

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA POLITECNICA SUPERIOR DE GANDIA

Grado en Ing. Sist. de Telecom., Sonido e Imagen



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCOLA POLITÈCNICA
SUPERIOR DE GANDIA

“Sistema interactivo de guiado en boulders aplicando técnicas de visión artificial y realidad aumentada”

TRABAJO FINAL DE GRADO

Autor/a:

Pedro José Laguía Roldán

Tutor/a:

Fernando Boronat Seguí

Clara Luzon Alvarez

GANDIA, 2018

Resumen

El propósito principal del presente proyecto es, fundamentalmente, el de crear un sistema interactivo que permita a cualquier usuario realizar sencillos ejercicios de escalada en un rocódromo, de manera guiada a través de un proyector y utilizando técnicas de visión artificial y de realidad aumentada. El usuario podrá seleccionar diferentes tipos de rutas con diferentes grados de dificultad desde una aplicación de usuario. Dicha aplicación también incluirá juegos multijugador (como, por ejemplo, un juego tipo pingpong entre dos jugadores en el *Boulder*) o juegos contrareloj. El sistema incluirá una aplicación de gestión que permita crear las diferentes rutas y clasificarlas por niveles de dificultad.

El sistema incluirá un PC, una cámara que capte imágenes del usuario en tiempo real, y, por último, un sistema de proyección de imágenes en el rocódromo formado por un videoprojector.

Abstract

The main objective in this project is, fundamentally, to create an interactive system that allows a user to perform climbing exercises in a climbing wall (a.k.a. boulder), guided by a projector and using artificial vision and augmented reality techniques. The user can select different types of routes with different degrees of difficulty from an end-user application. This application will also include multiplayer games (such as, for example, a Ping-Pong game between two players in the boulder) or time trial games. The system will include an additional management application that allows creating the different routes and classified by levels of difficulty.

The system includes a computer, a camera that will capture images of the user in real time, and, finally, a system for projecting images in the climbing wall formed by one video projector.

Palabras Clave

Rocódromo, Tratamiento Digital de Imagen, Visión Artificial, Realidad aumentada, Interactivo.

Key words

Climbing Wall, Digital Image Processing, Artificial Vision, Augmented Reality, Interactive.

Contenido

Resumen	
Abstract	
Palabras Clave	
Key words	
1. Introducción	4
2. Estado del arte.....	5
3. Herramientas	6
3.1. Software	6
3.2. Hardware.....	7
3.3. Arquitectura de la aplicación.....	8
4. Desarrollo	9
4.1. Captación de la superficie y procesado	9
4.1.1. Máscaras de color y operadores morfológicos	14
4.1.2. Implantación de etiquetas amarillas y detección de centros	18
4.2. Desarrollo de la aplicación	20
4.3. Proyección y análisis de los resultados	23
4.3.1. Proceso de auto-calibración.....	23
4.3.2. Resultados.....	27
4.4. Interfaz de la aplicación	32
5. Conclusiones	34
6. Bibliografía.....	35
7. Anexos	36

Lista de figuras

Figura 1. Ordenador Lenovo z500	7
Figura 2. Proyector Mitsubishi XD460U	7
Figura 3. Arquitectura de la aplicación	8
Figura 4. Primera toma del rocódromo.....	9
Figura 5. Cálculo del gradiente	10
Figura 6. Representación gráfica de modulo y fase del gradiente	10
Figura 7. Método "Hysteresis thresholding"	11
Figura 8. Detector de bordes aplicando el método de Canny Edge.....	12
Figura 9. Detección de bordes por Canny con filtrado bilateral	12
Figura 10. Detección de bordes con fondo uniforme	13
Figura 11. Toma mejorada del rocódromo	14
Figura 12. Representación del modelo HSV	14
Figura 13. Imagen binaria correspondiente a la mascara.....	15
Figura 14. Operación morfológica de erosión	15
Figura 15. Operación morfológica de dilatación	16
Figura 16. Máscara final tras aplicar la operación opening	16
Figura 17. Detector de bordes de Canny a la máscara final.....	17
Figura 18. Detección de contornos de las presas	17
Figura 19. Tercera toma del rocódromo con etiquetas amarillas.....	18
Figura 20. Centros de las presas del rocódromo.....	19
Figura 21. Ejemplo de ruta personalizada.....	20
Figura 22. Ejemplo de prueba de Splatoon	21
Figura 23. Símbolo del ganador del juego.....	21
Figura 24. Pingpong 2D	22
Figura 25. Requisito de posicionamiento entre cámara y proyector	24
Figura 26. Captación de la imagen de referencia.....	25
Figura 27. Puntos de las esquinas de la imagen de referencia	26
Figura 28. Proceso de homografía y cambio de perspectiva	27
Figura 29. Montaje del escenario.....	28
Figura 30. Resultado de "Juego 1: Elige tu ruta"	29
Figura 31. Resultado de "Juego 2: Splatoon"	30
Figura 32. Demostración de "Juego 3: Multijugador Pingpong".....	31
Figura 33. Menú principal de la aplicación	32
Figura 34. Instrucciones de juego	32
Figura 35. Menú de juegos	33
Figura 36. Menú de selección de ruta.....	33

1. Introducció

Desde los orígenes, se ha practicado una disciplina deportiva conocida con el nombre de montañismo y que consiste en la realización de ascensos y descensos por montaña, así como el conjunto de habilidades y conocimientos para realizar dicho objetivo. Posteriormente, allá en el siglo XIX surge una actividad derivada denominada escalada, actividad que se basa en la realización de ascensos en montañas de gran pendiente y que utiliza como herramientas tanto la fuerza física como la mental.

Hoy en día existen muchos tipos de escaladas, unas con mas riesgos que otras, pero en este proyecto haremos referencia al concepto de escalada deportiva en rocódromos. Este tipo de escalada surge como una solución para practicar la escalada sin necesidad de trasladarnos a la montaña. El rocódromo se compone presas artificiales simulando a las rocas naturales, hechas de resinas sintéticas con una buena adherencia que presentan diferentes tamaños y colores y que son atornilladas a la plancha o estructura del rocódromo con tornillos de *Allen* con lo que podemos variar su posición si así se quisiera en un determinado momento.

Con el desarrollo de las nuevas tecnologías, un equipo finlandés del departamento de Media Technology de la Universidad de Aalto dirigido por los investigadores *Raine Kajastila* y *Perttu Hämäläinen*, han ideado un sistema para realizar la escalada en rocódromos utilizando la realidad aumentada, es decir, añadiendo información virtual a la información física ya existente a través de un dispositivo tecnológico como puede ser un ordenador o un teléfono móvil, con la finalidad de combinar elementos virtuales como por ejemplo la proyección de rutas, con elementos físicos que en nuestro caso es el rocódromo, y crear así el concepto de una realidad aumentada en tiempo real.

Actualmente existe una gran limitación en las aplicaciones existentes de realidad aumentada dedicadas al mundo de la escalada. Estas aplicaciones se componen de un sistema de proyección, un sistema de captación y una superficie de proyección que será el rocódromo o *boulder*, por lo que el bajo coste del material y la sencillez de la instalación será un aliciente para que cualquier usuario pueda utilizarlo para crear sus propios entrenamientos desde una aplicación a nivel de usuario sin necesidad de ser experto en la materia.

2. Estado del arte

Existen varias compañías que han desarrollado este sistema, la más destacada es ValoMotion, una empresa finlandesa dedicada a desarrollar juegos, así como la fabricación de hardware de alta calidad. El primer proyecto que desarrollaron con el nombre de *ValoClimb* corresponde al desarrollo de juegos en realidad aumentada sobre un muro de escalada y que han implantado ya en diferentes centros comerciales, museos o salas fitness en mas de 40 países.

