo haría una persona normal. Cada espacio de colores brinda ventajas y desventajas, además cada uno se ajusta a diferentes aplicaciones. (Li & Plataniotis, 2015), a continuación, se analiza los espacios de colores aplicados para el desarrollo de este proyecto.

### 2.2.2.1 RGB

Las siglas RGB provienen del inglés Red (rojo), Green(verde) y Blue(azul), los cuales son considerados los colores primarios. Este modelo implica la unión de la luz roja, verde y azul en diferentes proporciones para generar una amplia gama de colores, es decir, una imagen digital se encuentra formada por tres planos de imágenes independientes, cada una de las cuales posee los valores de rojo, verde y azul (Blotta & Ballarin, 2012). En Figura 6 se muestra la representación del modelo RGB y la creación de cada color, el valor que toma la intensidad de rojo, verde y azul varía entre 0 y 255. Cuando la combinación de colores es  $[0, 0, 0]$  se genera el color negro, mientras que cuando es  $[255, 255, 255]$  el color obtenido será blanco.



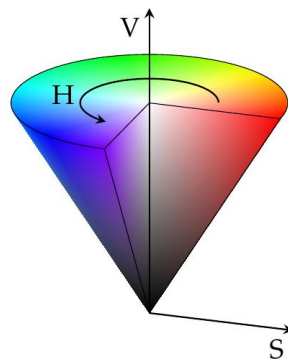
**Figura 7.** Representación del modelo RGB

Fuente: (ResearchGate, 2018)

El modelo RGB es con frecuencia el más utilizado para la generación de imágenes digitales, en cuanto al procesamiento de la misma, este modelo permite obtener formas y detalles significativos sin embargo no es el adecuado para procesarla (Kumar & Verma, 2010).

### 2.2.2.2 HSV

El modelo HSV caracteriza los colores en función de sus componentes H (tonalidad), S (saturación), y V (valor), y proporciona la representación de la percepción de acuerdo con la característica visual humana.



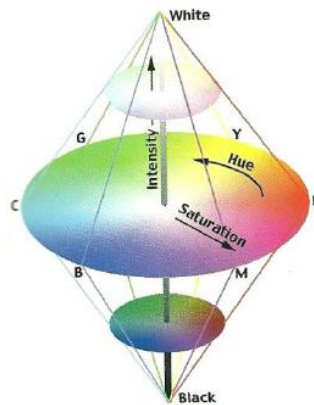
**Figura 8.** Representación modelo HSV

Fuente: (ResearchGate, 2018)

En Figura 7 se muestra la representación gráfica del modelo HSV, que muestra un cono en el cual la base superior muestra el valor de tonalidad, el mismo que se refiere al color variando su valor desde 0 hasta  $2\pi$ , iniciando desde 0. El valor de saturación medido entre 0 y 1, parte de desde el eje hacia el extremo de la circunferencia y determinar el nivel de blanco dentro del color, mientras más cerca se encuentra del eje más saturado y claro será el color. Finalmente, el valor se mide en el eje central del cono mostrando la cantidad de negro que posee el color, mientras más bajo sea el valor más oscuro será el color y su rango de valores va de 0 a 1. Para el procesamiento de una imagen digital, este modelo es muy usado para detectar colores y segmentación. (Kaur & Vijay, 2013)

### 2.2.2.3 HSI

El modelo HSI posee la capacidad de separar sus componentes entre H (tonalidad), S (saturación), e I (intensidad). La manera más tradicional de representar este modelo es un cono doble como se muestra en Figura 8, la tonalidad se encuentra en el centro y describe el color dependiendo su longitud de onda que toma valores entre 0 y  $2\pi$ , donde 0 corresponde a rojo. La saturación del color representada por la línea que va desde el centro hasta el exterior de la circunferencia tomar valores entre 0 y 1, y muestra la mezcla del color con luz blanca. Adicionalmente el eje entre los dos conos muestra la intensidad que varía entre 0 (negro) y 1 (blanco), e indica la cantidad de luz que se encuentra presente en un color.



**Figura 9.** Representación modelo HSI de colores

Fuente: (ResearchGate, 2018)

El modelo HSI es el indicado para procesar imágenes que presentan cambios de iluminación, debido a que los colores del entorno se distinguen entre si a través del componente de tonalidad (Blotta & Ballarin, 2012).

### 2.3 Homografía

En el campo de la visión por computador, cuando existen dos imágenes en las cuales se puede apreciar la misma superficie plana en el espacio, se dice que ambas imágenes se encuentran relacionadas por una matriz de homografía (Etienne & Laganier, 2001). La homografía relaciona los píxeles de las dos imágenes y existe bajo las siguientes condiciones:

- Ambas imágenes deben ser tomadas desde la misma cámara.
- Ambas imágenes están observando el mismo plano desde ángulos diferentes, es decir, la cámara gira sobre su centro de proyección.

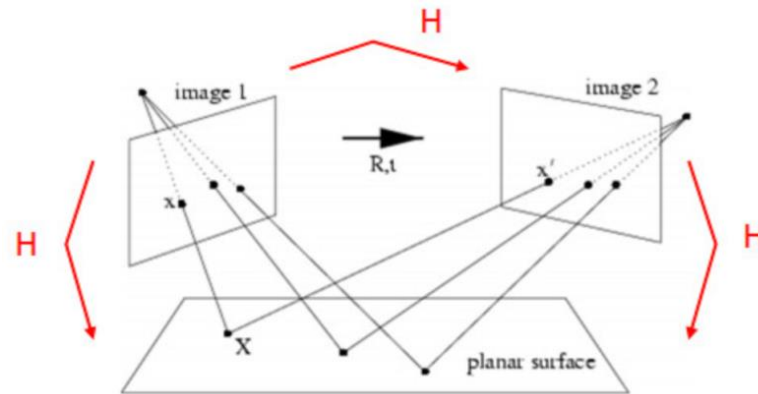
Considerando un punto  $p = (x, y, z)$  proyectado sobre la primera imagen, y el punto  $x' = (x', y', 1)$  en la segunda imagen, la matriz de homografía  $H$  de 3 por 3 se define como:

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad (9)$$

Entonces se establece la relación:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (10)$$

En Figura 10 se muestra de forma más visual como es el funcionamiento de la matriz de homografía sobre la captura de la misma superficie plana a través de dos imágenes diferentes.



**Figura 10.** Representación gráfica del funcionamiento de la matriz de Homografía.

Fuente: (OpenCV Docs, 2018)

La matriz de homografía presenta múltiples aplicaciones como es la corrección de imágenes en perspectiva, el registro de imagen, construcción de imágenes panorámicas, calibración de cámaras, estimación de pose de la cámara (fundamento para desarrollo de realidad aumentada), entre otros (Zhai, Yu, Wang, & Guo, 2017).

## 2.4 Caracterización de una imagen

Uno de los enfoques que se ha convertido en una herramienta muy útil en el campo de la visión por computador es la detección y descripción de características. Cuando es necesario comparar dos imágenes para diferentes usos, la comparación en sí trata de analizar características semejantes que poseen las dos imágenes para lo cual es necesario determinar un conjunto de puntos destacados en cada imagen para posteriormente realizar una correspondencia entre dichos puntos entre las dos imágenes (Awad & Hassaballah).



El procedimiento se establece primero con la detección de regiones de interés también llamados puntos clave (keypoints) obtenidos después de aplicar un cierto procesamiento a la imagen, posteriormente y como su nombre lo indica los descriptores son los encargados de asignar una descripción a la región de interés detectada previamente. En el proceso de extracción de características de una imagen se pueden tomar en cuenta que existen dos tipos: globales y locales. Las globales tratan de describir a una imagen de forma general (color, textura, etc.), es decir tratan de describir a la imagen como un todo. Por el contrario, cuando se extraen características locales, se trata de determinar diferentes tipos de características dentro de la misma imagen, para esto la detección y descripción de características es un principio fundamental (Awad & Hassaballah). En Figura 11 se puede observar los métodos de caracterización de una imagen.



**Figura 11.** Caracterización de una imagen de forma Global y Local

Fuente: (Awad & Hassaballah)

Las características locales pueden ser generalmente de cuatro tipos: bordes, esquinas (puntos de interés), blobs o crestas, basándose en la formación que presentan sus píxeles. En el caso de los bordes, estos están generalmente representados como el conjunto de puntos que poseen una fuerte magnitud de gradiente dentro de una imagen (Harris & Stephens, 1998). Los puntos de interés no

siempre están representados por una esquina, intersección de dos bordes, sin embargo, estos pueden definirse como un elemento aislado dentro de otros del mismo tipo lo que hace que prevalezca en su entorno (Harris & Stephens, 1998). Los blobs mejor conocidos como manchas son puntos de interés de una mayor extensión en la cual cada uno de sus elementos cuentan con características similares, cuando en estas regiones se aplica un detector de esquinas o bordes no surge ningún efecto debido a su composición (Lindeberg, Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention, 1993). Cuando se habla de crestas cabe mencionar que se trata de una curva unidimensional que representa un eje de simetría y cuenta con un ancho, estos objetos son las difíciles de extraer de una región a comparación de los otros.

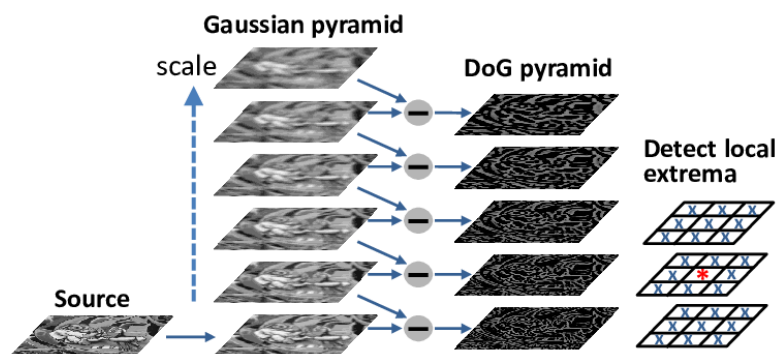
#### **2.4.1 Algoritmos de detección y descripción de puntos de interés**

A lo largo de los años se han desarrollado varios algoritmos para la detección y descripción de puntos de interés en imágenes, cada uno mantiene su propio enfoque. A continuación, se describirán los principales algoritmos que serán implementados posteriormente.

##### **2.4.1.1 SIFT**

Este algoritmo fue presentado en 2004 por David Lowe en su trabajo “Distinctive Image Features from Scale-Invariant Keypoints”, en el cual presenta un enfoque para detección de características basándose en funciones Gaussianas. La imagen inicial realiza varios procesos de convolución con varias funciones Gaussianas las cuales se encuentran a diferentes escalas obteniendo un nuevo conjunto de imágenes de las cuales se toma la diferencia sucesiva entre estas,

generando el bloque de imágenes conocido como Diferencia de Gaussianos (DoG), finalmente los puntos de interés son los valores máximos y mínimos que se producen en las múltiples escalas (Lindeberg, 1998), (Lindeberg, 1993), este proceso se puede observar en la Figura 12.



**Figura 12.** Proceso de detección de puntos de interés mediante el algoritmo SIFT

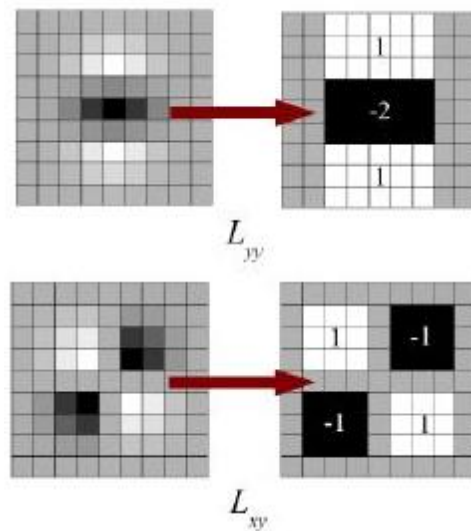
Fuente: (Research Gate, 2018)

Para la descripción de los puntos de interés encontrados se lo realiza mediante la toma de un conjunto de pixeles cercanos conocido como vecindario, que posee una dimensión de  $16 \times 16$ , dentro del vecindario se subdivide en bloques de  $4 \times 4$  y en cada uno se genera un histograma de orientaciones. Para obtener el vector descriptor se une los valores de los histogramas de todos los bloques del punto de interés, constituyendo el descriptor del mencionado punto.

#### 2.4.1.2 SURF

En 2006, Herbert Bay presento el algoritmo “Speeded Up Robust Features” para la detección y descripción de puntos de interés en una imagen. Está basado en el mismo principio de SIFT, con las mismas etapas, pero se diferencia en los detalles de cada paso. A diferencia de SIFT, SURF emplea filtros de cajas simples 2D para aproximar la Laplaciano de una Gaussiana, esto reduce el

tiempo de ejecución ya que se puede calcular fácilmente mediante imágenes integrales. Sin embargo, la diferencia notoria entre SIFT y SURF es que el primero genera una pirámide de imágenes Gaussianas mientras que SURF varía el tamaño del filtro que se aplica sobre la imagen. Finalmente se realiza el proceso de selección de mínimos y máximos al igual que SIFT para determinar los puntos de interés. En la Figura 13 se observa los filtros de cajas simples que usa SURF en lugar de la Gaussiana.



**Figura 13.** Proceso de detección de puntos de interés mediante el algoritmo SURF

Fuente: (Bay, Tuytelaars, & Van Gool, 2006)

Para el proceso de descripción de los puntos de interés, se genera una región cuadrada alrededor de este, y tomando todos los puntos de la región genera 16 subregiones en las cuales se calcula la gradiente  $(dx, dy)$ . Cada subregión crea un vector de dimensión  $16 \times 4$  con los gradientes:

$$v = \left( \sum dx, \sum dy, \sum |dx|, \sum |dy| \right) \quad (11)$$

### 2.4.1.3 ORB

“Oriented FAST and Rotated BRIEF” es un algoritmo para detección y descripción de puntos de interés publicado por Ethan Rublee en 2011. Está basado en la detección de FAST (Rosten & Drummond, 2006) y la descripción de BRIEF (Calonder, Lepetit, Strecha, & Fua, 2010) creando un nuevo algoritmo capaz de detectar esquinas dentro de una imagen en un tiempo mínimo, lo cual hace que ORB sea una robusta alternativa a SIFT y SURF.

Para el proceso de detección se toma un conjunto de 16 píxeles alrededor del punto  $p$  actual, cada píxel está numerado del 1 al 16 en sentido horario. Se asigna a  $t$  como un valor de umbral y se establece  $I_p$  como la intensidad de brillo que posee ese punto a ser evaluado y  $N$  un número determinado de píxeles, a partir del cual se considera  $p$  como una esquina si se cumple cualquiera de las dos condiciones:

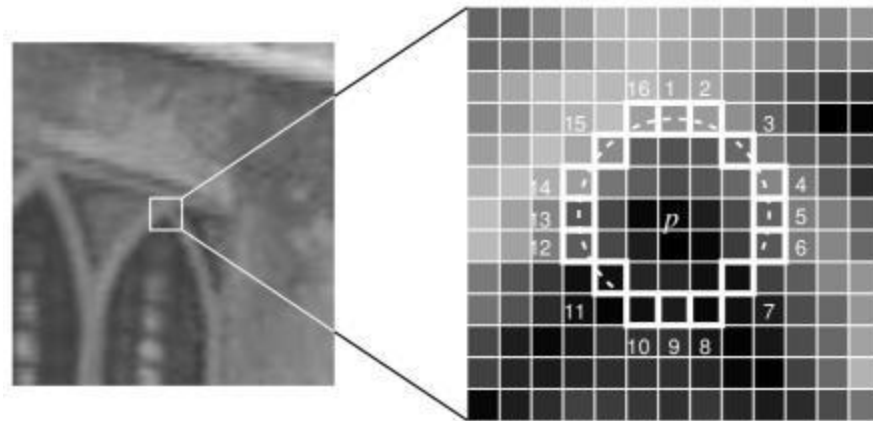
- El valor de la intensidad de al menos  $N$  vecinos es mayor que  $I_p + t$
- El valor de la intensidad de al menos  $N$  vecinos es menor que  $I_p - t$

En la Figura 14 se muestra el funcionamiento del algoritmo FAST.