El problema surge cuando este sistema se intenta adaptar a cualquier superficie de escalada, ya que esta empresa desarrolla una solución patentada con su propio muro de escalada con unas condiciones determinadas, así como también su propio sistema de captación y proyección.

La solución que se intenta ofrecer en este proyecto es la de desarrollar un sistema que permita realizar la escalada aplicando la realidad aumentada en cualquier tipo de rocódromo además de la posibilidad de utilizar un hardware de captación, procesado y proyección asequible económicamente para cualquier usuario, el cual no ha de tener unos conocimientos demasiado técnicos para la instalación de este sistema.

Por último, cabe destacar que este proyecto ya ha sido desarrollado anteriormente por otro alumno de la UPV (*Pablo Aguilar: "Realidad aumentada aplicada a la escalada"*), pero con la diferencia de que en este proyecto vamos a abordar una superficie de escalada muchísimo más compleja, siendo esta una superficie rocosa y con muchos relieves además de presentar una desigualdad de tonos oscuros en la misma, que nos va a llevar a realizar distintos experimentos para conseguir una efectividad de detección de presas del cien por cien. Además, las condiciones de iluminación han sido bastante tediosas y nos han complicado el proceso de detección, presentado un ventanal de metro y medio de alto justo al lado de la superficie, por lo que los tonos del rocódromo eran variantes dependiendo de la hora del día.

Por otro lado el desarrollo de los juegos será distinto, coincidiendo solamente en el de la creación de rutas ya que este juego es el que da esencia y fundamenta la escalada, por lo tanto era obvio implantarlo, aunque con alguna innovación como la selección de distinto grosor de línea de trazado para dar prioridad a un camino u a otro.

Como conclusión, cabe comentar que el objetivo principal de este proyecto es desarrollar y demostrar todos los juegos que hemos ideado, tratando de programar un procesado lo más rápido a la vez que eficaz con el fin de aumentar la sensación de realidad aumentada percibida por el usuario.

3. Herramientas

En el presente capítulo se detallarán, por un lado, el software o el conjunto de programas informáticos que hacen posible la realización de unas tareas específicas, y, por otro lado, el hardware o parte física del sistema informático que es responsable de ejecutar las ordenes que recibe del software y que por lo tanto hace posible el correcto funcionamiento de dicho sistema.

3.1. Software

El sistema operativo utilizado para llevar a cabo todo el desarrollo del proyecto ha sido *Windows 10*. Por otro lado, se ha de comentar que el entorno de desarrollo integrado empleado para realizar la implementación del código de la aplicación ha sido *IDLE* el cual es multiplataforma y además está diseñado exclusivamente para construir aplicaciones con el lenguaje Python, que es el lenguaje que ha sido utilizado para la implementación de la aplicación ya que tiene una sintaxis bastante sencilla y además la importación de módulos externos no supone un coste computacional elevado. Por último, se han de citar la utilización de cada una de las librerías externas que nos han facilitado la implementación funcional de la aplicación, éstas son:

- *OpenCV*: Es la librería que fundamenta la mayor parte de este proyecto. Ésta es libre y se caracteriza por ser multiplataforma de visión artificial y desarrollada por Intel, que contiene más de quinientas funciones y que es utilizada en aplicaciones de detección de movimiento o de reconocimiento de objetos, entre otras.
- *NumPy*: es una extensión de Python, la cual ofrece una mayor consistencia a los vectores y matrices, constituyendo de este modo una biblioteca de funciones matemáticas para operar con dichos vectores o matrices.
- *Matplotlib*: Es una biblioteca dedicada a la generación de gráficos a partir de los datos que se encuentran en listas o arrays en el lenguaje de programación Python y su extensión matemática *NumPy*.
- *Pygame*: Es un conjunto de módulos del lenguaje Python que permiten la creación de videojuegos en dos dimensiones de una manera sencilla.

3.2. Hardware

Por otro lado, la parte física seleccionada para llevar a cabo el funcionamiento de la aplicación ha sido bastante sencilla a la vez que efectiva, con el fin efectuar una instalación del escenario accesible a cualquier tipo de usuario. Los materiales utilizados para cumplir el objetivo previamente citado han sido los siguientes:

- Ordenador portátil: Se ha utilizado el modelo Lenovo z500 IdeaPad que es un computador de gama media alta, con un procesador Intel Core i7 y con una tarjeta gráfica NVIDIA GeForce GT 740M, es decir, suficiente para procesar cómodamente el coste computacional que implica todo el tratamiento digital de la imagen a tiempo real.
- Cámara Web: Herramienta con la que vamos a realizar la captación de la escena para posteriormente procesarla, con una resolución más que suficiente (1280x720) para enviar la información a tiempo real y hacer posible el correcto funcionamiento de la aplicación. Ésta va integrada en el propio ordenador portátil y hemos evitado utilizar una externa conectada por USB con la finalidad de evitarnos el uso de cables y hacer más sencilla la instalación.



Figura 1. Ordenador Lenovo z500

- Proyector: Se ha empleado el modelo Mitsubishi XD460U con una potencia luminosa de 2600 lúmenes, algo escasa para las complejas condiciones de luminosidad que se nos han presentado en la sala polivalente donde se halla el rocódromo debido al ventanal anteriormente citado, pero que nos ha permitido proyectar unas condiciones de luminosidad nocturnas para poder demostrar el funcionamiento de la aplicación.



Figura 2. Proyector Mitsubishi XD460U

3.3. Arquitectura de la aplicaci3n

Teniendo en cuenta tanto el software como el hardware que vamos a utilizar ya explicado en los capítulos anteriores, es necesario saber el orden de actuaci3n que van a seguir dentro de la estructura de la cual se compone la aplicaci3n que vamos a desarrollar en este proyecto.

En primer lugar, entra en juego la superficie del roc3dromo que ser3 captada por la c3mara la cual estar3 integrada dentro del ordenador, con lo que interesa que la posici3n de esta, as3 como las condiciones de iluminaci3n de la sala polivalente en la que se all3 el roc3dromo sean lo 3ptimas posibles con el fin de facilitar el segundo paso funcional de la aplicaci3n, el procesado.

En segundo lugar, tenemos que hablar del procesado de la imagen que captamos a trav3s de la c3mara. En este procesado vamos a aplicar diferentes t3cnicas de tratamiento digital de imagen tanto para mejorar la imagen (contraste, saturaci3n...) como para realizar la detecci3n de las presas (filtrados, detecci3n de bordes...) dentro de la superficie con el objetivo de poder proyectar dichos puntos y que el usuario pueda interactuar con ellos a tiempo real creando as3 una sensaci3n de realidad aumentada.

Por 3ltimo, cabe destacar que el ultimo paso y no menos importante tiene que ver con la proyecci3n que se realiza tras aplicar el procesado de la imagen. Se proyectar3n cada uno de los puntos detectados correspondientes a las presas en la superficie en forma de c3rculo blanco, as3 como las diferentes rutas o el cambio de color de estos c3rculos si el usuario entrara en contacto con ellos, por ejemplo.