A continuación, se presenta se realiza un suavizado de la imagen para dar lugar a BRIEF, el cual encuentra una cadena binaria directamente sin encontrar descriptores. Son seleccionados un conjunto  $n_d(x, y)$  pares de los puntos de interés detectados y se realiza la siguiente comparación:

$$\tau(p; x, y) = \begin{cases} 1 & p(x) < p(y) \\ 0 & p(x) \geq p(y) \end{cases} \quad (12)$$

Donde la función  $p()$  devuelve el nivel de intensidad en  $x$  o  $y$ , los descriptores serán obtenidos en un vector binario de dimensión 256.



**Figura 14.** Proceso de detección de puntos de interés mediante el algoritmo FAST

Fuente: (Rosten & Drummond, 2006)

## CAPÍTULO III

### GENERALIDADES DEL SISTEMA

En este capítulo, se describe de manera puntual los elementos tanto de hardware como de software para la implementación del proyecto, así como la distribución de estos dentro de ambiente controlado, se presenta también la integración de OpenCV con Unity.

#### 3.1 Especificaciones de hardware

Para la implementación de este prototipo, tres son los componentes fundamentales:

- Proyector de video
- Cámara web
- Estación de trabajo

##### 3.1.1 Proyector de video

El principal objetivo en el funcionamiento del prototipo mostrar escenarios virtuales los cuales contengan siluetas o targets claramente visibles para los usuarios, de tal manera que puedan ser fácilmente identificados. Surge la necesidad de utilizar un proyector de video sobre una superficie plana, donde dichos ambientes sean reflejados de manera adecuada y con resolución de alta calidad. Por lo mencionado se ha utilizado un proyector LCD EB-X04 de la empresa Epson que cuenta con Full HD, además de varias resoluciones incluida 1024 x 768 pixeles, en la cual se trabajara. Brillo

de colores y blancos de 2800 lm (lúmenes) y una variedad de colores que sobrepasa 1 billón de colores. En Figura 15 se muestra el proyector seleccionado.



**Figura 15.** Proyector Epson LCD EB-X04

Fuente: (Epson Gallery, 2018)

### 3.1.2 Cámara web

La obtención de imágenes se lo realizara mediante una cámara web, la cual debe ser capaz de entregar una imagen digital con la mejor resolución y el tiempo de muestreo debe ser el adecuado. Se ha seleccionado la cámara Logitech c920 que brinda una óptima calidad de video con resolución de 1080 pixeles a 30 FPS (frames por segundo) o 720 pixeles a 60 FPS. Cuenta con un campo visual de 78° y es compatible con el ordenador mediante tecnología USB 3.0. La Figura 16 muestra la cámara web.



**Figura 16.** Cámara web Logitech c920 HD

Fuente: (Logitech Gallery, 2018)



Debido a la cantidad de FPS que brinda la cámara, es idónea para el rastro del destello laser que se expone por apenas unas décimas de segundo.

### **3.1.3 Disparador Laser**

El dispositivo seleccionado en este proyecto para la evaluación del disparo que simula un arma es un puntero laser que posee una longitud de onda de 532 nm, es decir es un haz visible para el ojo humano con un destello de color verde. Genera una potencia de 200 mW, la cual no genera daños en objetos. Con estas características este laser es adecuado para comprobar el funcionamiento de los algoritmos que se desarrollaran.

### **3.1.4 Estación de trabajo**

La estación de trabajo es una computadora responsable del monitoreo de la detección del disparo, presentación de las calificaciones correspondientes y el desarrollo de los entornos virtuales. El proyector de video y la cámara web presentadas anteriormente se encontrarán conectadas a la computadora, la cual gestionara los tiempos adecuados para la proyección y la recolección de imágenes. Las especificaciones bajo las cuales trabaja el ordenador son:

**Tabla 1***Características de la estación de trabajo*

<b>Características</b>	
Fabricante	Dell Inc.
Procesador	Intel® Core™ i7-2600 CPU @ 3.40GHz
RAM	8.00 GB
Sistema Operativo	Windows 10 pro
Sistema	64 bits

Establecidas las especificaciones del computador se puede comprobar que satisfacen las necesidades de Unity 3D para trabajar en esta plataforma, de tal manera que el proceso de renderizado pueda realizarse sin disminuir el rendimiento del sistema.

### **3.2 Especificaciones de software**

Este proyecto se encuentra enfocado en la interacción humana con entornos virtuales mediante el reconocimiento de un haz laser, por esto, el software que se ha determinado para cumplir con estas funcionalidades es el motor de desarrollo de videojuegos Unity, este brinda las facilidades para poder modelar objetos en 3D, además de recrear escenarios con los detalles necesarios para crear una sensación de inmersión en el usuario. Se han usado la versión estable 2017.2.0f3 que brinda completa compatibilidad con OpenCV y brinda los recursos necesarios para la creación de escenarios virtuales.



*Figura 17.* Logotipo del software Unity

Fuente: (Unity, 2018)

Unity brinda compatibilidad con librerías de diferentes proveedores, para la detección de los disparos laser se requiere del uso de OpenCV, una librería especializada en visión artificial y machine learning, la versión empleada es la 3.1.0 la cual es completamente funcional dentro del software Unity, se ha obtenido del repositorio de OpenCV.



*Figura 18.* Logotipo de OpenCV

Fuente: (OpenCV, 2018)

Finalmente se genera un reporte con las calificaciones de los participantes, esta será emitida por Unity y presentada en una hoja de cálculo en Excel, aplicación de Microsoft Office. Para la exportación de datos ha sido necesario el uso de un paquete especial para la comunicación entre Unity y Microsoft Office cuyo nombre es Uni-Excel el cual permite crear, editar y leer archivos del tipo .xls.

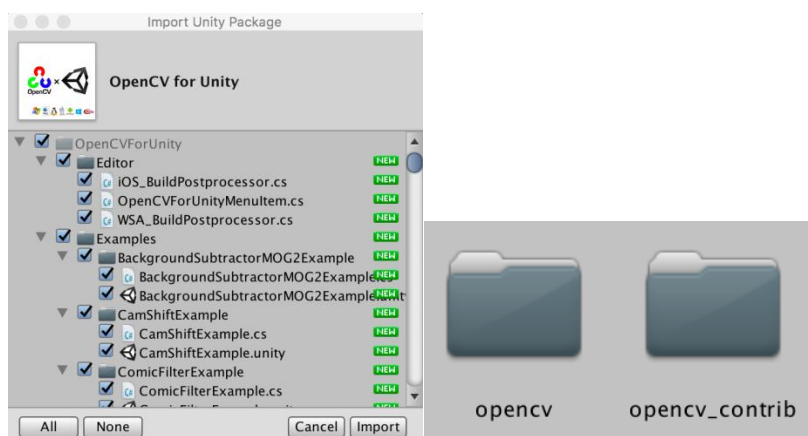
### 3.2.1 Instalación de librerías OpenCV en Unity

Las librerías de OpenCV especializadas para su uso en Unity se encuentran en el repositorio de aplicaciones para esta plataforma: Unity Assets Store. Los archivos que contiene son .dll compilados en el lenguaje de programación C++ compatibles con C#, el lenguaje que se usa para desarrollo en Unity. Estas librerías son compatibles con Windows, Mac, Linux y brindan soporte para dispositivos móviles como iOS y Android, además soporte para WebGL.



**Figura 19.** Logotipo de librería OpenCV para Unity en Assets Store

Una vez descargadas las librerías (pago adicional) se debe importarlas en un nuevo proyecto generado. Los archivos que se añaden incluyen dos carpetas principales “opencv” y “opencv\_contrib” en los cuales se encuentran los archivos compatibles para su uso en C# y los cuales son utilizados durante el desarrollo del proyecto. Adicionalmente incluye un set de ejemplos del uso de sus funciones, para de esta manera familiarizarnos de una forma adecuada con los comandos. Debido a la gran funcionalidad que presenta OpenCV, esta librería también brinda algoritmos para desarrollo en plataformas móviles, como es Android y iOS. Sin embargo, a pesar de los algoritmos de ejemplo que se incluyen, la documentación es limitada debido a que los implementados que se han realizado entre ambas plataformas son reducidos.



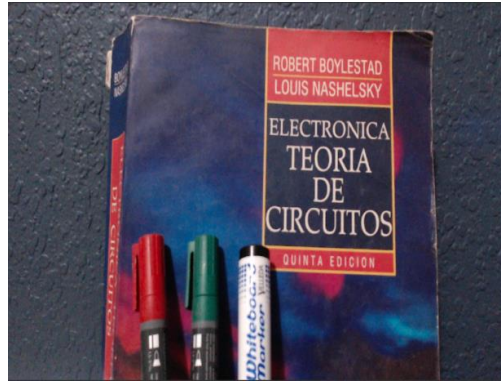
**Figura 20.** Importación de OpenCV y archivos generados

Cuando la importación ha finalizado está listo para ser usado en los proyectos necesarios, adicionalmente se puede mencionar que esta es la única manera en que se pueden acceder a las librerías desde Unity.

### 3.2.2 Extracción de imágenes de cámara web

El desarrollo de este proyecto está fundamentado en el tratamiento de imágenes, por tanto, la primera etapa constituye el acceso a la cámara web desde Unity. El primer paso es la creación de un objeto (VideoCapture) relacionándolo a la cámara que se usará, se debe mencionar específicamente el id o nombre de la cámara. Generalmente cuando un nuevo dispositivo se conecta en el ordenador se le asigna un id que va desde 0 hasta el número de cámaras detectadas, entonces es necesario conocer el identificador específico del dispositivo que se va a usar. El siguiente paso es la asignación de parámetros en la cámara, estos pueden ser los FPS (fotogramas por segundo), brillo, contraste, exposición, etc., estos valores pueden ser los que la cámara viene configurada previamente, sin embargo, es necesario conocer acerca de cada parámetro y su funcionamiento.

Finalmente, ya se puede acceder al fotograma captado y mostrarlo en pantalla. A continuación, en la Figura 21 se observa una imagen extraída desde un algoritmo en Unity.



**Figura 21.** Imagen extraída desde Unity mediante cámara web

A partir de la extracción de imágenes se puede continuar con el procesamiento de estas. Los algoritmos que se presentan para la detección del área de disparo y posición del láser necesitan un tratamiento más complejo sobre las imágenes, por tanto, es necesario aplicar inicialmente procesamientos básicos para evidenciar el trabajo correcto de las librerías.

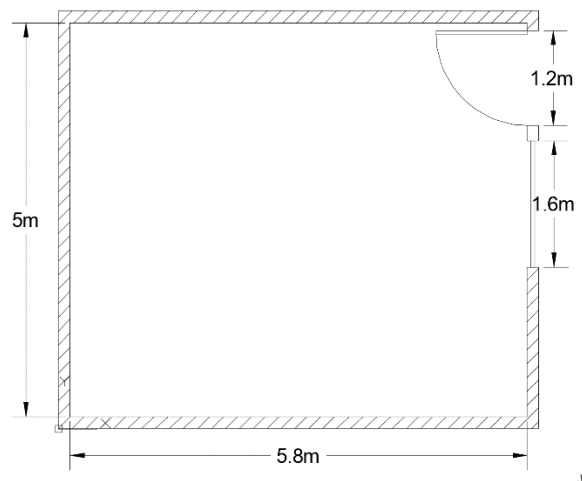
En la Figura 22 se puede observar el funcionamiento de diferentes algoritmos, la conversión entre espacios de colores que inicialmente se obtiene la imagen en RGB y se varía sus valores para obtener la misma imagen en escala de grises y en HSV (parte superior derecha e inferior derecha respectivamente), además se puede ver cómo funciona el proceso de binarización (parte inferior izquierda). Mediante la implementación de estos básicos algoritmos podemos comprobar el adecuado funcionamiento de la librería de OpenCV en el motor Unity.



*Figura 22.* Procesamiento de imagen en Unity

### 3.3 Especificaciones físicas del sistema

Una vez se conocen los equipos necesarios para el desarrollo del proyecto se debe realizar la instalación de estos, de tal manera que genere una sensación de inmersión en el usuario y que el sistema pueda trabajar de manera correcta. Inicialmente se debe considerar un entorno el cual brinde el espacio físico suficiente para realizar una práctica de tiro con pistola laser, adicionalmente es importante mencionar que este tipo de proyectos se los implementa en entornos totalmente controlados en cuanto a la iluminación y ventilación del lugar. Este proyecto se encuentra en el laboratorio de simulación en los hangares del Centro de Investigación Científica y Tecnológica del Ejercito (CICTE) ubicado en la Universidad de las Fuerzas Armadas ESPE. En este entorno se cuenta con el espacio necesario para la instalación de los equipos, como se puede ver en la Figura 23 las dimensiones son de 5.8 metros de ancho y 5 metros de largo con una altura de 2.35 m y cuenta con una puerta de 1.2 x 2.1 metros y una ventana de 1.6 x 1.25 metros.



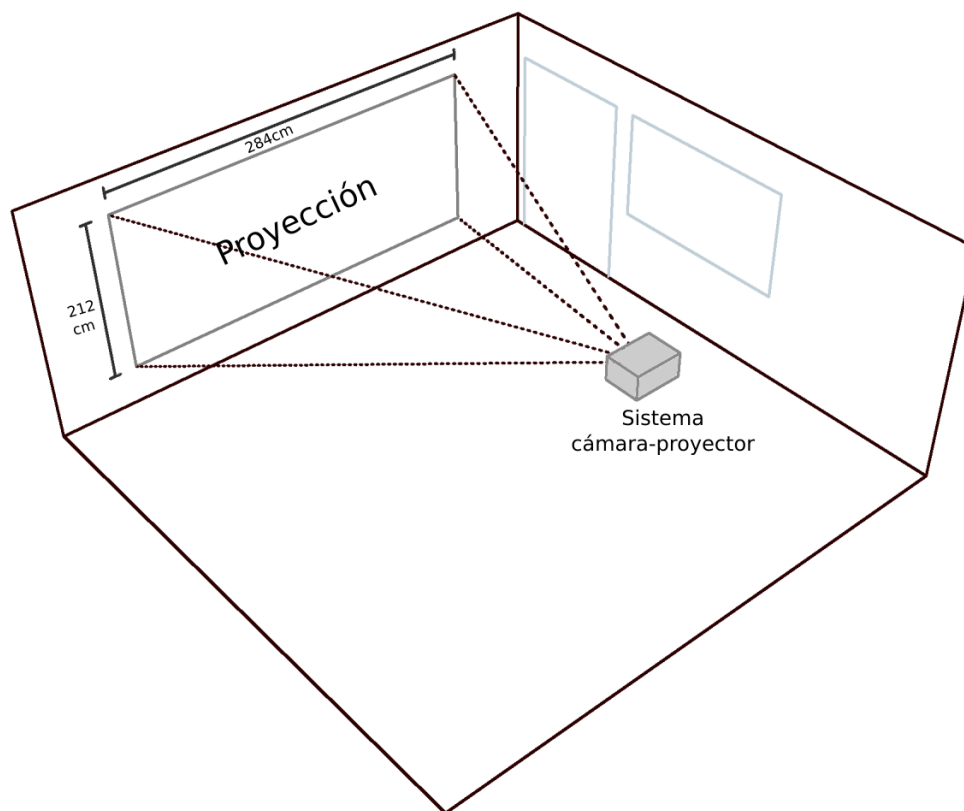
**Figura 23.** Plano del laboratorio de simulación

Para un mejor funcionamiento es ideal colocar la cámara web lo más próxima al lente del proyector de tal manera que la imagen obtenida se ajuste con la imagen proyectada. Conociendo esto, entonces se puede considerar como un sistema cámara-proyector (Rohit, 2006) (Lapointe & Godin, 2005) de tal manera que se lo toma en cuenta como un único dispositivo de hardware. Para que la imagen proyectada sea lo suficientemente visible y la cámara no tenga dificultad en recopilar datos el sistema cámara-proyector ha sido colocado a una distancia de cuatro metros desde la superficie de proyección y suspendido a una altura de 2.1 metros desde el piso, de tal manera que no sea una obstrucción para el usuario. A partir de la posición de nuestro sistema, la proyección que se genera es de 2.12 metros de alto por 2.84 metros de ancho sobre una superficie uniforme.

En la Figura 24 se puede apreciar un modelo en 3D de la instalación con la que se cuenta y como se han colocado los equipos en su lugar asignado. Ha sido necesario realizar adecuaciones adicionales al entorno, enfocados principalmente en la reducción de iluminación para que las prácticas de tiro se lo realicen de manera adecuada y brinden los resultados esperados. Debido a la



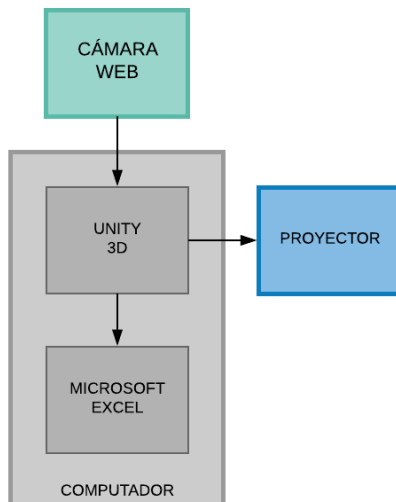
existencia de una puerta de cristal y una ventana la habitación posee iluminación de origen externo, esto será considerado en los siguientes capítulos.



*Figura 24.* Diagrama de instalación del sistema

### 3.4 Diagrama de bloques del sistema

Para continuar con la implementación del proyecto es adecuado identificar previamente los componentes que intervienen y cuál es la relación que poseen entre sí. Como parte central, o cerebro de todo el proyecto contamos con un computador que posee las características mencionadas anteriormente, este se encuentra conectado a sus respectivos periféricos que son una cámara web y un proyector. En la Figura 25 se puede observar el diagrama de bloques.



**Figura 25.** Diagrama de bloques del sistema

Internamente en el computador se ejecuta todos los algoritmos desarrollados en el entorno virtual Unity 3D, el mismo que se encarga de proyectar los escenarios virtuales creados y recoger la imagen de la cámara sobre dicha proyección. Internamente en Unity se procesa la imagen para reconstruir el disparo laser en la dirección adecuada y poder asignar una calificación en función de la silueta alcanzada, todas las calificaciones se ingresarán en un registro durante toda la práctica de tiro, y finalmente será exportado hacia un archivo de tipo .xlsx que puede ser visible en Microsoft Excel.

## CAPÍTULO IV

### CALIBRACIÓN DE LA CÁMARA

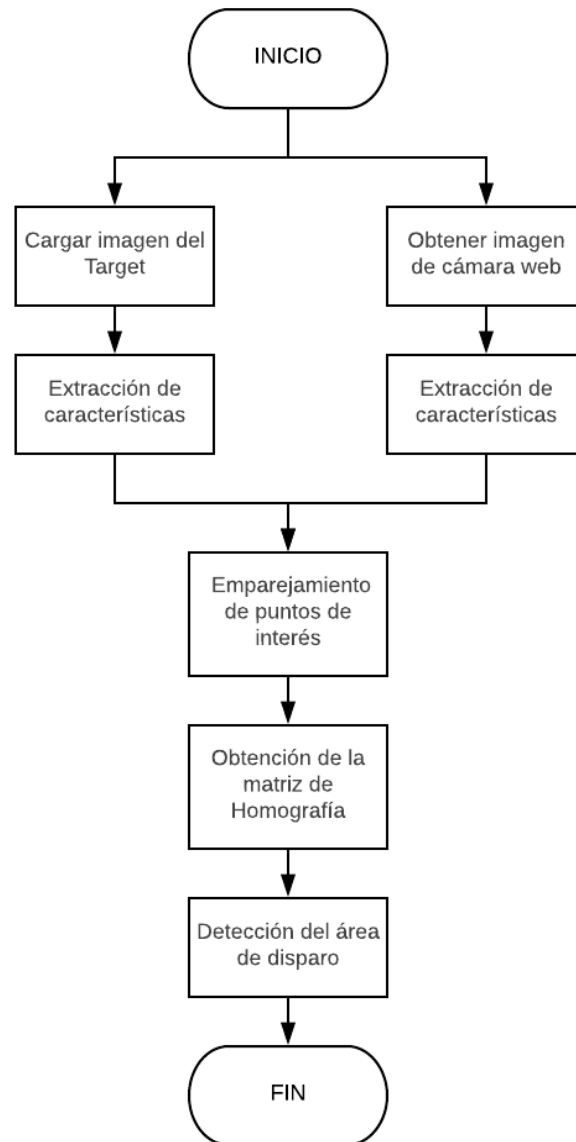
En esta sección se describe la detección del área de disparo, mediante el procesamiento de imágenes, detección de puntos de interés y descripción de estos para su posterior emparejamiento y estimación de los bordes de la zona de tiro.

#### 4.1 Algoritmo para detección del área de disparo

Para el desarrollo del polígono virtual de tiro inicialmente es necesario realizar una calibración, en la cual se discrimina el área de disparo que se encuentra comprendida por la imagen generada por el proyector sobre una superficie. Una vez que se han colocado todos los equipos en su posición como se menciona en el Capítulo III, la cámara se encuentra en la capacidad de recibir la imagen del ambiente el cual no solo contiene el área de disparo sino también parte del entorno controlado.

En la Figura 26 se puede observar el algoritmo creado para la discriminación y detección de la zona de disparo en la imagen entregada por la cámara. Para este proceso se requiere de una imagen (target) que cuente con puntos de interés notorios, la misma que al ser proyectada pueda ser reconocida por el sistema. En este proceso la extracción de características y su descripción juegan un papel importante ya que a partir de esto se procederá con el emparejamiento de puntos de interés entre el target y la imagen obtenida por la cámara. A continuación, se realizará la obtención de la matriz de homografía que cuenta con los parámetros de rotación del target proyectado y finalmente se puede reconocer el área de disparo entre los fotogramas que genera la cámara. A continuación,

se realizará la obtención de la matriz de homografía que cuenta con los parámetros de rotación del target proyectado y finalmente se puede reconocer el área de disparo entre los fotogramas que genera la cámara.

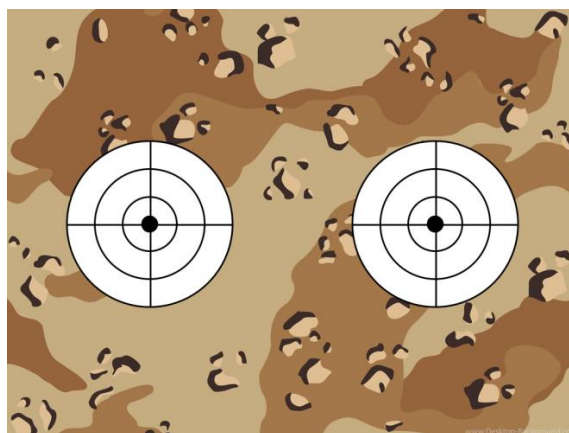


**Figura 26.** Algoritmo para detección de la zona de disparo

Cada etapa se detalla concretamente para conocer a profundidad como se desarrolla y funciona el proyecto.

## 4.2 Detección del área de disparo

Dentro de todo el campo visual de la cámara se obtiene una imagen con elementos no necesarios para el proyecto, por esto es necesario limitar el área de trabajo para identificar claramente la región de interés. Ha sido necesario diseñar una imagen (target) la cual al ser proyectada y capturada por la cámara presente una distorsión de colores mínima y una resolución alta de sus formas y regiones. La Figura 27 presenta el target que se usara para la detección de área de disparo.



**Figura 27.** Imagen utilizada para la detección de la zona de interés (target)

El target proyectado sobre una superficie se mantendrá fijo mientras se ejecutan los algoritmos correspondientes para detectar los puntos de interés. Se han desarrollado dos métodos para la detección: de forma manual y de forma automática. En ambos casos se muestran 4 puntos los cuales corresponden a las esquinas de la imagen. Cuando se realiza una configuración manual estos puntos pueden ser reubicados fácilmente con el puntero del mouse y colocarse en el lugar

indicado. Sin embargo, para una identificación automática del target se han implementado algoritmos de detección, descripción y emparejamiento de puntos de interés que junto con el uso de homografía identifican adecuadamente la zona de disparo.

#### **4.2.1 Detección y descripción de los puntos de interés**

La detección de características locales es necesario para la identificación del target dentro de la imagen obtenida por la cámara (Aguilar & Angulo, Robust video stabilization based on motion intention for low-cost micro aerial vehicles, 2014) (Aguilar & Angulo, Estabilización de vídeo en micro vehículos aéreos y su aplicación en la detección de caras, 2014) (Aguilar & Angulo, Estabilización robusta de vídeo basada en diferencia de nivel de gris, 2013) (Aguilar & Angulo, Compensación de los efectos generados en la imagen por el control de navegación del robot Aibo ERS 7, 2012) (Aguilar & Angulo, Compensación y aprendizaje de efectos generados en la imagen durante el desplazamiento de un robot, 2012). En este trabajo se han implementado tres de los algoritmos de detección y descripción que brindan una mayor robustez: SURF, ORB, SIFT. Además, se ha realizado combinaciones entre estos algoritmos como es SURF-ORB y ORB-SURF, el primer método se refiere a la detección de características y el segundo a su correspondiente descripción.

Para el análisis en la detección de puntos de interés en cada algoritmo se han utilizado cinco fotografías tomados del target proyectado dentro del entorno controlado, ubicando la cámara desde diferentes ángulos. Como resultado obtenemos la cantidad de puntos de interés encontrados y el tiempo que tardo en encontrarlos. A continuación, en la Tabla 2 se pueden apreciar los resultados

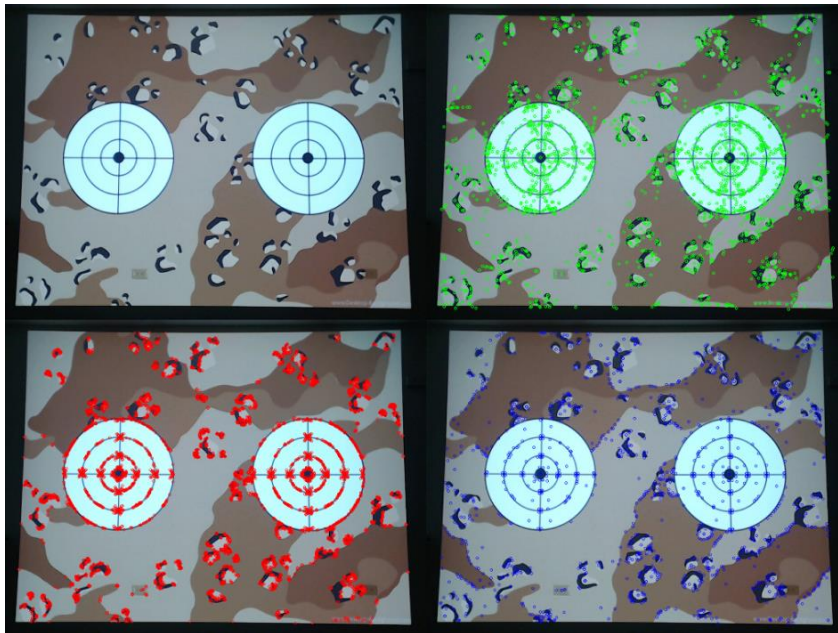
obtenidos. A partir de estos datos podemos destacar que ORB logra obtener una mayor cantidad de puntos de interés en menor tiempo a comparación de SURF y SIFT, entonces se puede mencionar que resulta más efectivo en cuanto a la detección de características dentro de una imagen.

**Tabla 2**

*Comparación de rendimiento de algoritmos de detección de puntos de interés*

Imagen	SURF		ORB		SIFT	
	Tiempo (s)	Puntos	Tiempo (s)	Puntos	Tiempo (s)	Puntos
1	1.08	2560	0.26	7725	2.77	2234
2	0.52	2373	0.18	6995	1.27	1538
3	0.69	1855	0.24	5444	2.47	1521
4	0.59	2175	0.18	6978	1.54	1208
5	0.45	1898	0.15	6124	1.14	1595

Adicionalmente en la Figura 28 se puede observar un ejemplo del funcionamiento de cada algoritmo para la detección de puntos de interés a partir de una imagen obtenida por la cámara del target. Se puede apreciar que SURF (verde), ORB (rojo) y SIFT (azul) aplican sus métodos correspondientes como se analizó en el Capítulo II para la detección de características.



*Figura 28.* Superior izquierda: imagen obtenida por la cámara, Superior derecha: detección de características mediante SURF, Inferior izquierda: detección de características mediante ORB, Inferior derecha: detección de características mediante SIFT.

#### 4.2.2 Emparejamiento entre imágenes

Una vez detectado los puntos de interés en la imagen obtenida por la cámara web de la proyección del target sobre una superficie, es necesario también procesar la imagen del target para de esta manera poder determinar puntos de interés en común entre ambos y establecer la correspondencia entre estos. En esta etapa se han implementado cinco algoritmos: SURF, ORB, SIFT, ORB-SURF y SURF-ORB, en las tres iniciales tanto la detección de características como su respectiva descripción se realiza mediante el mismo método, sin embargo, en las restantes se combina, en el primer caso se realiza la detección mediante el algoritmo ORB y su descripción con SURF, el mismo procedimiento se realiza en SURF-ORB combinando los algoritmos. La



descripción de las características es un proceso de vital importancia debido que a partir de estas se realiza el emparejamiento de puntos entre ambas imágenes y así establecer las características coincidentes para proceder con la obtención de la matriz de homografía.