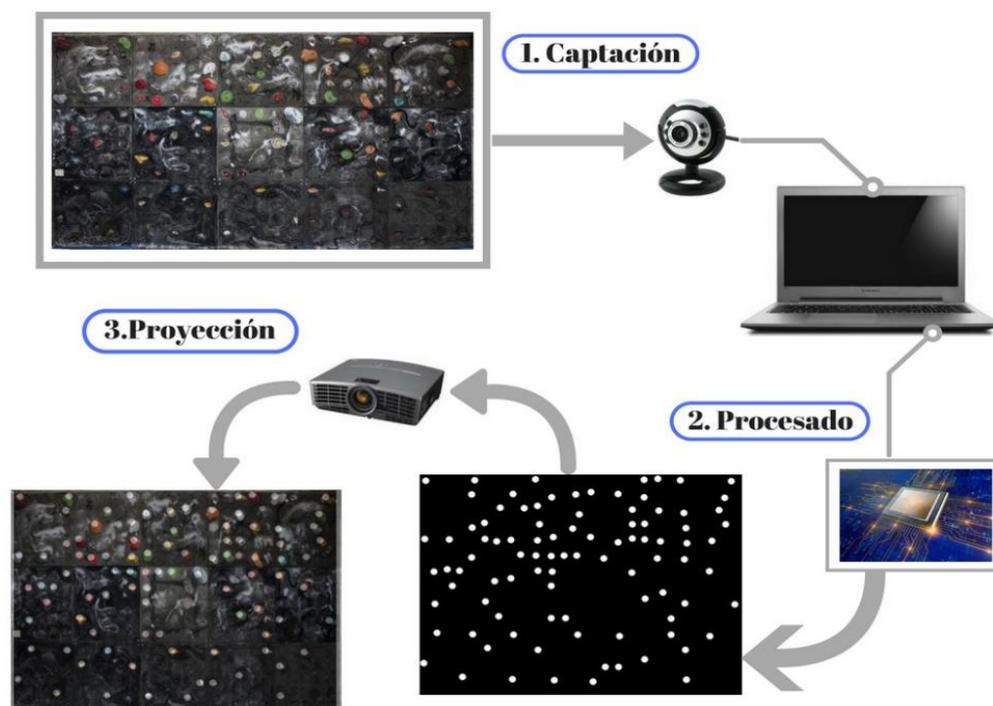


Figura 3. Arquitectura de la aplicaci3n

Como conclusión, cabe señalar que este diseño tiene un funcionamiento cíclico y a tiempo real, es decir, capturamos al usuario en ese momento, actualizamos la información recibida por la cámara y por último proyectamos con la información procesada en ese instante con el fin de crear la sensación de realidad aumentada.

4. Desarrollo

4.1. Captación de la superficie y procesado

La primera toma de contacto con el rocódromo del Campus de Gandía se realizó tomando unas fotografías de la superficie de escalada para su posterior análisis.



Figura 4. Primera toma del rocódromo

En primer lugar, se creó un código utilizando las funciones de tratamiento digital de la imagen de la librería OpenCV previamente citada, con el fin de detectar las presas del rocódromo, lo cual será nuestro objetivo fundamental, ya que el saber dónde están situadas todas ellas dentro de la imagen que captamos será la base para desarrollar los distintos tipos de juegos en el bouldering.

Dado que se tenían previos conocimientos de tratamiento digital de la imagen, asignatura cursada previamente en el primer cuatrimestre del Grado en Ingeniería de Sistemas de Telecomunicación, Sonido e Imagen, se daba por hecho que el método para cumplir este objetivo era utilizar el algoritmo de Canny con el que podemos detectar los bordes de una imagen, en nuestro caso, el borde de las presas, ya que hoy en día es el más utilizado, éste método define tres objetivos fundamentales:

- Baja tasa de error: Se deben encontrar todos los bordes sin respuestas falsas.
- Buena localización: Los bordes han de estar los mas próximos posible a los bordes reales.
- Respuesta única: Se debe obtener una única respuesta por cada punto perteneciente al borde verdadero.

Este detector hace uso varios algoritmos empleando la información de fase del gradiente, que un vector cuya magnitud es la máxima tasa de cambio de luminosidad y apunta hacia la dirección en la que se produce dicho cambio, el gradiente es perpendicular al borde y su amplitud es proporcional a la magnitud del cambio de luminosidad para proporcionar mejores resultados. El cálculo del gradiente se puede observar de forma más aclaratoria en la siguiente ilustración:

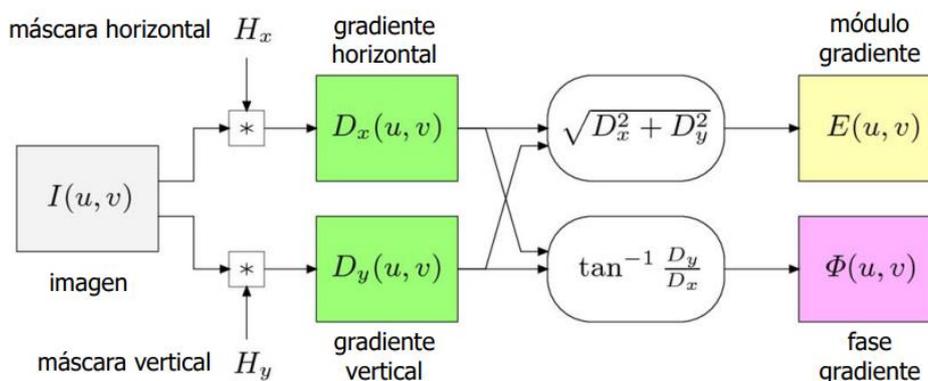


Figura 5. Cálculo del gradiente

A continuación, se muestra un ejemplo del resultado que surge de la operación anterior:

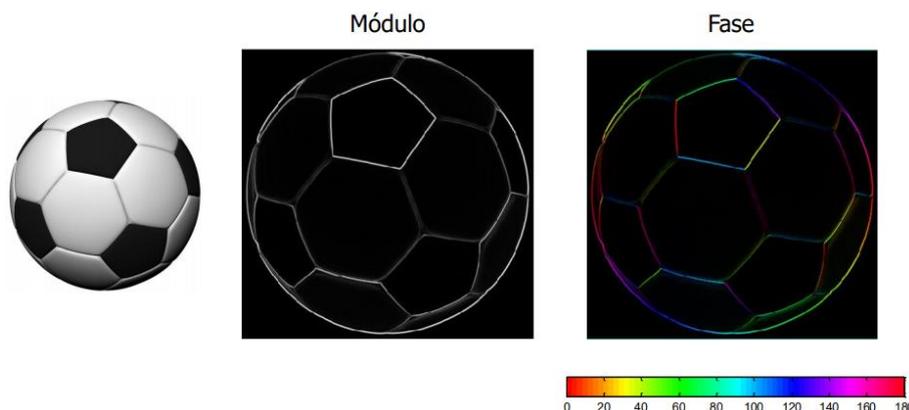


Figura 6. Representación gráfica de modulo y fase del gradiente

Teniendo claro el concepto de gradiente, el detector de Canny se puede resumir en los siguientes pasos:

1. Reducción de ruido en la imagen: Se aplica un filtro de suavizado gaussiano, el cual emborrona o suaviza la imagen con el fin de eliminar el posible ruido que pueda existir en ella.
2. Cálculo del módulo y la fase del gradiente.
3. Se aplica el operador "*Non maximal supression*" para adelgazar los bordes al máximo, éste emplea la dirección del módulo del gradiente.
4. Se aplica un seguimiento de borde estableciéndose un doble umbral para unir tramos no conectados, denominado como "*Hysteresis thresholding*", el cual realiza lo siguiente:
 - a. Se definen dos umbrales: el alto y el bajo
 - b. Se umbraliza el gradiente con el umbral más alto
 - c. Se añaden los pixeles vecinos que superen el umbral bajo
 - d. El umbral alto sirve para empezar los bordes y el bajo para continuarlos.