En el proceso de obtención de descriptores cada punto de interés genera un vector descriptor el cual representa las características del mismo. Estos vectores son una parte fundamental para emparejamiento de coordenadas ya que proceso consiste en el cálculo de la distancia vectorial entre estos al comparar los vectores establecidos por la imagen target con los obtenidos por la imagen de la cámara. Dos métodos son conocidos para el emparejamiento de imágenes: “Brute-Force” y “FLANN”. Por un lado, Brute-Force es idóneo para trabajar con descriptores basados en cadenas binarias como es ORB, BRIEF, etc., y se fundamenta en la distancia de Hamming como método de medición, en este método se toman los descriptores de la primera imagen (target) y se compara de uno en uno con los descriptores de la imagen obtenida por la cámara, para obtener esta distancia se utiliza la siguiente formula:

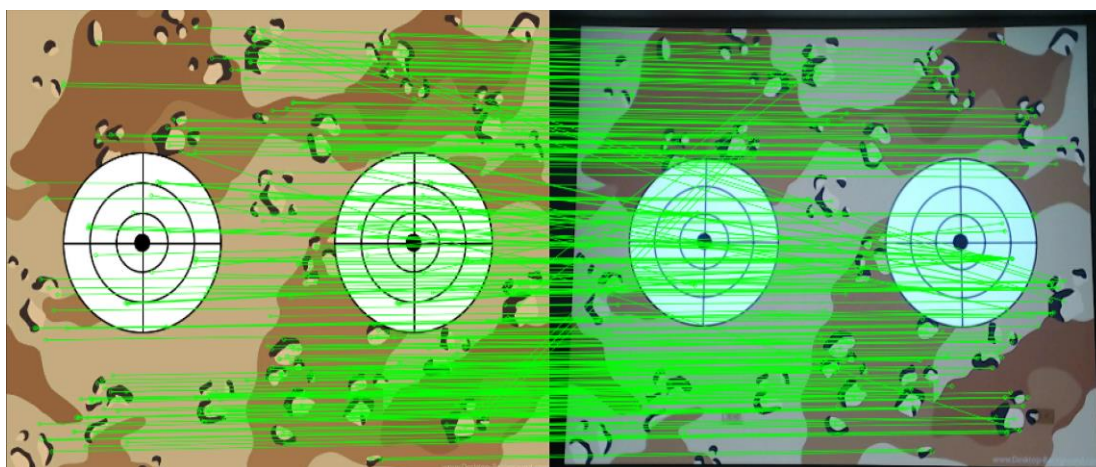
$$d(\vec{p}, \vec{q}) = \sum_i (p_i \neq q_i) \cdot 1 : 0 \quad (113)$$

En el caso de FLANN, este trabaja de mejor manera con SIFT y SURF. La distancia que se calcula entre los vectores se lo realiza en un espacio euclidiano y se define como:

$$d(\vec{p}, \vec{q}) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (124)$$

Cuando ya se han aplicado estos métodos se forman los nuevos pares de puntos los cuales debido a su corta distancia han sido seleccionados como puntos de correspondencia, adicionalmente se

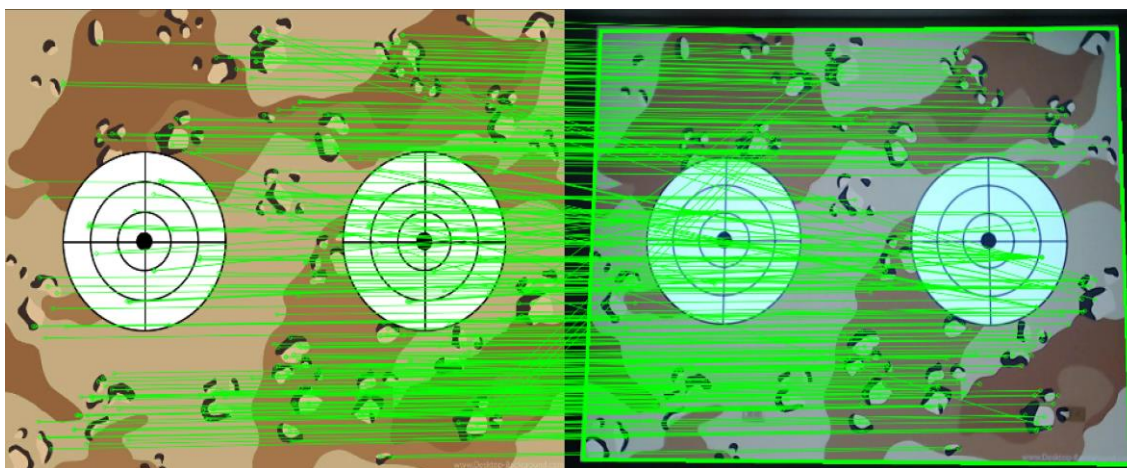
puede realizar un proceso adicional para reducir la cantidad de puntos de correspondencia y aumentar la certeza de que los puntos sean los correctos, en la Figura 29 se muestran un ejemplo del emparejamiento de imágenes. Mediante estos pares de puntos se determina la matriz de homografía la cual necesita mínimo cuatro puntos por cada imagen para ser calculada (Aguilar, Salcedo, Sandoval, & Cobeña, 2017) (Aguilar, Angulo, & Costa-Castello, Autonomous Navigation Control for Quadrotors in Trajectories Tracking, 2017) (Aguilar & Angulo, Real-Time Model-Based Video Stabilization for Microaerial Vehicles, 2016) (Aguilar, Angulo, Costa, & Molina, 2014) (Aguilar & Angulo, Real-time video stabilization without phantom movements for micro aerial vehicles, 2014). Una técnica usada en este proceso es RANSAC (Random Sample Consensus) que ayuda a descartar los puntos que no se ajusten con el modelo obtenido, manteniendo los pares de puntos más fiables.



*Figura 29.* Emparejamiento de imágenes

Para la obtención del área de disparo, se obtiene la matriz de homografía, la cual, al ser multiplicada por los valores de las esquinas de nuestro target obtenemos como resultado las

coordenadas aproximadas de las esquinas de nuestro target proyectado obtenido en la imagen de la cámara. Como se muestra en la Figura 30 los bordes detectados son lo más próximos a los bordes del target, esto varía dependiendo del algoritmo que se aplique.

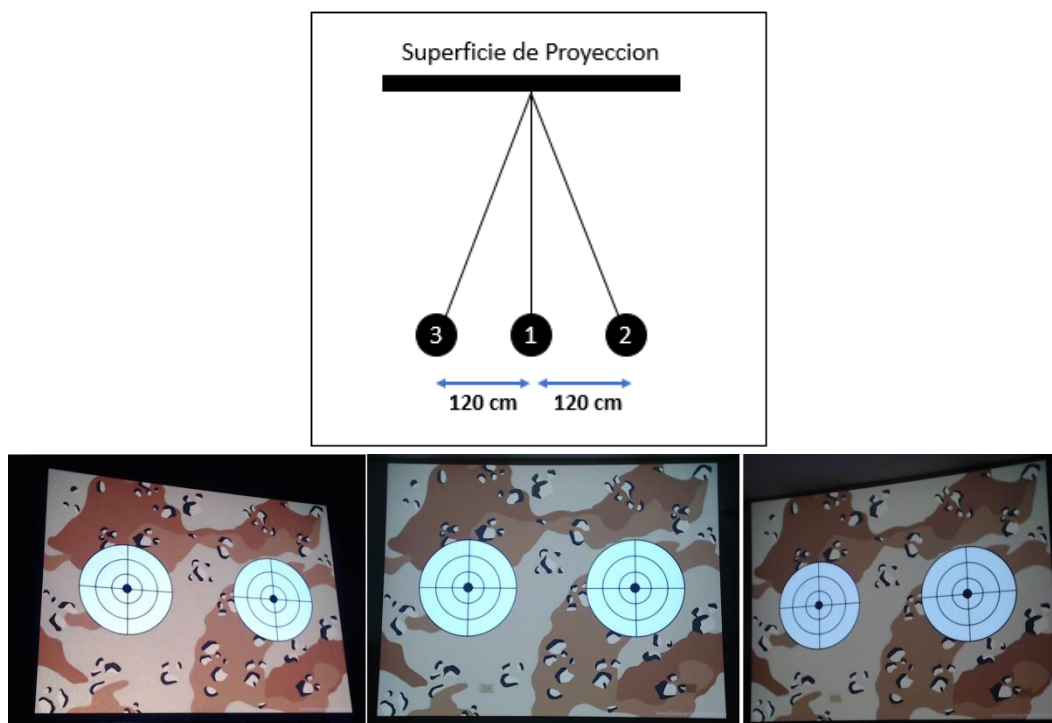


*Figura 30.* Detección del área de disparo

#### 4.2.3 Comparación de rendimiento de algoritmos implementados

Ya implementados los algoritmos es necesario comprobar la eficiencia de cada uno de ellos, puestos a prueba en tres escenarios, cada escenario es una ubicación diferente de la cámara con respecto a la proyección como se puede apreciar en la Figura 31, se describen de la siguiente manera:

- Posición 1: frente al área de proyección separada por 4 metros.
- Posición 2: desplazamiento de 1.2 metros hacia la derecha con respecto a la posición 1.
- Posición 3: desplazamiento de 1.2 metros hacia la izquierda con respecto a la posición 1.



**Figura 31.** Escenarios para pruebas de funcionamiento de algoritmos implementados

Adicionalmente se ha considerado la iluminación externa como uno de los principales parámetros que intervienen en el rendimiento de los algoritmos empleados debido al ruido que pueden generar en el momento de captura de imágenes. Debido a esto se han establecido tres posibles escenarios que se pueden dar dentro del ambiente controlado que se ha creado para este proyecto. Como se mencionó anteriormente dentro de la instalación existe una ventana y una puerta las cuales permiten el ingreso de luz a la habitación, se usarán estas para realizar respectivas pruebas, los escenarios propuestos serán de acuerdo a la cantidad de iluminación que se genera dentro de la habitación, se ha considerado: baja, media y alta de acuerdo a mediciones realizadas cuando todas las fuentes de luz del lugar se encarnas apagadas y se han cubierto las entradas de iluminación externa. A continuación, se presentan los escenarios posibles:

**Tabla 3**

*Escenarios de iluminación propuestos para realización de pruebas de funcionamiento.*

Iluminación	Descripción	Lúmenes
Baja	Aislamiento total de iluminación externa	0 (+10%)
Media	Aislamiento parcial de iluminación externa	20(±10%)
Alta	Sin aislamiento	40(±10%)

Propuestas todas las condiciones necesarias se han realizado pruebas de funcionamiento para determinar las coordenadas estimadas de las esquinas del target proyectado, además del tiempo transcurrido, con las siguientes consideraciones:

- Se ejecutan 10 pruebas de funcionamiento por cada algoritmo.
- Cada algoritmo se aplicará sobre cada posición establecida.
- Cada posición se lo realiza bajo cada tipo de iluminación.

La métrica que se ha utilizado para determinar la precisión de cada algoritmo implementado es el cálculo del error entre las esquinas reales del target en la imagen obtenida por la cámara y las esquinas detectadas en cada algoritmo. Para realizar este procedimiento, inicialmente y de forma manual se asignan los píxeles que se encuentran en las esquinas para posteriormente ejecutar la detección, descripción emparejamiento y cálculo de la matriz de homografía, de esta manera obtener las coordenadas estimadas de las esquinas. Para determinar la precisión de cada algoritmo es necesario calcular la diferencia que existe entre los valores reales de las coordenadas de cada esquina con los calculados mediante cada algoritmo, estos errores son calculados tanto en el eje X como en el eje Y, y se emplea la función de Error Cuadrático Medio (MSD):

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (135)$$

Donde:  $n$  es el número de elementos,  $Y$  es un vector con los valores reales de cada esquina, y  $\hat{Y}$  es un vector con los valores calculados.

El valor de error obtenido en cada caso se encuentra dado por la variación de pixeles, para brindar información con mayor relevancia se transformará estos datos a centímetros, lo cual permitirá que el error sea más fácil de comprender. Para realizar esto se ha obtenido el equivalente entre centímetros y pixeles tomando información ya conocida como es la dimensión del área proyectada (284cm x 212cm) y calculando su equivalente en función de la cantidad de pixeles que ocupa en la imagen (691 pixeles x 504 pixeles). Se ha obtenido que 1 cm = 2.4 pixeles o 1 pixel = 0.4 cm.

A continuación, se presentan los resultados obtenidos, se lo ha dividido en función de la posición en la cual se ha colocado la cámara mediante tres diferentes tablas, en cada una se presentan los promedios de los datos obtenidos durante 10 ejecuciones en los tres niveles de iluminación, los promedios vienen dados en el eje X y eje Y, además se considera el promedio de ejecución de cada algoritmo.

**Tabla 4**

*Resultados de algoritmos con la cámara se encuentra en la posición 1.*

Posición	Iluminación	Algoritmo	Error en X (píxeles)	Error en X (cm)	Error en Y (píxeles)	Error en Y (cm)	Tiempo ejecución
1	Baja	SURF	5,475	2,250	7,450	3,134	0,306
		ORB	9,425	3,874	11,375	4,785	0,056
		SIFT	4,150	1,706	6,575	2,766	0,490
		OR-SU	7,725	3,175	7,650	3,218	0,127
		SU-OR	>100	>40	>100	>40	0,234
	Media	SURF	2,425	0,997	2,775	1,167	0,313
		ORB	8,400	3,452	4,475	1,882	0,048
		SIFT	2,200	0,904	2,300	0,967	0,477
		OR-SU	4,150	1,706	2,475	1,041	0,138
		SU-OR	>100	>40	>100	>40	0,198
	Alta	SURF	3,225	1,325	9,275	3,901	0,306
		ORB	11,050	4,542	20,775	8,739	0,052
		SIFT	3,475	1,428	8,800	3,702	0,469
		OR-SU	9,825	4,038	15,700	6,604	0,132
		SU-OR	>100	>40	>100	>40	0,204

**Tabla 5**

*Resultados de algoritmos con la cámara se encuentra en la posición 2.*

Posición	Iluminación	Algoritmo	Error en X (píxeles)	Error en X (cm)	Error en Y (píxeles)	Error en Y (cm)	Tiempo ejecución
2	Baja	SURF	6,225	2,558	5,175	2,177	0,353
		ORB	11,850	4,870	8,900	3,744	0,058
		SIFT	5,050	2,076	3,125	1,314	0,558
		OR-SU	9,850	4,048	5,675	2,387	0,144
		SU-OR	>100	>40	>100	>40	0,228
	Media	SURF	2,350	0,966	2,325	0,978	0,314
		ORB	7,075	2,908	4,850	2,040	0,062
		SIFT	2,400	0,986	1,650	0,694	0,553
		OR-SU	7,200	2,959	3,700	1,556	0,154
		SU-OR	>100	>40	>100	>40	0,224

**CONTINÚA**

Alta	SURF	5,575	2,291	7,950	3,344	0,328
	ORB	10,500	4,315	13,800	5,805	0,058
	SIFT	4,925	2,024	6,825	2,871	0,561
	OR-SU	12,400	5,096	11,775	4,953	0,147
	SU-OR	>100	>40	>100	>40	0,228

**Tabla 6**

*Resultados de algoritmos con la cámara se encuentra en la posición 3.*

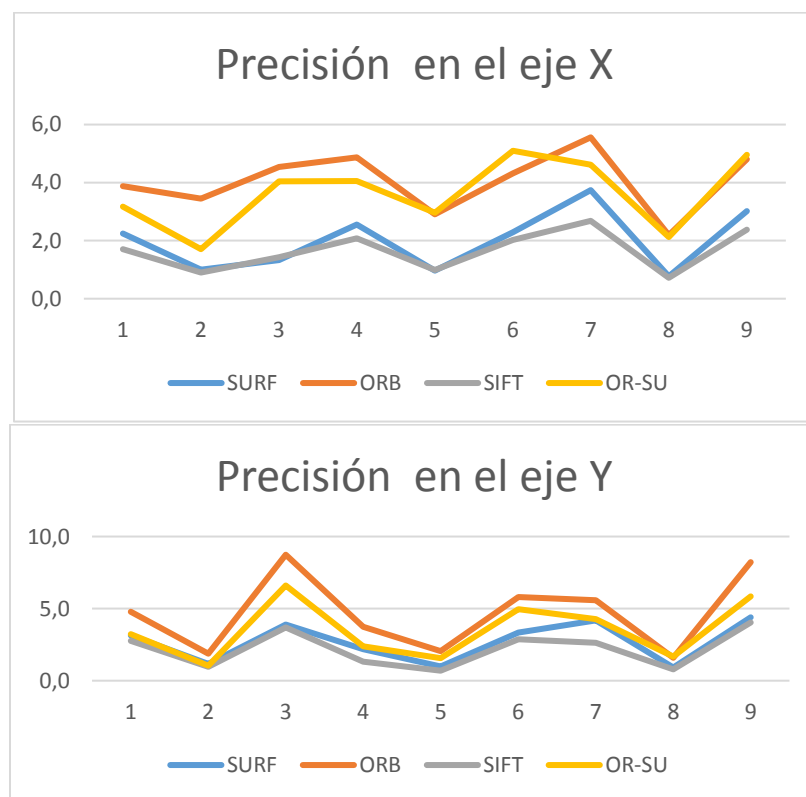
Posición	Iluminación	Algoritmo	Error en X (píxeles)	Error en X (cm)	Error en Y (píxeles)	Error en Y (cm)	Tiempo ejecución
3	Baja	SURF	9,100	3,740	9,950	4,185	0,326
		ORB	13,525	5,559	13,250	5,573	0,052
		SIFT	6,525	2,682	6,250	2,629	0,541
		OR-SU	11,225	4,613	10,175	4,280	0,135
		SU-OR	>100	>40	>100	>40	0,225
	Media	SURF	1,875	0,771	2,175	0,915	0,328
		ORB	5,325	2,189	3,775	1,588	0,055
		SIFT	1,750	0,719	1,875	0,789	0,552
		OR-SU	5,175	2,127	4,000	1,683	0,158
		SU-OR	>100	>40	>100	>40	0,220
	Alta	SURF	7,350	3,021	10,450	4,396	0,332
		ORB	11,700	4,809	19,550	8,223	0,056
		SIFT	5,775	2,374	9,600	4,038	0,540
		OR-SU	12,075	4,963	13,900	5,847	0,139
		SU-OR	>100	>40	>100	>40	0,225

Como se muestra en las tablas anteriores, podemos mencionar que SIFT presenta un promedio de error en X y Y menor en comparación con el resto a pesar de que el tiempo de ejecución no sea el mínimo, por otro lado, SURF mantiene una precisión con un error ligeramente inferior que SIFT; y ORB se ejecuta en un tiempo menor que todos, pero la cantidad de correspondencias es la menor por tanto su exactitud disminuye. En los casos de combinación de algoritmos se observa que ORB-SURF trabajan adecuadamente en conjunto a pesar de no poseer el mínimo error, por el contrario,



SURF-ORB en este caso no presento los resultados esperados debido a que el descriptor no puedo realizar un trabajo adecuado con los puntos de interés detectados. En función de los datos obtenidos se ha decidido continuar el desarrollo del proyecto con el algoritmo SIFT.

Para el desarrollo de este proyecto, en esta etapa el tiempo de ejecución no es tan relevante considerarlo debido a que se da mayor prioridad a la precisión ya que nuestro target captado por la cámara no se encuentra en movimiento sino es indispensable la exactitud del reconocimiento del área de trabajo. En la Figura 32 se presentan gráficamente los datos de las tablas anteriores, mostrando los promedios de error en centímetros de X y Y de los nueve escenarios posibles.



**Figura 32.** Precisión de los algoritmos implementados

## CAPÍTULO V

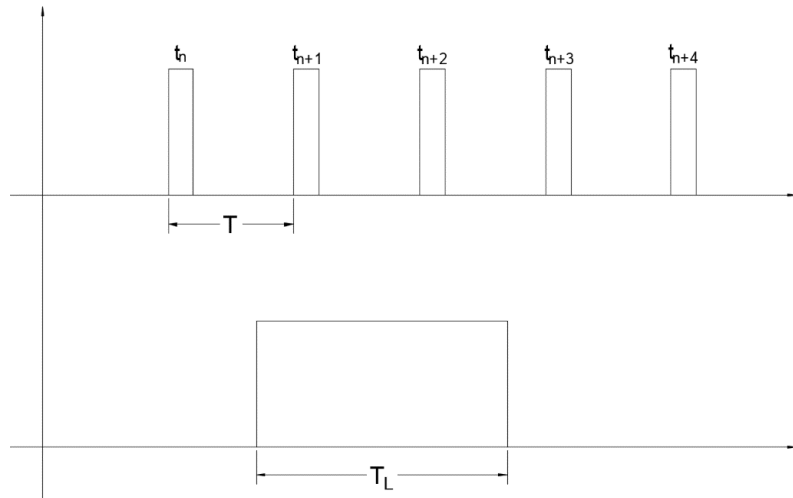
### DETECCIÓN DEL DISPARO LÁSER

En este capítulo se establecen varias consideraciones sobre el dispositivo laser a ser empleado, además se establece y desarrolla el algoritmo para la detección del haz del disparador laser; se describe el proceso de virtualización del disparo para insertarlo dentro de un escenario virtual

#### 5.1 Consideraciones

Para continuar con el desarrollo del proyecto es necesario analizar el evento en el cual es captado el destello laser por la cámara web. Tomando las características del dispositivo mencionadas en el Capítulo III, se puede considerar que la cámara web mantiene una captura de imágenes con un promedio de 30 FPS. Al realizar pruebas en el entorno Unity se ha determinado que la cámara recepta entre 25 a 30 fotogramas por segundo, lo cual implica un análisis sobre el tiempo que el láser debe ser captado para de esta manera comprobar que el arma a utilizar será efectiva.

En la Figura 33 se puede observar un diagrama en el cual se muestran los ciclos de captura de imagen que realiza la cámara, es decir en cada uno de estos capta el ambiente y retorna un frame. Estos ciclos se ejecutan cada cierto periodo  $T$  y cada uno se realiza en un cierto tiempo  $t$ . Además, se presenta el tiempo de exposición del destello laser, el cual se encuentra activo durante un periodo  $T_L$  tomando en cuenta estos datos, se determina el periodo en el cual será necesario que el láser se encuentre activo.



**Figura 33.** Ciclos de operación de la cámara y laser

Tomando el peor de los casos cuando la cámara recepte 25 FPS:

$$T = \frac{1}{25} = 0.04 \text{ s} = 40 \text{ ms} \quad (146)$$

Considerando que el láser debe ser captado durante al menos dos ciclos consecutivos y se puede esperar que en el peor de los casos durante dos ciclos no sea receptado entonces el láser deberá estar encendido durante al menos cuatro ciclos:

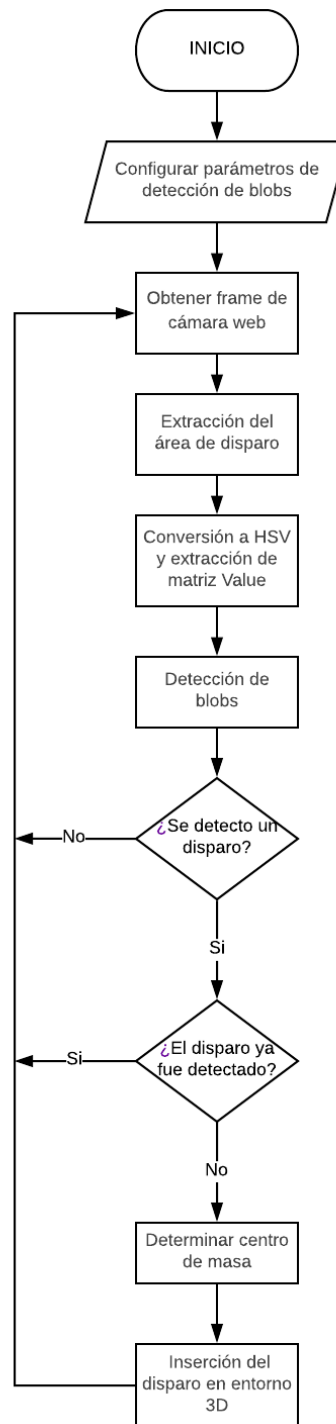
$$T_L = 4 * T = 160 \text{ ms} \quad (157)$$

Aplicando el mismo concepto cuando la cámara recepta 30 FPS:

$$T_L = 133 \text{ ms} \quad (168)$$

Se puede determinar que el tiempo de exposición del láser debe ser mayor que 160 ms, sin embargo, se ha considerado establecerlo en 200 ms tomando en cuenta las posibles fallas.

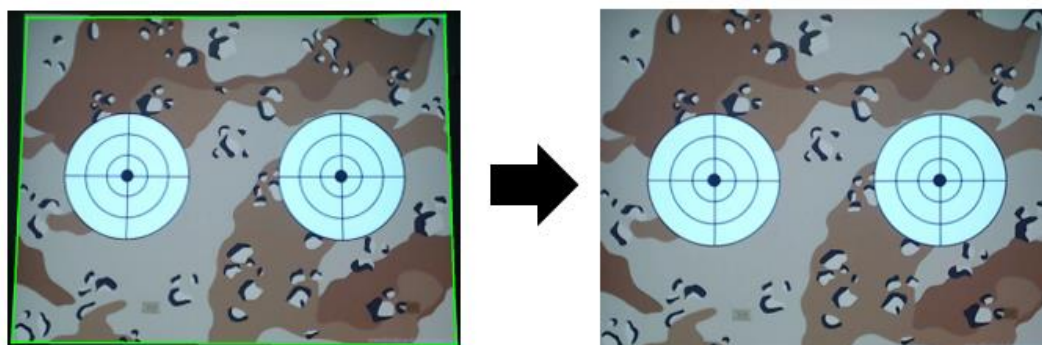
## 5.2 Algoritmo de detección de laser



El desarrollo de este algoritmo nos permitirá determinar la dirección en la cual se realiza el disparo y asignar una calificación. En este proceso el fundamento es la detección de blob (manchas) que se presentan en la matriz de la cámara, por lo cual es necesario establecer una serie de parámetros iniciales que permitan identificar claramente el destello del láser sobre la proyección. La adquisición de imágenes será en tiempo al igual que su procesamiento. Se inicia obteniendo la imagen de la cámara y adicionalmente se extrae el área de trabajo la cual se determina en el Capítulo IV, esta nueva matriz es transformada de su espacio de colores original (RGB) a HSV en el cual se puede seleccionar la matriz de Value para trabajar sobre ella. A continuación, se realiza la detección de blobs, si el disparo es detectado se procede con el cálculo del centro de masa de este y finalmente se lo incorpora dentro de un entorno virtual.

### **5.3 Extracción del área de disparo**

Como se vio en el capítulo anterior es de vital importancia establecer el área de disparo para lo cual mediante el uso del algoritmo SIFT podemos realizar esta tarea. Una vez obtenidos los puntos correspondientes a las esquinas del target proyectado se procede a realizar una transformación geométrica para extraer el área en una nueva matriz tal y como muestra la Figura 34, esta nueva matriz posee otras dimensiones, su ancho es el valor mayor entre las longitudes de las líneas horizontales graficadas en la matriz de detección y el alto corresponde al mayor entre la longitud de las líneas verticales.



*Figura 34.* Extracción del área de disparo

#### 5.4 Conversión de espacio de colores

Las imágenes adquiridas por la cámara web originalmente está formada de tres matrices unidas, cada una de ellas está constituida por un cierto filtro. En este caso las imágenes vienen en espacio de colores RGB lo cual significa que cada matriz representa la intensidad de cada color en la imagen, es decir, en la matriz R los valores de cada pixel constituye la intensidad de rojo, la matriz G en verde y la matriz B en azul. Lo mismo sucede en otros espacios de colores, para este proyecto ha sido necesario la conversión de RGB a HSV, de esta manera podemos obtener tres matrices Hue, Saturation y Value, de las cuales Value es la necesaria para la detección del destello laser debido a que esta nos entrega un espectro de brillo que posee la imagen. Para la implementación del algoritmo se ha tomado una captura de un destello laser verde sobre una proyección de color gris, a partir del cual se realizará la detección, en la Figura 35 se puede observar la imagen original del láser y las matrices H, S y V de la conversión entre espacios de colores, se puede observar que en V el espectro del láser resalta de mejor manera que en las otras.



*Figura 35.* Superior Izquierda: imagen original,  
Superior Derecha: matriz Hue,  
Inferior Izquierda: matriz Saturation,  
Inferior Derecha: matriz Value.

## 5.5 Detección de blobs

Como se definió en el Capítulo II, un blob representa una mancha que no contiene puntos de interés como bordes o esquinas que nos facilite su detección, sin embargo, el reconocimiento de estos se lo puede realizar mediante varios parámetros de búsqueda. El uso de este método en el proyecto se debe a que las formas generadas por el láser se muestran como manchas dentro de la matriz Value, y debido a la constitución del láser, las manchas siempre mantienen una forma de tipo circular convexa que sobresalen por su color brillante, además, al usar el mismo tipo de laser a la distancia establecida para la práctica de tiro proyecta un destello sobre la imagen un área que siempre estará dentro del mismo rango de valores.

La detección de blobs se lo realiza mediante una serie de filtros en el cual se pueden configurar ciertos valores los cuales permiten identificar el disparo de entre todo el entorno captado por la cámara, los filtros mantiene parámetros configurables son los siguientes:

**Tabla 7**

*Parámetros para la detección de blobs*

Parámetro	Descripción	Rango de valores
Umbral	Convierte la imagen original en una serie de imágenes binarias manteniendo las manchas más notorias	Umbral mínimo: 0-255 Umbral máximo: 0-255
Repetibilidad	Las veces que el blob se repite dentro de la imagen	1-Valor deseado
Color	La intensidad que posee la mancha	0 – 255 0 = oscuro, 255 =claro
Área	La superficie que cubre el blob	Valores deseados en pixeles
Circularidad	Establece la forma de la mancha y que tan parecida es a un círculo.	0 – 1 1 = círculo 0.785= cuadrado 0 – 1
Inercia	Mide que tan alargada es la forma de mancha	0 = línea 1 = círculo >0 y <1 = elipse 0 – 1
Convexidad	Define la convexidad o concavidad que posee el blob	0 = cóncava 1 = convexa



Tomando en cuenta todos los tipos de filtros y la silueta que forma el láser al ser captado por la cámara se han seleccionado únicamente cuatro parámetros para distinguir el láser de la imagen: umbral, circularidad, convexidad y área. Los rangos de valores que se les atribuye se los asignado debido a una serie de pruebas realizadas al accionar el láser sobre la proyección a una establecida distancia (4 m), y se consideran los siguientes valores:

**Tabla 8**

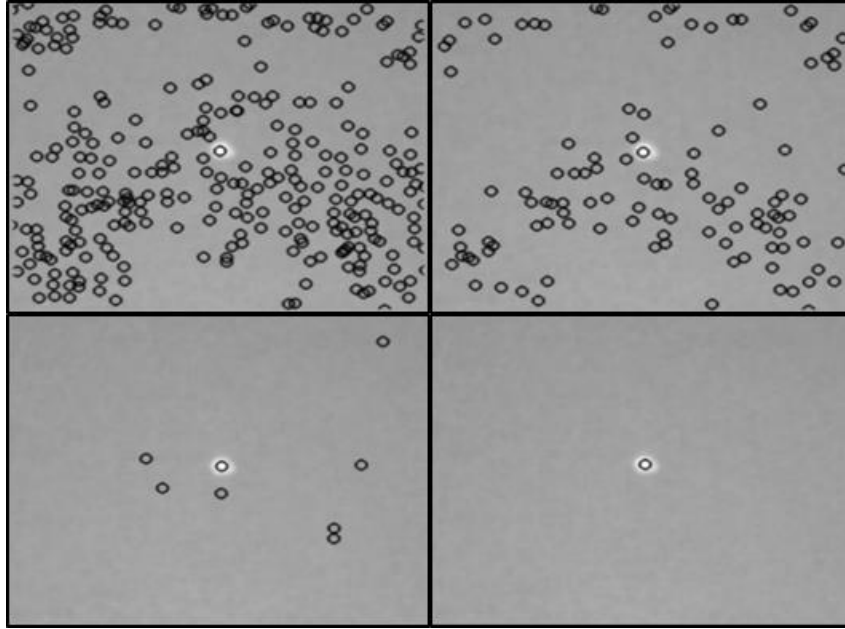
*Parámetros seleccionados para detección del disparo*

Parámetro	Valor
Umbral	Mínimo = 150, máximo = 255
Circularidad	Mínimo = 0.8, máximo = 1
Convexidad	Mínimo = 0.7, máximo = 0.9
Área	Mínimo = 85, máximo = 100

En este punto es necesario realizar un análisis de los valores que se han tomado para el parámetro del Área ya que se debe tomar en cuenta que para el desarrollo de este proyecto se ha utilizado un generador de laser en lugar de un arma laser. Para realizar el acople de cualquier arma laser se debe considerar el área que cubre el haz, ya que estos valores serán diferentes para cada tipo de armamento laser. La manera en la cual se ha determinado los valores mínimo y máximo para el área ha sido realizando una cuenta de los pixeles blancos sobre pixeles negros que deja el rastro del destello laser sobre la imagen de la cámara, para este proceso únicamente se debe colocar la cámara a la distancia señalada anteriormente (4 metros desde la superficie de proyección) y aislar la habitación de luz externa, posteriormente se obtiene una imagen binarizada de la cámara con un contador de valores de pixeles que poseen únicamente dos valores 0 (negro) y 255 (blanco), el

número de píxeles con un valor de 255 será el área que genera el destello del láser sobre la imagen obtenida por la cámara, cabe mencionar que en este proceso no se debe encender el proyector ya que puede dificultar este proceso. Este proceso es de total importancia cuando se desee incorporar un nuevo disparador laser que puede ser un arma, ya que varía la intensidad del destello.