En la siguiente figura se muestra un gráfico para representar este proceso de manera más clara:

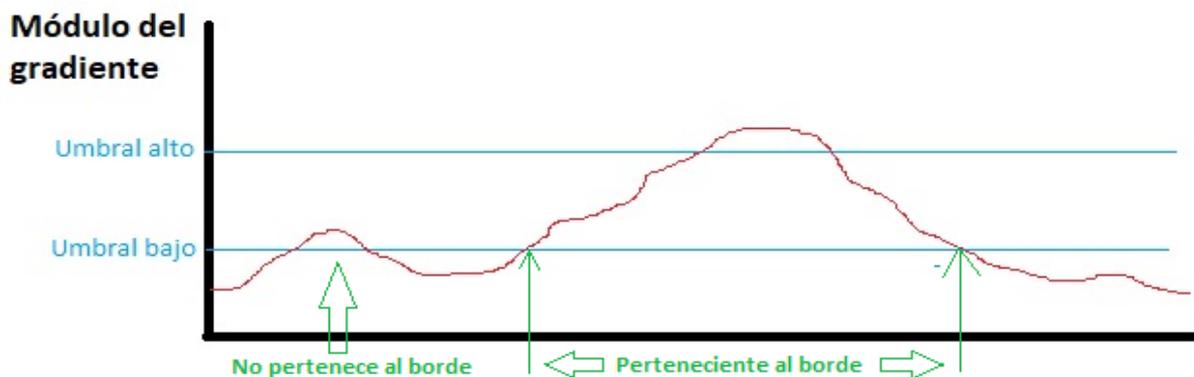


Figura 7. Método "*Hysteresis thresholding*"

Tras describir detalladamente en qué consiste el algoritmo de Canny, el siguiente paso ha sido escribirlo en lenguaje Python llamando a la función **cv2.Canny()** que se encuentra en la librería OpenCV y la cual nos realiza todos los pasos del algoritmo descritos anteriormente, el resultado obtenido es el siguiente:

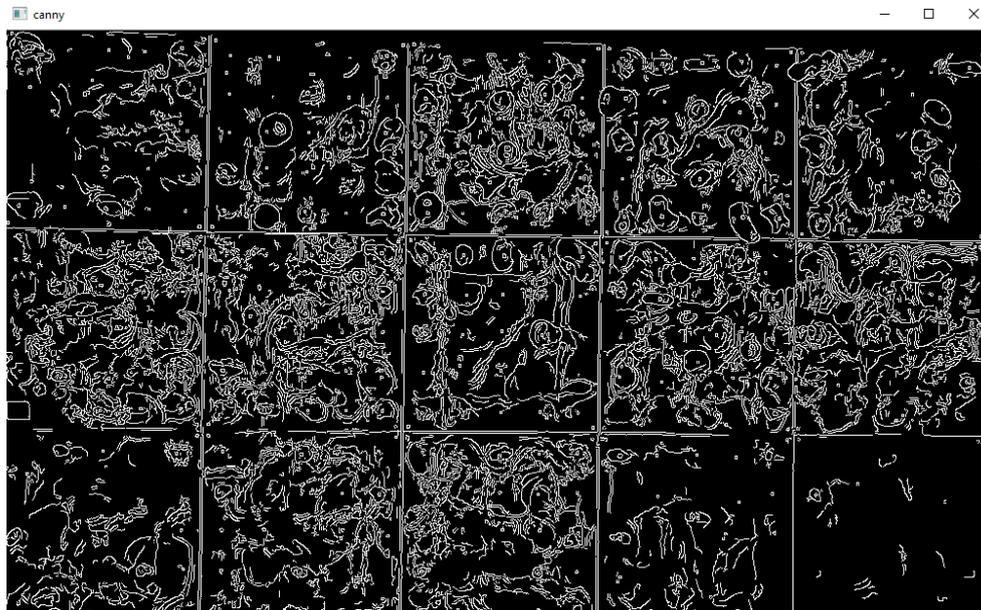


Figura 8. Detector de bordes aplicando el método de Canny Edge

Viendo que este resultado es bastante pésimo, se decide aplicar un filtrado bilateral previo a la imagen llamando a la función **cv2.bilateralFilter()**, el cual nos proporciona una reducción de ruido en ésta con un suavizado gaussiano, pero con la diferencia de que se conservan los bordes afilados, los resultados mejoran considerablemente como se muestra a continuación:

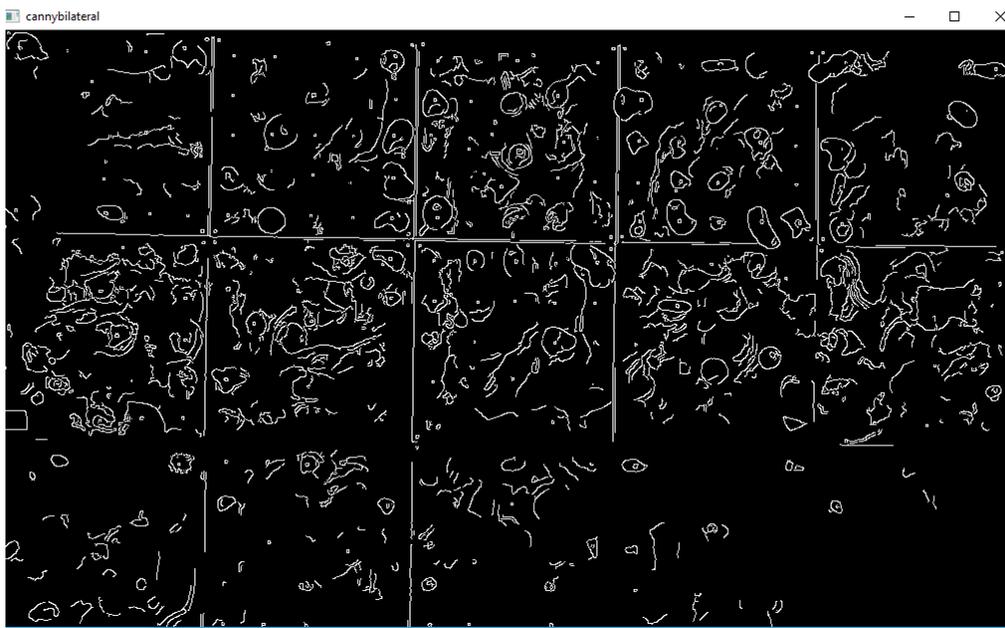


Figura 9. Detección de bordes por Canny con filtrado bilateral

No obstante, los resultados obtenidos aplicando el algoritmo de Canny con este tipo de filtrado, aun variando el doble umbral en éste, no son los deseados. Esto se debe a que la superficie de este rocódromo es un poco peculiar a diferencia de otros, ya que tiene muchos relieves y diferente tonalidad de marrón en ella. Por lo general, la mayoría de las superficies son planas y de color uniforme, y por lo consecuente, las presas se detectarían bastante bien utilizando solamente este algoritmo. A continuación, se demuestra lo dicho:



Figura 10. Detección de bordes con fondo uniforme

Dada esta situación nos damos cuenta de que se genera una confusión considerable para conseguir únicamente la detección de los bordes de las presas. Por lo tanto, ante la aparición del primer problema, se decide emplear máscaras de color como solución para diferenciar las presas del fondo, ya que estas presentan colores con un tono distinto al fondo como se puede observar en la imagen original. Este proceso se explicará en el apartado 4.1.1 de este proyecto.

4.1.1. Màscaras de color y operadores morfològics

Tras los resultados no deseados obtenidos anteriormente, se decide detectar las presas a partir de los diferentes colores que éstas presentan, por lo que realizaremos una nueva fotografía con una cámara Canon, y aumentaremos el contraste de la imagen, con el fin de proporcionar una mejor calidad en la tonalidad de los colores. La nueva fotografía por analizar es la siguiente:



Figura 11. Toma mejorada del rocódromo

Para detectar los colores de las presas, seguiremos una serie de pasos:

- Paso 1: Debemos convertir la imagen al rango de colores HSV (del inglés *Hue*, *Saturation*, *Value* –Matiz, Saturación, Valor). Cada valor del matiz corresponde a un color, la saturación se representa como la distancia al eje de brillo negro-blanco, y, por último, el valor representa la altura en el eje blanco-negro. Para ofrecer una vista mas sencilla de este modelo de color, se muestra a continuación el cono del modelo HSV:

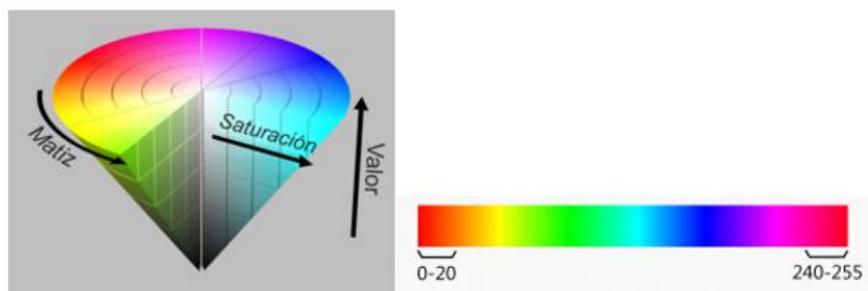


Figura 12. Representación del modelo HSV

- Paso 2: Ahora debemos de crear un rango para cada color, en nuestro caso, definiremos cuatro rangos de colores fundamentales: azules, verdes, blancos y dos para rojos, ya que, si se observa la rueda de color HSV, el valor del matiz de los rojos está entre 0 – 20, y 240 – 255 (más o menos). Tras múltiples pruebas se han elegido los valores más adecuados con el fin de conseguir un resultado lo más equilibrado posible. A continuación, se crean las máscaras atendiendo a los distintos rangos de colores creados y, por último, fusionamos todas las máscaras en una sola y la mostramos por pantalla:

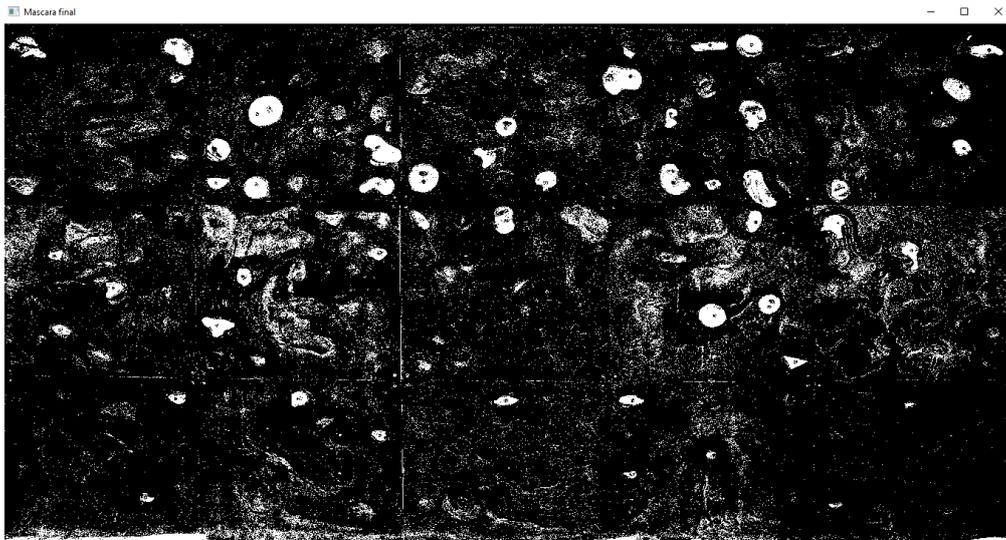


Figura 13. Imagen binaria correspondiente a la mascara

Dado que el resultado correspondiente a la máscara final presenta mucho ruido, como solución se decide realizar la operación morfológica binaria de apertura o también llamada *opening* cuya función es reducir dicho ruido en una imagen binaria y simplificar formas, entre otras cosas. Para entender esta operación, es imprescindible tener claras otras dos operaciones: erosión y dilatación, ya que estas últimas van intrínsecas en ella.

- Erosión: Se obtiene un efecto de adelgazamiento del área que compone la imagen:



Figura 14. Operación morfológica de erosión

- Dilatación: Obtenemos el efecto de expandir la región de la imagen que estemos trabajando, es lo inverso a la erosión:



Figura 15. Operación morfológica de dilatación

Por lo tanto, esta la operación de *opening* que aplicaremos sobre la máscara surge del resultado de aplicar una erosión y posteriormente dilatación, lo que nos servirá para eliminar el ruido que se encuentre fuera de la estructura. El resultado obtenido ha sido el siguiente:

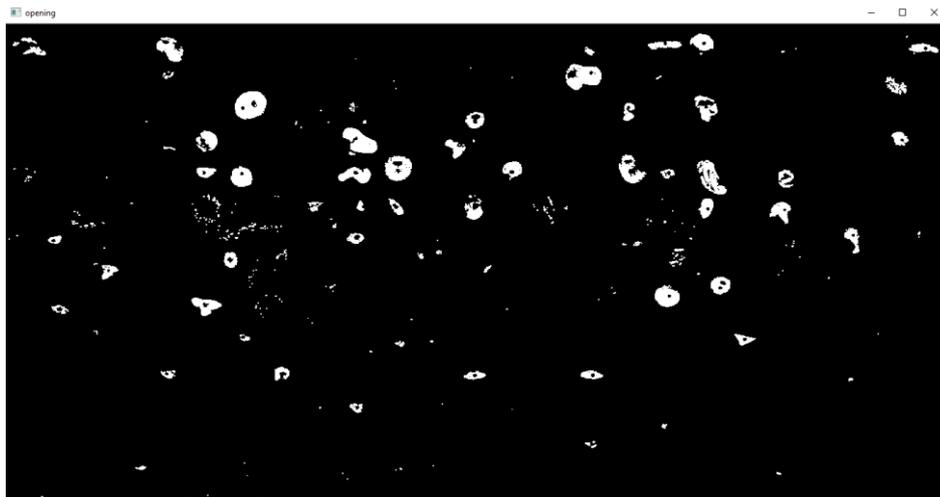


Figura 16. Máscara final tras aplicar la operación opening

Aplicamos Canny para quedarnos solamente con los bordes de las presas, y obviar el resto:

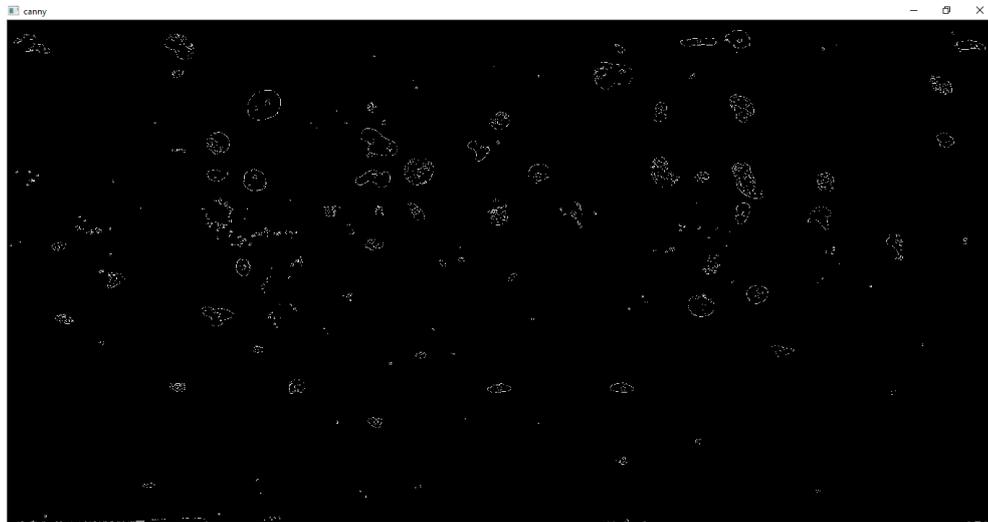


Figura 17. Detector de bordes de Canny a la máscara final

Obviamente, los resultados han mejorado muchísimo respecto a la imagen binaria de la máscara. Para observar mejor los resultados obtenidos, dibujamos una serie de contornos refiriéndonos a la imagen binaria de la detección por Canny aplicada sobre la operación morfológica de apertura, plasmándolos en la imagen original y para ello llamamos a las funciones `cv2.findContours()`:

El resultado es el siguiente:

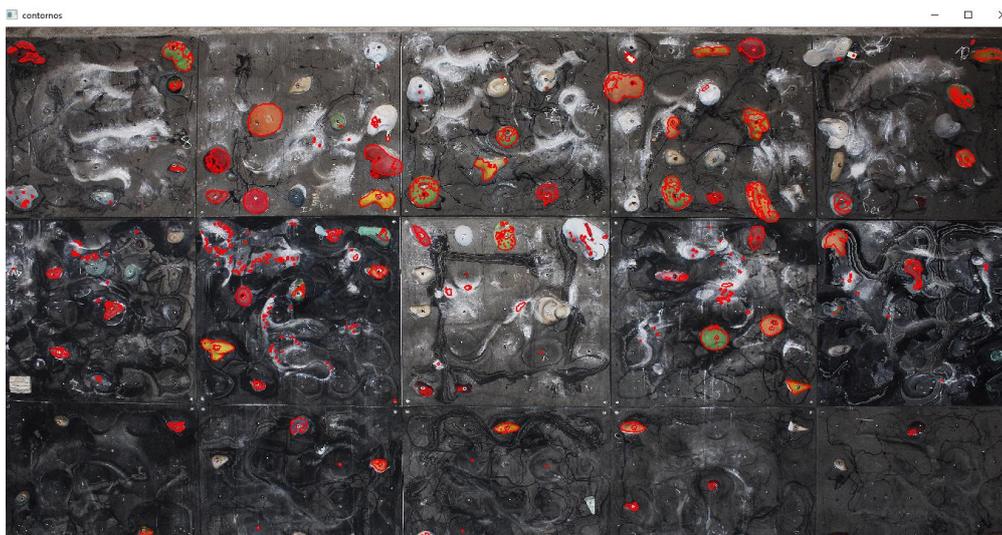


Figura 18. Detección de contornos de las presas

Como conclusión, podemos comentar que los resultados han mejorado bastante, obteniendo una efectividad de detección un 80%, teniendo en cuenta que tenemos unas 92 presas, y no detectamos aún 18. Se debe conseguir una efectividad del 100%, para que el mapeo sea correcto, por lo que surge una nueva idea que quizás pueda cumplirla, ésta se detalla en el siguiente apartado.

4.1.2. Implantación de etiquetas amarillas y detección de centros

Como hemos comentado en el anterior apartado, dada la situación, se va a proceder a realizar un nuevo experimento que será el de cubrir las presas del rocódromo una a una con unas bolsas de color amarillo chillón para realizar el mapeado de detección, con el fin de filtrar un único color y mejorar la eficiencia de la detección. Para que nos hagamos una idea, el escenario montado, aunque falten las presas de arriba por cubrir, fue el siguiente:



Figura 19. Tercera toma del rocódromo con etiquetas amarillas

Tras cubrir todas las presas, se realizó el proceso de detección a tiempo real realizando un filtrado mediante máscara de color a la cual se le asociaron unos valores ajustados a ese tono de amarillo. Cuando ejecutamos la aplicación comprobamos que el funcionamiento no defraudó, se detectaban todas las presas.

Ahora bien, como podemos observar desafortunadamente debemos de tener en cuenta un factor en contra como es el de la iluminación no uniforme y que afectará por tanto a la variabilidad de ese tono amarillo complicando en el proceso de detección de las presas, por lo que tenemos que realizar el experimento con unas condiciones de iluminación adecuadas, es decir, en un horario en el que el sol no incida directamente en la ventana, para conseguir el correcto funcionamiento.

Una vez hemos detectado el contorno de todas las presas, el siguiente paso será el de buscar el centro de cada una de las presas detectadas, con el fin de guardar cada uno de estos centros como un punto con sus respectivas coordenadas (x e y) dentro de un de la imagen. Para cumplir el objetivo, recurrimos a la función **cv2.moments()** con la que podemos hallar las coordenadas del centro del área del contorno encontrada. Para calcular dichas coordenadas del punto se realizan las siguientes operaciones:

$$\bar{x} = \frac{M_{10}}{M_{00}} \quad \bar{y} = \frac{M_{01}}{M_{00}}$$

Donde M_{00} es el área del contorno

Por tanto, cada una de las coordenadas correspondientes a los centros se guardarán en un array de centros, al cual recurriremos para realizar los juegos interactivos a tiempo real. Estos centros se representan en forma de círculos blancos (función **cv2.circle()** de OpenCV) sobre un fondo negro para que, a la hora de dar paso a el proceso de proyección, dichos puntos blancos se proyecten sobre las presas para destacarlas respecto al fondo negro que cubrirá la superficie restante. A continuación, colocamos la cámara de tal forma que se capte toda la superficie con una perspectiva lo más óptima posible con el fin de encontrar todos los centros de las presas, el resultado ha sido el siguiente:

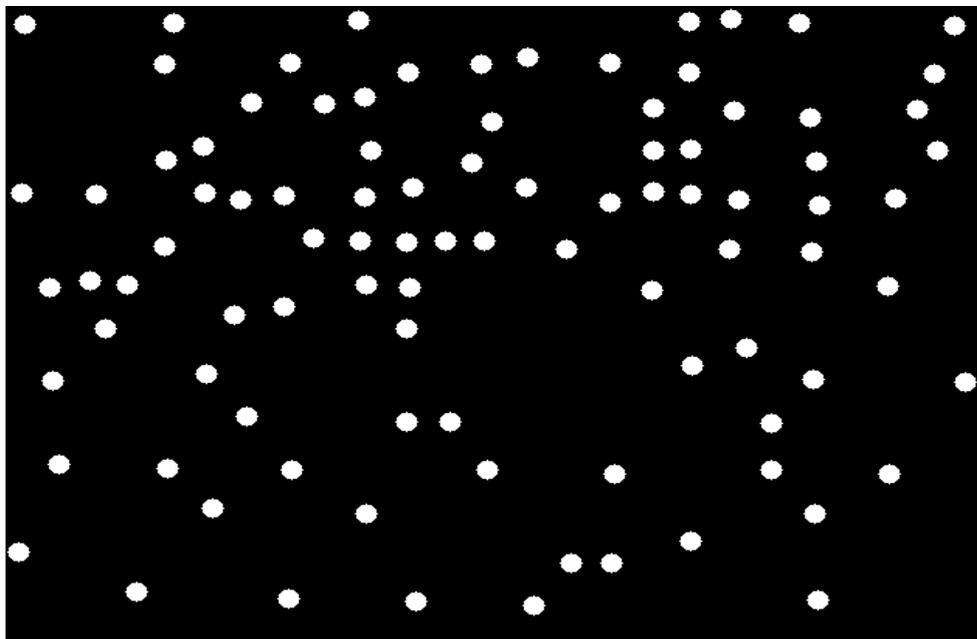


Figura 20. Centros de las presas del rocódromo

Llegado a este punto debemos de darle mera importancia a la obtención de todos los puntos correspondientes a las presas del rocódromo, ya que si no tuviéramos esta información no podríamos desarrollar ningún tipo de juego interactivo debido a que en el momento en el que el usuario entra en contacto con alguna de las presas y el punto de esta no lo tuviéramos guardado no habría efecto alguno, por tanto no tiene ningún sentido desarrollar cualquier juego sin antes haber obtenido dicha información.