Anteriormente en la Figura 35 se mostró la imagen original sobre la cual se aplicará el algoritmo junto con la matriz Value, a continuación, en la Figura 36 se presenta la aplicación de la detección de blobs y como va reaccionando el sistema ante cada filtro que se usa. Como se puede observar en la figura superior izquierda se detectan todos los blobs que se mantienen cuando se aplica una umbralización, en la parte superior derecha se aplica la detección por regiones circulares, posteriormente en la parte inferior izquierda se muestra el filtro de convexidad y en la figura inferior derecha la detección por área. Aunque en la imagen original a simple vista no se percibe otras manchas el algoritmo puede detectarlas y filtrarlas, de tal manera que cuando se aplique sobre un escenario en el cual existan más siluetas, podrá discriminar al laser de entre todas ellas.



**Figura 36.** Aplicación de filtros para detección de blobs

Una vez detectado el láser se procede con el cálculo del centro de masa, debido a que para la transición entre la proyección y el entorno virtual se lo realiza mediante una coordenada que será el mencionado centro del disparo, para determinar este punto primero es necesario encontrar el momento central de la imagen que viene dado por:

$$mu_{ji} = \sum_{x,y} (vector(x, y) \cdot (x - \bar{x})^j \cdot (y - \bar{y})^i) \quad (179)$$

Donde  $(\bar{x}, \bar{y})$  es el centro de masa:

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}} \quad (20)$$

Obtenido el centro de masa se puede establecer exactamente la posición del disparo, a continuación, se muestra en la Figura 37 la detección del láser, en la cual se lo identifica por una circunferencia graficada con centro en las coordenadas del centro de masa de la mancha producida.

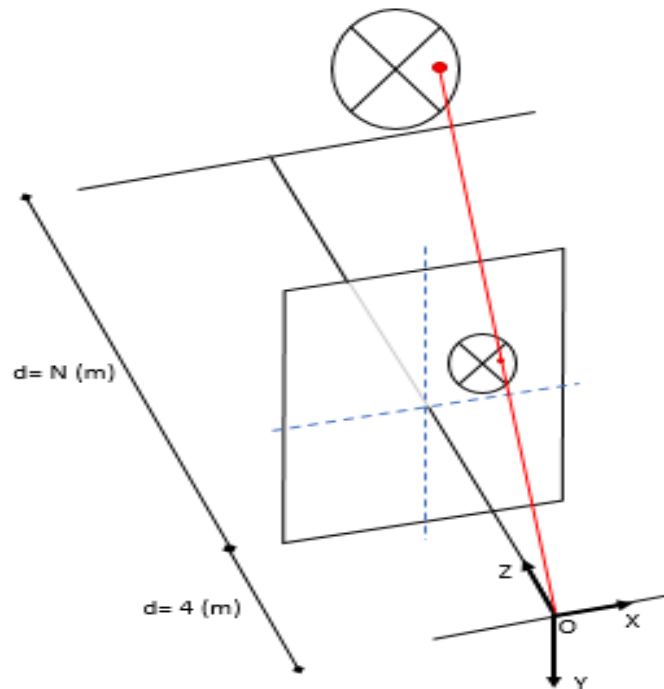


*Figura 37.* Reconocimiento del disparo laser

## 5.6 Virtualización del disparo

Una vez que se ha obtenido las coordenadas exactas del disparo es necesario introducirlo dentro del entorno virtual, para esto lo que se debe realizar es la recreación del escenario virtual con las mismas dimensiones, Unity brinda una gran ventaja en la creación de entornos 3D ya que un metro real es igual a un valor de una unidad en el editor, de esta forma se pueden recrear modelos con las mismas medidas que las reales. Una vez que el escenario está listo con los objetos en las posiciones establecidas inicialmente, se asigna el origen que en este caso es el lugar donde se encuentra parada la persona que realizara el disparo. La manera de realizar la virtualización del disparo es la construcción de un vector en tres dimensiones, denominado  $D$ , este vector brinda una dirección desde el origen que es el lugar del cual sale el disparo y podemos asignar una trayectoria. En la Figura 38 se puede observar una recreación de este proceso, en este caso conocemos cuales son los

valores de nuestro vector  $D$  ya que conocemos que son cuatro metros de distancia desde donde se realiza el disparo además conocemos los valores en  $x$  y  $y$  del destello captado por la cámara en el reconocimiento del disparo. Gracias a uno de los componentes que brinda Unity, denominado RayCast, podemos generar un rayo que parte desde el origen en dirección del vector  $D$  (color rojo en Figura 38) y permite detectar el contacto con otros objetos. De esta forma conocemos exactamente cuál fue la figura o pieza virtual de las siluetas alcanzadas por el disparo.

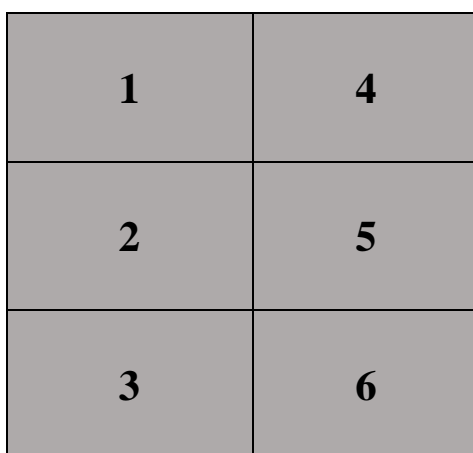


*Figura 38.* Virtualización del disparo laser

## 5.7 Pruebas de funcionamiento del algoritmo

Para determinar el error que posee el algoritmo para detectar un disparo se ha creado un escenario en el cual se mantiene proyectado una imagen completamente gris y dividida en 6

regiones, con el fin de estimar el error del algoritmo a lo largo de la zona de disparo. En este caso y considerando el proceso de virtualización del disparo, se considera que nuestra silueta se encuentra a cuatro metros, de esta el reconocimiento del disparo se lo puede asimilar como que se realiza sobre la superficie de proyección. En la Figura 39 se puede observar la imagen proyectada en el ambiente controlado.



**Figura 39.** Imagen proyectada para pruebas de funcionamiento del algoritmo de detección del disparo laser

Al igual que en el Capítulo IV se considera los niveles de iluminación externa debido a que son escenarios reales que se presentan dentro de este ambiente controlado. Las pruebas de funcionamiento se realizaron al ejecutar 10 disparos laser en cada región y esto en los tres niveles de iluminación. Cabe recalcar que en cada disparo detectado el sistema grafica una señal en las coordenadas estimadas. Para calcular los errores de los disparos ha sido necesario emplear una segunda cámara montada sobre la primera, la cual nos permita realizar una grabación de los disparos ejecutados. En el post procesamiento, se han extraído los frames de los disparos del video

justamente cuando el destello laser se produce y el instante en el cual el sistema nos señala la posición estimada del disparo. En ambas imágenes se extrae las coordenadas tanto del láser real y del disparo detectado y se procede a realizar un cálculo del error tanto en X como en Y, y posteriormente se realiza el cálculo del error en centímetros obteniendo el equivalente (centímetros - pixeles) de la misma forma que se realiza en el Capítulo IV, es decir, 1 cm = 2.4 pixeles o 1 pixel = 0.4 cm. Se tomaron mediciones en 180 disparos de los cuales se muestran los resultados a continuación:

**Tabla 9**

*Resultados de pruebas de funcionamiento en iluminación baja*

Iluminación		Baja				
Región	1	2	3	4	5	6
<b>Error en X (pixeles)</b>	4,60	4,90	3,80	4,40	4,20	4,60
<b>Error en X (cm)</b>	1,89	2,01	1,56	1,81	1,73	1,89
<b>Error en Y (pixeles)</b>	6,00	4,40	2,40	2,00	1,80	3,90
<b>Error en Y (cm)</b>	2,52	1,85	1,01	0,84	0,76	1,64

**Tabla 10**

*Resultados de pruebas de funcionamiento en iluminación media*

Iluminación		Media				
Región	1	2	3	4	5	6
<b>Error en X (pixeles)</b>	3,50	3,40	5,80	3,30	5,20	7,40
<b>Error en X (cm)</b>	1,44	1,40	2,38	1,36	2,14	3,04
<b>Error en Y (pixeles)</b>	10,30	3,80	3,50	2,00	0,80	2,40
<b>Error en Y (cm)</b>	4,33	1,60	1,47	0,84	0,34	1,01

**Tabla 11***Resultados de pruebas de funcionamiento en iluminación alta*

<b>Iluminación</b>	<b>Alta</b>					
<b>Región</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
<b>Error en X (píxeles)</b>	3,70	2,70	1,30	2,70	6,90	4,40
<b>Error en X (cm)</b>	1,52	1,11	0,53	1,11	2,84	1,81
<b>Error en Y (píxeles)</b>	13,80	4,20	6,60	2,20	4,20	3,80
<b>Error en Y (cm)</b>	5,80	1,77	2,78	0,93	1,77	1,60

Observando los datos que obtuvieron, se puede observar que los valores no se encuentran alejas, por el contrario, en promedio no superan los 2.5 centímetros de error, excepto en casos específicos en el eje Y en la región 1. Además, los resultados en el nivel de iluminación baja presentan un error menor en comparación al resto, por esto se recomienda trabajar de esta manera, aislando al entorno de las fuentes de iluminación externas.



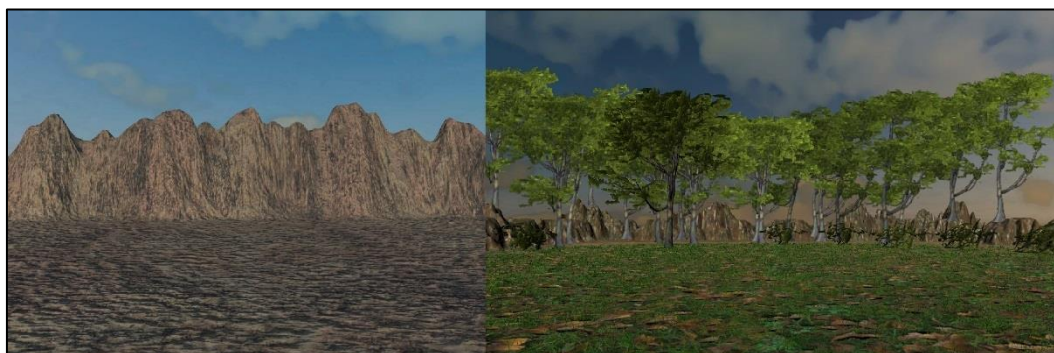
## CAPÍTULO VI

### DESARROLLO DEL SISTEMA DE CALIFICACIÓN

En este capítulo se presenta el resultado de la implementación de los algoritmos desarrollados anteriormente, así como, el desarrollo de escenarios y siluetas 3D que dan lugar a los escenarios virtuales, los menús del sistema y la generación del reporte de calificaciones.

#### 6.1 Implementación de entornos virtuales

Unity brinda amplia gama de opciones para el desarrollo de escenarios virtuales, para la implementación del polígono virtual de tiro, se han creado dos entornos basados en los requerimientos del personal del Centro de Investigación Científica y Tecnológica del Ejército (CICTE). En la Figura 40 se presenta a la izquierda una representación de un paredón, que es un lugar común para realizar prácticas de tiro reales, y en la parte derecha se muestra un entorno selvático el cual se enfoca en una escena real de guerra.



*Figura 40.* Escenarios desarrollados para el polígono virtual de tiro

Las siluetas virtuales que se han elaborado para este proyecto son modelos virtuales de las formas más comunes de siluetas con la cuales se realiza prácticas de tiro reales, además se han elaborado modelos 3D de soldados en diferentes posiciones de combate con el objetivo de generar una mayor sensación de inmersión en el usuario, en la Figura 41 se observan las siluetas elaboradas que se colocan dentro de los escenarios virtuales.



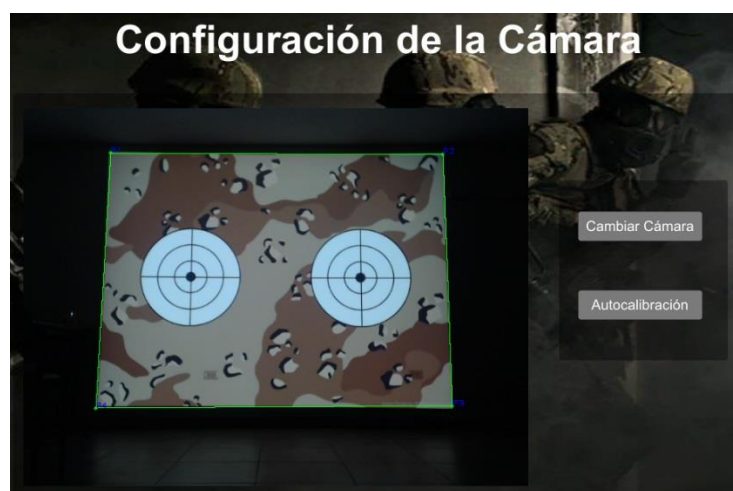
**Figura 41.** Siluetas desarrolladas para el polígono virtual de tiro

Cada silueta se encuentra personalizada para asignar una calificación dependiendo de la parte en la cual ha sido alcanzada por el disparo laser. En el caso de los círculos concéntricos el valor se va incrementando de 1 a 10 conforme se va acercando al centro, para las siluetas del personal militar todas mantienen la misma valoración en todas sus partes que es de 5 puntos, y en el caso de los popers (siluetas humanas) varía de una a otra conforme la posición del cuerpo de ha sido afectada.

## 6.2 Implementación del software del polígono virtual de tiro

El software del polígono virtual de tiro comprende todo lo que se ha desarrollado hasta este punto del proyecto, iniciando por la configuración y detección del área de disparo, los menús de configuración de las prácticas de tiro, los escenarios y siluetas virtuales, la calificación de disparos en tiempo real y la generación de un reporte de calificaciones de la práctica.

Se inicia el proceso con el menú de configuración de la cámara el cual nos permite primeramente alternar entre todas las cámaras reconocidas por el computador para seleccionar la deseada, y mediante un botón se realiza la detección automática del área de disparo, que cuenta con el algoritmo implementado en el Capítulo IV, en la Figura 42 se presenta es menú correspondiente a la configuración de la cámara.