4.2. Desarrollo de la aplicación

En este capítulo se va a explicar el desarrollo de cada uno de los juegos interactivos que se han creado, en los que los movimientos que realiza el usuario en todo momento van a ser captados por la cámara a tiempo real y por lo tanto el sistema se irá actualizando acorde estos movimientos. Los juegos que se han desarrollado han sido los siguientes:

Juego 1: “Elige una ruta”.

En este primer juego, y el más sencillo, el usuario tiene la libre decisión de crear cualquier tipo de ruta trazando líneas de una presa a otra conforme a este más le convenga dependiendo del tipo de entrenamiento que el usuario guste. Además, existen rutas ya predefinidas atendiendo a diferentes niveles de dificultad y que el usuario podrá seleccionar. Las coordenadas de los puntos correspondientes a cada una de las presas sobre los que hace “clic” el usuario son mostrados por consola. Además, también tenemos una barra que permite seleccionar el grosor de la línea que conforma la trayectoria, el cual determina la prioridad de elegir un camino u otro.

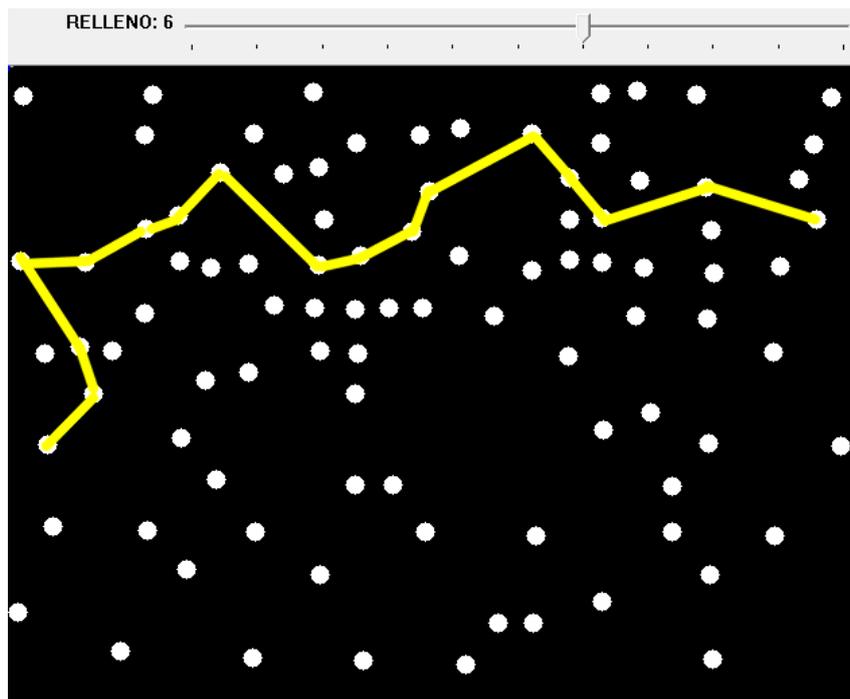


Figura 21. Ejemplo de ruta personalizada

Juego 2: “Splatoon”

Splatoon es un juego multijugador en el que cada usuario es identificado con una pulsera de distinto color, con el fin de pintar el mayor número de presas en 40 segundos. Además, ambos jugadores, tendrán la posibilidad de pintar una presa que ya haya sido pintada anteriormente por el contrincante, es decir, de sobrescribirla con su color. En el juego se mostrará la puntuación de cada jugador a tiempo real, así como el tiempo de juego transcurrido. Una vez finaliza el tiempo, se abrirá una ventana con una imagen que muestra un logo con la palabra *winner* de color azul o amarillo dependiendo del ganador o un logo con la palabra *empate* en caso de empate. En la siguiente figura se muestra un ejemplo de manera ilustrativa del funcionamiento básico:

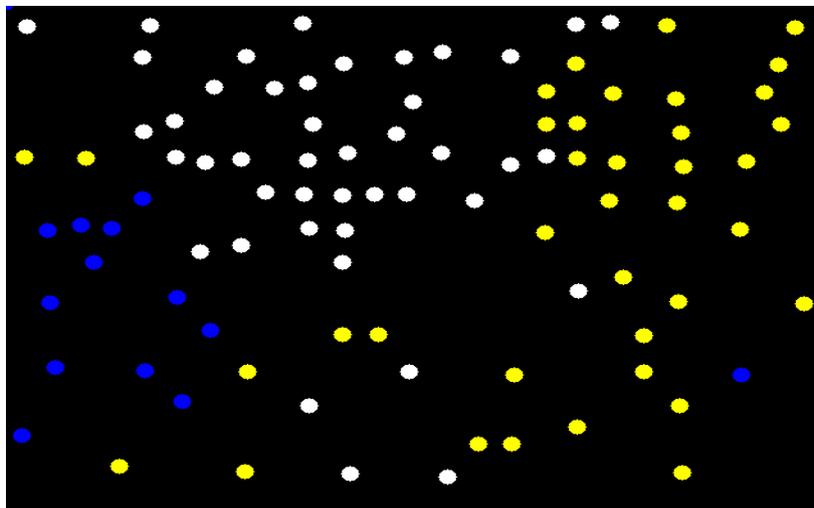


Figura 22. Ejemplo de prueba de Splatoon

Como en este caso existen mas presas coloreadas de azul que de amarillo, se mostrará el logo de ganador en azul, con el resultado final justificando que el jugador ha ganado en la esquina superior izquierda, como se muestra a continuación:



Figura 23. Símbolo del ganador del juego

Estas capturas se han obtenido desde la aplicación en el ordenador, en el capítulo 4.3.2, se mostrarán las fotografías del rocódromo practicando este juego, así como el análisis de los resultados obtenidos tras la realización del juego.

Juego 3: “Multijugador: Pingpong”

El pingpong en 2D es uno de los primeros videojuegos que se crearon para el ordenador y que intenta simular de alguna manera uno de los deportes más practicados en el mundo y derivado del tenis, el Pingpong. Este deporte se practica entre dos jugadores o parejas (en nuestro caso, dos jugadores) sobre una mesa rectangular dividida de manera transversal por una red. Existe una pelota que es golpeada de manera alternativa por cada jugador con una paleta haciendo que ésta rebote en el campo contrario, de tal manera que, si uno de los jugadores no logra devolver la pelota al campo contrario, el oponente logra conseguir un punto. El jugador que consiga 11 puntos gana el juego.