**Figura 42.** Menú de configuración de la cámara

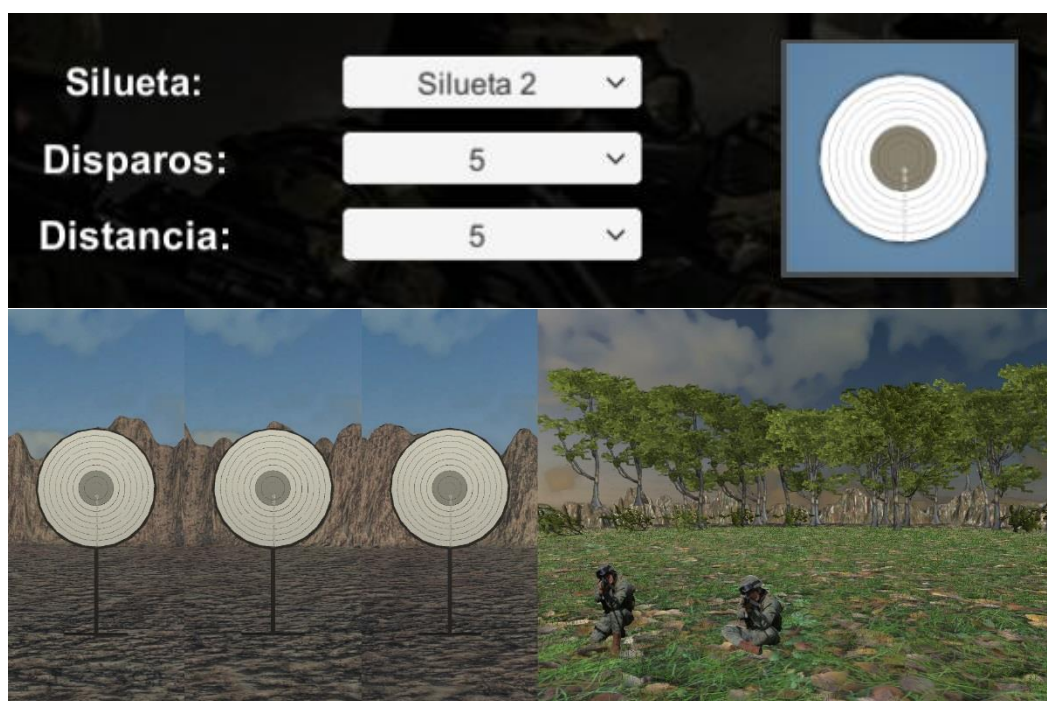
A continuación, se realiza la configuración de los participantes en la práctica de tiro, asignando la cantidad y el nombre de cada uno. Además, se puede seleccionar el tipo de practica deseada.

Aquí existen dos tipos, en el primer caso la práctica es de tipo libre, es decir solo realizara la calificación del escenario actual seleccionado, y en el segundo caso es un practica de tiro completa la cual acumula los puntos de todos los escenarios de tiro realizados. En la Figura 43 se presenta el menú de configuración de la práctica de tiro.



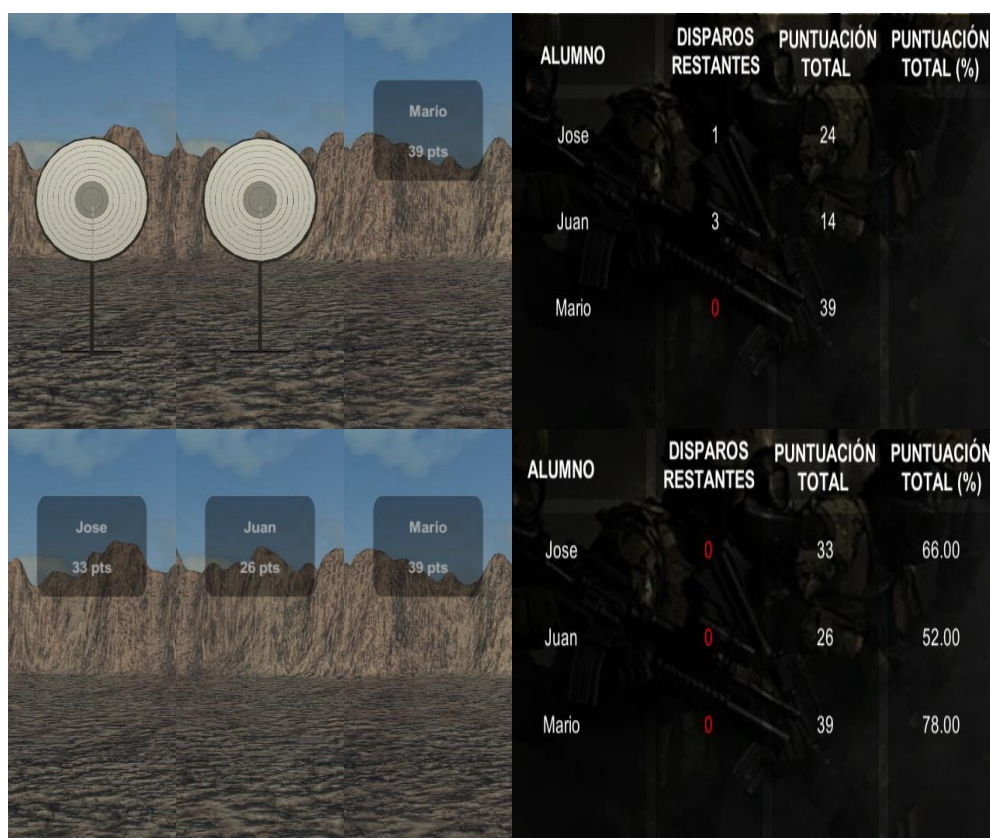
**Figura 43.** Menú de configuración de la práctica de tiro

Existen tres tipos de practica las cuales son: precisión, combate y avanzado. En cada una de estas las siluetas son diferentes. En el caso de precisión se trabajó con círculos concéntricos, en combate se utiliza los popers y siluetas humanas y en avanzado se incorporan siluetas de soldados reales. Indistintamente del tipo seleccionado se tiene el menú para la selección de silueta, el número de disparos y la distancia a la cual se encuentra la silueta. Como se muestra a continuación en la Figura 44 el menú general para todos los tipos de practica y como se incorpora las siluetas dentro del escenario virtual.



**Figura 44.** Selección de siluetas para escenarios virtuales

El proceso de calificación se lo realiza una vez que las siluetas virtuales aparecen, en la proyección se pueden observar únicamente los escenarios virtuales y en la parte del operador se observa el tablero de calificación que en tiempo real va sumando las calificaciones parciales de cada disparo, además muestra el número de disparos disponibles para cada usuario. Cuando en un carril se queda sin disparos se procede a eliminar la silueta virtual y mostrar la calificación total, cuando todos han terminado se genera un porcentaje de efectividad tomando en cuenta la cantidad de disparos realizados y la calificación que se asignó. En la Figura 45 se puede observar este proceso.



**Figura 45.** Calificación en tiempo real de disparos laser

Se ha creado una tabla general que contiene información de cada uno de los escenarios seleccionador para la práctica de tiro, en la Figura 46 se presentan los resultados obtenidos, se muestra inicialmente el tipo de practica seleccionado, en número de disparos realizados y la puntuación total en porcentaje que ha obtenido en ese escenario. Además, se puede agregar una descripción en cada uno por parte del operador. Los resultados finales se muestran en la parte inferior y se cuenta con el total de tiros realizados y el promedio de todas las practicas realizadas. Finalmente, estos valores son enviados a un archivo en Microsoft Excel el cual completa los campos correspondientes y permite configurar los campos de la instrucción tiro.

Registro de Calificación					
TIPO	DESCRIPCION	TIROS	Jose	Juan	Mario
Precisión	Familiarización	5	66.00 %	52.00 %	78.00 %
Avanzado	Siluetas de soldados	5	60.00 %	40.00 %	40.00 %
Figuras	Habilidad	5	40.00 %	80.00 %	80.00 %
Combate	Siluetas con reenes	5	88.00 %	80.00 %	88.00 %
Combate	Larga distancia	10	86.00 %	81.00 %	91.00 %
<b>TOTAL</b>		30	68.00 %	66.60 %	75.40 %

*Figura 46.* Registro de calificación

**Tabla 12**

*Registro de calificaciones en Microsoft Excel*

REGISTRO DE TIRO						
FECHA:		25-nov-18				
UNIDAD:		GEO				
INSTRUCTOR:		CRNL. NARVAEZ F				
ALUMNO		1	2	3	4	
PRACTICA	DESCRIPCION	DISPAROS	Jose	Juan	Mario	-
<b>Precisión</b>	Familiarización	5	66,00%	52,00%	78,00%	-
<b>Avanzado</b>	Siluetas de soldados	5	60,00%	40,00%	40,00%	-
<b>Figuras</b>	Habilidad	5	40,00%	80,00%	80,00%	-
<b>Combate</b>	Siluetas con rehenes	5	88,00%	80,00%	88,00%	-
<b>Combate</b>	Larga distancia	10	86,00%	81,00%	91,00%	-
Total		<b>30</b>	<b>68,00%</b>	<b>66,60%</b>	<b>75,40%</b>	

### 6.3 Consideraciones del uso del software del polígono virtual de tiro

A lo largo del desarrollo del proyecto se han establecido diferentes condiciones las cuales permiten que su funcionamiento sea el adecuado y cumpla con las expectativas planteadas, de tal forma que se debe tomar en cuenta las siguientes consideraciones:

- El hardware empleado en este proyecto (proyector-cámara) debe trabajar de manera conjunta, es decir, la posición de la cámara debe ser lo más cercada posible al lente del proyector lo cual permita la menor deformación de la imagen obtenida por la cámara y el error en la detección del área de disparo y del disparo sea el menor posible.
- Entre los niveles de iluminación, el adecuado siempre será el que se encuentra totalmente aislado de la iluminación externa debido a que esta puede interferir en la imagen de la cámara y generar falsos positivos, creando disparos no existentes.
- Como se ha mencionado anteriormente se ha asignado la posición de disparo a una distancia de cuatro metros desde la superficie de proyección debido a que las siluetas recreadas en los escenarios virtuales se encuentran colocadas en función de esta distancia, y al colocarse en una posición que no sea la señalada se pierde el efecto de inmersión en el entorno virtual.



- En el Capítulo V se menciona la diferencia entre el rastro que deja cada disparador laser sobre la imagen de la cámara, es por esto que, es indispensable realizar la medición del área en pixeles generada por el láser para poder variar los parámetros en el algoritmo de detección del disparo, este proceso permite acoplar diferentes tipos de laser o armas laser con el sistema.

## CAPÍTULO VII

### CONCLUSIONES Y RECOMENDACIONES

#### 7.1 Conclusiones

Se desarrollo satisfactoriamente un sistema de calificación para un polígono virtual de tiro basado en visión por computador, el cual reconoce automáticamente el área de disparo proyectado sobre una superficie vertical y detecta los destellos laser sobre esta zona asignando una calificación en función de la silueta virtual alcanzada por el disparo, el fundamento principal para este proyecto ha sido la detección de puntos de interés.

En la etapa de detección del área de disparo se ha considerado implementar tres algoritmos: SURF, ORB, SIFT y dos combinaciones entre ellos: ORB-SURB y SURF-ORB. Los resultados muestran claramente que SIFT y SURF presentan un error en centímetros menor al resto tanto en el eje X como en el eje Y en todos los escenarios que se han propuesto. ORB realiza la detección del área en el tiempo mínimo con un valor promedio de 60 ms, sin embargo, la precisión en la estimación del área de disparo presenta un alto valor de error. En el caso de la combinación de algoritmos ORB-SURF presenta mejores resultados que ORB en un tiempo promedio de 140 ms, pero su error aun es superior que SURF y SIFT. Por otro lado, SURF-ORB presento valores de error demasiado altos por lo cual no se considera como una solución para este problema. Tomando en cuenta que el sistema prioriza la precisión y no el tiempo de ejecución, SIFT se ha elegido la mejor solución.

La detección del disparo se lo realiza mediante un proceso de detección de blobs en las cuales se aplicaron cuatro tipos de filtrado sobre la imagen: umbralización, detección de circularidad, detección de convexidad y rango del área. Los resultados que arrojo el sistema, cuando fue sometido a diferentes niveles de iluminación externa y diferentes secciones del área de disparo, demuestran que no existe una variación significativa entre los valores de error tanto en el eje X como en el eje Y en los tres escenarios propuestos. Los valores solo en casos aislados superan los 3 cm de error. Considerando los resultados se puede concluir que el escenario más optimo para el funcionamiento del sistema es con bajo nivel de iluminación externa debido a que existe una mayor inferencia de luz produce una mayor cantidad de ruido en el sistema. Lo que se debe considerar es que en el eje Y el error siempre es mayor, este fenómeno proviene desde la detección del área de disparo por lo cual el error final se encuentra determinado por los errores en ambos algoritmos.

Los escenarios que se han propuesto para realizar las pruebas de funcionamiento de los algoritmos implementados son situaciones reales que se presentan dentro del laboratorio de simulación de los hangares del Centro de Investigación Científica y Tecnológica del Ejercito (CICTE) y debido a que este proyecto se encuentra desarrollado en este ambiente, es necesario considerar todos los elementos que pueden interferir con el correcto funcionamiento del sistema para determinar el escenario ideal en el cual se obtenga los mejores resultados.

El algoritmo para la detección de disparo laser fue diseñado para poder acoplarse a diferentes punteros o armas laser que se pretendan usar. Debido al uso de una serie de filtros para la detección del destello, se puede aislarlo y eliminar los ruidos existentes, pero sobre todo al usar como ultimo filtro la detección por área permite que sea ajustable a diferentes rastros generados por las

variedades de laser que existen en el mercado. Dado el caso cuando se desee implementar un nuevo disparador, se debe seguir los pasos establecidos en el Capitulo V para determinar el área estimada por el laser y variar los parámetros del algoritmo.

El proceso de renderizado de los escenarios virtuales aumenta el consumo tanto de la tarjeta gráfica como del procesador interno del computador, adicionalmente el procesamiento de imágenes consume también recursos, lo cual genera una latencia en el sistema mientras más participantes se encuentran realizando una práctica virtual de tiro, es por esto, que es necesario contemplar los recursos necesarios para que el ordenador trabaje adecuadamente y en casos como este en el cual se emplea un motor de videojuegos que entrega una alta resolución, la tarjeta gráfica debe seleccionarse bajo las especificaciones del software de desarrollo.

## **7.2 Recomendaciones**

Es necesario tomar en cuenta todas las consideraciones que se han establecido a lo largo del desarrollo del proyecto, principalmente en lo que tiene que ver con la colocación adecuada del hardware (sistema cámara-proyector), la adecuación del entorno controlado al nivel de iluminación recomendado, que es totalmente aislado de luz, además la distancia a la cual el usuario debe colocarse para realizar el disparo y sobre todo se debe tomar en cuenta el tipo de laser que se usa, ya que se debe realizar una configuración del parámetro del área del mismo para que el sistema pueda reconocerlo adecuadamente.

Se recomienda realizar un estudio completo previo de software necesario para determinar los equipos necesarios, ya que se deben considerar varios aspectos como es los protocolos de

comunicación, memoria RAM necesaria, GPU necesaria para el procesamiento de la imagen. Caso contrario se pueden generar inconvenientes en el proceso de integración, tal es el caso de latencia cuando se ejecuta Unity con una tarjeta grafica de mínimas capacidades, o perdida de comunicación cuando se trata de comunicar Unity con una cámara de tipo IP.

Es recomendable revisar la documentación adecuada e involucrarse completamente con el tema en desarrollo, en este caso ha sido fundamente realizar reconocimientos de escenarios reales de tiro y los protocolos que se siguen en las prácticas de tiro, además es necesario comprender claramente cuál es el problema para determinar la solución óptima.

## REFERENCIAS BIBLIOGRÁFICAS

- Aguilar, W. G., & Angulo, C. (2012). Compensación de los efectos generados en la imagen por el control de navegación del robot Aibo ERS 7. *VII Congreso de Ciencia y Tecnología ESPE*. Sangolquí, Ecuador.
- Aguilar, W. G., & Angulo, C. (2012). Compensación y aprendizaje de efectos generados en la imagen durante el desplazamiento de un robot. *X Simposio CEA de Ingeniería de Control*. Barcelona, Spain.
- Aguilar, W. G., & Angulo, C. (2013). Estabilización robusta de vídeo basada en diferencia de nivel de gris. *VIII Congreso de Ciencia y Tecnología ESPE*. Sangolquí, Ecuador.
- Aguilar, W. G., & Angulo, C. (2014). Estabilización de vídeo en micro vehículos aéreos y su aplicación en la detección de caras. *IX Congreso de Ciencia y Tecnología ESPE*. Sangolquí, Ecuador.
- Aguilar, W. G., & Angulo, C. (2014). Real-time video stabilization without phantom movements for micro aerial vehicles. *EURASIP Journal on Image and Video Processing*, 1, 1-13.
- Aguilar, W. G., & Angulo, C. (2014). Robust video stabilization based on motion intention for low-cost micro aerial vehicles. *11th International Multi-Conference on Systems, Signals & Devices (SSD)*. Barcelona, Spain.

Aguilar, W. G., & Angulo, C. (2016). Real-Time Model-Based Video Stabilization for Microaerial Vehicles. *Neural Processing Letters*, 43(2), 459-477.

Aguilar, W. G., & Morales, S. (2016). 3D Environment Mapping Using the Kinect V2 and Path Planning Based on RRT Algorithms. *Electronics*, 5(4), 70.

Aguilar, W. G., Abad, V., Ruiz, H., Aguilar, J., & Aguilar-Castillo, F. (2017). RRT-Based Path Planning for Virtual Bronchoscopy Simulator. En *Lecture Notes in Computer Science* (págs. 155-165).

Aguilar, W. G., Angulo, C., & Costa-Castello, R. (2017). Autonomous Navigation Control for Quadrotors in Trajectories Tracking. En *Lecture Notes in Computer Science* (págs. 287-297).

Aguilar, W. G., Angulo, C., Costa, R., & Molina, L. (2014). Control autónomo de cuadricópteros para seguimiento de trayectorias. *IX Congreso de Ciencia y Tecnología ESPE*. Sangolquí, Ecuador.

Aguilar, W. G., Casaliglla, V. P., & Pólit, J. L. (2017). Obstacle Avoidance Based-Visual Navigation for Micro Aerial Vehicles. *Electronics*, 6(1), 10.

Aguilar, W. G., Casaliglla, V. P., & Pólit, J. L. (2017). Obstacle Avoidance for Low-Cost UAVs. *IEEE 11th International Conference on Semantic Computing (ICSC)*. San Diego.

Aguilar, W. G., Casaliglla, V. P., Pólit, J. L., Abad, V., & Ruiz, H. (2017). Obstacle Avoidance for Flight Safety on Unmanned Aerial Vehicles. En *Lecture Notes in Computer Science* (págs. 575-584).

Aguilar, W. G., Luna, M. A., Moya, J. F., Abad, V., Ruiz, H., Parra, H., & Angulo, C. (2017). Pedestrian Detection for UAVs Using Cascade Classifiers and Saliency Maps. En *Lecture Notes in Computer Science* (págs. 563-574).

Aguilar, W. G., Luna, M. A., Moya, J. F., Abad, V., Ruiz, H., Parra, H., & Lopez, W. (2017). Cascade Classifiers and Saliency Maps Based People Detection. En *Lecture Notes in Computer Science* (págs. 501-510).

Aguilar, W. G., Luna, M. A., Moya, J. F., Luna, M. P., Abad, V., Ruiz, H., & Parra, H. (2017). Real-Time Detection and Simulation of Abnormal Crowd Behavior. En *Lecture Notes in Computer Science* (págs. 420-428).

Aguilar, W. G., Luna, M. A., Ruiz, H., Moya, J. F., Luna, M. P., Abad, V., & Parra, H. (2017). Statistical Abnormal Crowd Behavior Detection and Simulation for Real-Time Applications. En *Lecture Notes in Computer Science* (págs. 671-682).

Aguilar, W. G., Luna, M., Moya, J., Abad, V., Parra, H., & Ruiz, H. (2017). Pedestrian Detection for UAVs Using Cascade Classifiers with Meanshift. *IEEE 11th International Conference on Semantic Computing (ICSC)*. San Diego.



Aguilar, W. G., Morales, S., Ruiz, H., & Abad, V. (2017). RRT\* GL Based Optimal Path Planning for Real-Time Navigation of UAVs. En *Lecture Notes in Computer Science* (págs. 585-595).

Aguilar, W. G., Morales, S., Ruiz, H., & Abad, V. (2017). RRT\* GL Based Path Planning for Virtual Aerial Navigation. En *Lecture Notes in Computer Science* (págs. 176-184).

Aguilar, W. G., Rodríguez, G. A., Álvarez, L., Sandoval, S., Quisaguano, F., & Limaico, A. (2017). On-Board Visual SLAM on a UGV Using a RGB-D Camera. En *Lecture Notes in Computer Science* (págs. 298-308).

Aguilar, W. G., Rodríguez, G. A., Álvarez, L., Sandoval, S., Quisaguano, F., & Limaico, A. (2017). Real-Time 3D Modeling with a RGB-D Camera and On-Board Processing. En *Lecture Notes in Computer Science* (págs. 410-419).

Aguilar, W. G., Rodríguez, G. A., Álvarez, L., Sandoval, S., Quisaguano, F., & Limaico, A. (2017). Visual SLAM with a RGB-D Camera on a Quadrotor UAV Using on-Board Processing. En *Lecture Notes in Computer Science* (págs. 596-606).

Aguilar, W. G., Salcedo, V. S., Sandoval, D. S., & Cobeña, B. (2017). Developing of a Video-Based Model for UAV Autonomous Navigation. En *Communications in Computer and Information Science* (págs. 94-105).

Amaguaña, F., Collaguazo, B., Tituaña, J., & Aguilar, W. G. (2018). Simulation System Based on Augmented Reality for Optimization of Training Tactics on Military Operations.

*International Conference on Augmented Reality, Virtual Reality and Computer Graphics* (págs. 394-403). Springer.

Andrea, C. C., Byron, J. Q., Jorge, P. I., Inti, T. C., & Aguilar, W. G. (2018). Geolocation and Counting of People with Aerial Thermal Imaging for Rescue Purposes. *International Conference on Augmented Reality, Virtual Reality and Computer Graphics* (págs. 171-182). Springer.

Awad, A. I., & Hassaballah, M. (s.f.). *Image Feature Detectors and Descriptors Foundations and Applications* (Vol. 630).

Basantes, J., Godoy, L., Carvajal, T., Castro, R., Toulkeridis, T., Fuentes, W., . . . Addison, A. (2018). Capture and processing of geospatial data with laser scanner system for 3D modeling and virtual reality of Amazonian Caves. *IEEE Ecuador Technical Chapters Meeting (ETCM)*. Samborondón, Ecuador.

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). *SURF: Speeded Up Robust Features*. Obtenido de Computer Vision – ECCV 2006 : doi: 10.1007/11744023\_32

Blotta, E., & Ballarin, V. (2012). *Enhancement of medical images in HSI color space*. Obtenido de Journal of Physics: Conference Series: doi:10.1088/1742-6596/332/1/012041

Cabras, P., Rosell, J., Pérez, A., Aguilar, W. G., & Rosell, A. (2011). Haptic-based navigation for the virtual bronchoscopy. *18th IFAC World Congress*. Milano, Italy.

- Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). *BRIEF: Binary Robust Independent Elementary Features*. Obtenido de 11th European Conference on Computer Vision (ECCV): doi: 10.1007/978-3-642-15561-1\_56
- CDC. (2013). *Reducing Exposure to Lead and Noise at Outdoor Firing Ranges*. Obtenido de NIOSH Publications and Products: doi: 10.26616/NIOSH PUB2013104
- Domínguez, R. (2016). *Desarrollo de simulador de armas y su integración en una cueva de realidad virtual-CAVE*.
- Double Action Indoor Shooting Center. (2018). <https://doubleaction.com/services/shooting-range/>.
- Epson Gallery. (2018). Obtenido de <https://www.epson.co.in/Projectors/Corporate-and-Education/Epson-X04-XGA-3LCD-Projector/p/V11H717052>
- Etienne, V., & Laganier, R. (2001). *Detecting Planar Homographies in an Image Pair*. Obtenido de Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis: doi: 10.1109/ISPA.2001.938625
- General Services Administration. (2011). *GSA Guidance Document - Indoor Firing Ranges - Design and Operations Criteria*.
- Harris, C., & Stephens, M. (1998). *A combined corner and edge detector*. Obtenido de Proceedings of the 4th Alvey Vision Conference.

- Jara-Olmedo, A., Medina-Pazmiño, W., Mesías, R., Araujo-Villaroel, B., Aguilar, W. G., & Pardo, J. A. (2018). Interface of Optimal Electro-Optical/Infrared for Unmanned Aerial Vehicles. *En Smart Innovation, Systems and Technologies* (págs. 372-380).
- Jara-Olmedo, A., Medina-Pazmiño, W., Tozer, T., Aguilar, W. G., & Pardo, J. A. (2018). E-services from Emergency Communication Network: Aerial Platform Evaluation. *International Conference on eDemocracy & eGovernment (ICEDEG)* (págs. 251-256). IEEE.
- Karami, E., Prasad, S., & Shehata, M. (2017). *Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images*. Obtenido de arXiv:1710.02726.
- Kardous, C. A., & Murphy, W. J. (2010). *Noise control solutions for indoor firing range*. Obtenido de Noise Control Engineering Journal: doi: 10.3397/1.3455050
- Kaur, S., & Vijay, K. (2013). *Content based image retrieval: Survey and comparison between rgb and hsv model*. Obtenido de International Journal of Engineering Trends and Technology.
- Kumar, T., & Verma, K. (2010). *A Theory Based on Conversion of RGB image to Gray image*. Obtenido de International Journal of Computer Applications: doi: 10.5120/1140-1493
- Lapointe, J.-F., & Godin, G. (2005). *On-Screen Laser Spot Detection for Large Display Interaction*. Obtenido de IEEE International Workshop: doi: 10.1109/HAVE.2005.1545655

- Las Vegas Outdoor Shooting Range. (2018). <https://vegasoutdooradventures.com/>.
- Leung, H., Chan, J. C., Tang, J., & Komura, T. (2011). *A Virtual Reality Dance Training System Using Motion Capture Technology*. Obtenido de IEEE Transactions on Learning Technologies: DOI: 10.1109/TLT.2010.27
- Li, X., & Plataniotis, K. (2015). *Color model comparative analysis for breast cancer diagnosis using H and E stained images*. Obtenido de In Medical Imaging 2015: Digital Pathology (Vol. 9420, p. 94200L):. <https://doi.org/10.1117/12.2079935>
- Li, Y., Wang, S., Tian, Q., & Ding, X. (2015). *A survey of recent advances in visual feature detection*. Obtenido de Neurocomputing: doi:10.1016/j.neucom.2014.08.003
- Lindeberg, T. (1993). *Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention*. Obtenido de International Journal of Computer Vision: doi:10.1007/BF01469346
- Lindeberg, T. (1998). *Feature Detection with Automatic Scale Selection*. Obtenido de International journal of computer vision: doi:10.1023/A:1008045108935
- Logitech Gallery. (2018). Obtenido de <https://www.logitech.com/es-es/product/hd-pro-webcam-c920>
- Lowe, D. G. (2004). *Distinctive image features from scale-invariant keypoints*. Obtenido de International journal of computer vision: doi: 10.1023/B:VISI.0000029664.99615.94

Ma, Y., Soatto, S., Košecká, J., & Sastry, S. (2003). *An Invitation to 3-D Vision: From Images to Models*. Obtenido de SpringerVerlag: <https://doi.org/10.1007/978-0-387-21779-6>

MEŠKO, M., & TOTH, Š. (2013). *Laser spot detection*.

MiloRange. (2018). Obtenido de <https://www.faac.com/milo-range/simulators/>

OpenCV. (2018). [www.opencv.org](http://www.opencv.org).

OpenCV Docs. (2018). [https://docs.opencv.org/3.4.1/d9/dab/tutorial\\_homography.html](https://docs.opencv.org/3.4.1/d9/dab/tutorial_homography.html).

Orbea, D., Moposita, J., Aguilar, W. G., Paredes, M., León, G., & Jara-Olmedo, A. (2017). Math Model of UAV Multi Rotor Prototype with Fixed Wing Aerodynamic Structure for a Flight Simulator. En *Lecture Notes in Computer Science* (págs. 199-211).

Orbea, D., Moposita, J., Aguilar, W. G., Paredes, M., Reyes, R. P., & Montoya, L. (2017). Vertical take off and landing with fixed rotor. *Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. Pucón, Chile.

Pardo, J. A., Aguilar, W. G., & Toulkeridis, T. (2017). Wireless communication system for the transmission of thermal images from a UAV. *Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. Pucón, Chile.

Pratt, W. K. (2007). *Digital Image Processing* (4 a Edition. Wiley-Interscience ed.).

Research Gate. (2018). *Diagram of SIFT keypoint detection algorithm showing one octave with 6 Gaussian image layers*. Obtenido de [https://www.researchgate.net/figure/Diagram-of-SIFT-keypoint-detection-algorithm-showing-one-octave-with-6-Gaussian-image\\_fig1\\_256546531](https://www.researchgate.net/figure/Diagram-of-SIFT-keypoint-detection-algorithm-showing-one-octave-with-6-Gaussian-image_fig1_256546531).

ResearchGate. (2018). <http://www.researchgate.com>. Obtenido de RGB model representation: [https://www.researchgate.net/figure/RGB-model-representation\\_fig1\\_239814963](https://www.researchgate.net/figure/RGB-model-representation_fig1_239814963)

ResearchGate. (2018). <http://www.researchgate.com>. Obtenido de HSV model representation: [https://www.researchgate.net/figure/The-HSV-color-model\\_fig1\\_243770240](https://www.researchgate.net/figure/The-HSV-color-model_fig1_243770240)

ResearchGate. (2018). <http://www.researchgate.com>. Obtenido de HSI model representation: [https://www.researchgate.net/figure/HSI-model-representation-The-letters-R-Y-G-C-B-M-refer-to-Red-Yellow-Green\\_fig2\\_239814963](https://www.researchgate.net/figure/HSI-model-representation-The-letters-R-Y-G-C-B-M-refer-to-Red-Yellow-Green_fig2_239814963)

Rohit, J. (2006). *Projector Camera Applications: Shooting Range Simulation*.

Rosten, E., & Drummond, T. (2006). *Machine learning for high-speed corner detection*. Obtenido de Computer Vision – ECCV 2006: doi: 10.1007/11744023\_34

Rosten, E., & Drummond, T. (2006). *Machine learning for high-speed corner detection*. Obtenido de European conference on computer vision: doi: [https://doi.org/10.1007/11744023\\_34](https://doi.org/10.1007/11744023_34)

Rubble, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). *ORB: An efficient alternative to SIFT or SURF*. Obtenido de 2011 International Conference on Computer Vision: doi: 10.1109/ICCV.2011.6126544

Ryt, A., & Sobel, D. (2014). *Real-Time Laser Point Tracking*. Obtenido de Computer Vision and Graphics .

Soliman, A. (2016). *Vision-Aided Navigation for Autonomous Vehicles Using Tracked Feature Points*.

Sturm, P. (2016). *Pinhole Camera Model*. Obtenido de Computer Vision: doi:10.1007/978-0-387-31439-6\_472

Unity. (2018). *www.unity3d.com*.

Unreal Engine. (2018). *www.unrealengine.com*.

Virtra homepage. (2018). Obtenido de <https://www.virtra.com/overview-le>

Xu, D. (2008). *A Neural Network Approach for Hand Gesture Recognition in Virtual Reality Driving Training System of SPG*. Obtenido de 18th International Conference on Pattern Recognition: DOI: 10.1109/ICPR.2006.109



Zhai, Y., Yu, G., Wang, H., & Guo, X. (2017). *Image matching for structured scenes based on ASIFT and homography constraint*. Obtenido de 2017 3rd IEEE International Conference on Computer and Communications (ICCC): doi: 10.1109/CompComm.2017.8322914

Zhang, M., Wu, T., & Beeman, S. C. (2016). *Efficient Small Blob Detection Based on Local Convexity, Intensity and Shape Information*. Obtenido de IEEE Transactions on Medical Imaging: doi: 10.1109/TMI.2015.2509463

Zitova, B., & Flusser, J. (2003). *Image registration methods: a survey*. Obtenido de Image and Vision Computing: doi: 10.1016/S0262-8856(03)00137-9