De la misma manera, en este videojuego 2D, el campo corresponderá con una superficie rectangular con un fondo de color negro y con una línea blanca divisoria por la mitad que determinará la asignación de los campos de cada jugador los cuales disponen de una barra vertical que simula la raqueta de cada uno y que, en un principio, se mueve con el teclado para evitar que la pelota entre en cada campo y evitar así que el oponente gane un punto. A continuación, se muestra el ejemplo:

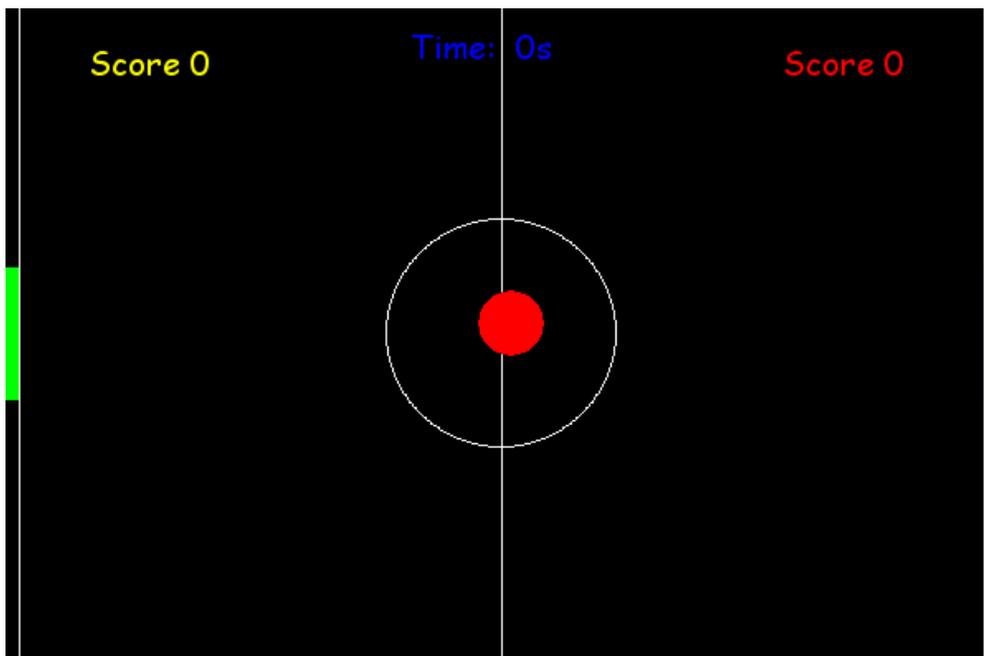


Figura 24. Pingpong 2D

Ahora bien, teniendo en cuenta el funcionamiento del videojuego, el objetivo que nos hemos propuesto es innovarlo de tal manera que en la superficie del rocódromo podamos disfrutar de un Pingpong en realidad aumentada a la vez que escalamos. Para ello hemos utilizado las librerías pygame y random para realizar el videojuego y para el movimiento de la pelota respectivamente.

En un principio surgió la idea de captar la silueta de cada usuario a tiempo real y que cuando la pelota entrase en contacto con dicha silueta, ésta rebotara hacia el campo contrario. Cuando se procede a realizar el experimento, nos damos cuenta de que no se dispone de un hardware lo suficientemente rápido computacionalmente como para cumplir dicho objetivo, ya que captar la silueta de una persona a tiempo real se requiere de un sistema de captación bastante robusto debido a que la persona se mueve muy rápido y la detección de bordes requiere una umbralización local (umbral de intensidad alrededor de cada píxel) muy estricta.

Posteriormente surge una idea mucho más sencilla respecto a la primera ya que ésta nos permite movernos por toda la superficie del rocódromo y que por lo tanto es más completa, pero que será lo suficientemente práctica como para poder cumplir el objetivo: cada jugador se identificará con una pulsera amarilla en un caso, y azul, en el otro, como en el anterior juego. En este momento, cuando la cámara capte el color amarillo o azul de la pulsera que posee cada jugador, y éste se disponga a tocar pelota esta barra se moverá verticalmente a la misma posición donde se encuentre dicha pulsera a tiempo real. Esto se consigue tras haber relacionado el movimiento de la barra con un filtrado por máscara de color amarillo o azul respectivamente a cada jugador. Finalmente, el primer jugador que llegue a 11 puntos será el ganador y saldrá por pantalla el símbolo de ganador azul o amarillo respectivamente como en el anterior juego, reflejando el resultado del partido.

Por último, cabe comentar que tanto las fotografías como la descripción de resultados obtenidos tras la práctica de este juego se explicaran en el posterior capítulo 4.3.2.

4.3. Proyección y análisis de los resultados

En este apartado se va a explicar como tanto los resultados obtenidos tras realizar diferentes pruebas en un entorno de trabajo real, es decir, en el rocódromo, así como el proceso de auto-calibración previo que resulta de vital importancia para establecer cada dispositivo en el lugar que le corresponde, así como el cálculo de las posiciones relativas para hacer posible el fenómeno de realidad aumentada.

4.3.1. Proceso de auto-calibración

En este capítulo como se ha citado anteriormente, surge la necesidad de determinar la posición relativa entre cámara, proyector y superficie de proyección y por ello aparece la obligación de crear un proceso de auto-calibración que nos permitirá mediante el tratamiento de la imagen capturada y la proyección de diferentes imágenes de referencia calcular dichas distancias.

Esta resolución de cálculo de distancias dará pie a que la aplicación pueda efectuar la realización de una homografía, es decir, una transformación proyectiva entre la imagen capturada por la cámara y la imagen que será proyectada, de tal forma que cada uno de los puntos y rectas de una imagen le corresponden respectivamente, un punto y una recta de la otra, con el fin de poder crear la sensación de realidad aumentada en el usuario.

Teniendo clara la previa y breve introducción del presente capítulo, se da paso a ejecutar dicho proceso de auto calibración que seguirá una serie de pasos:

1. **Posicionamiento de la cámara y proyector.** En este paso es necesario que la cámara, al menos, capte la superficie de la pared sobre la que se está proyectando, ya que si esto no fuese posible evidentemente sería imposible detectar la posición del escalador a través de la cámara. Por lo tanto, el proyector también se debe situar a una distancia razonable con el fin de poder facilitar la captación de dicha proyección por parte de la cámara. En la siguiente figura se muestra un sencillo croquis aclaratorio de este proceso:

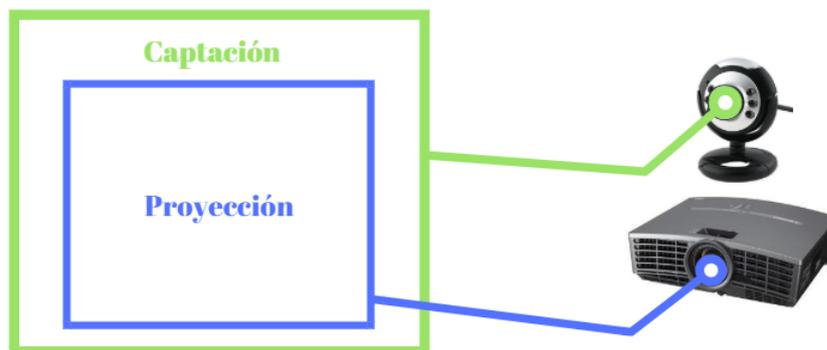


Figura 25. Requisito de posicionamiento entre cámara y proyector

2. **Proyección de una imagen de referencia.** Tras certificar que el posicionamiento de la cámara y el proyector es correcto atendiendo a las condiciones comentadas anteriormente, el segundo paso consiste en proyectar una imagen de referencia sobre la superficie la cual será capturada por la cámara y posteriormente procesada por el ordenador con el fin de determinar la posición relativa entre la cámara y el proyector. Este proceso nos permitirá obtener la información correspondiente a los centros de cada una de las presas que se encuentran dentro de la proyección, por lo que aquí es donde se hace evidente que la detección de las presas se puede realizar en cualquier superficie atendiendo previamente a este proceso de proyección de una imagen de referencia.

Además, como obtenemos la posición de cada presa que se encuentra dentro de la proyección, tenemos la libertad de proyectar una imagen de referencia en una sola parte de la superficie del rocódromo si interesara hacerlo de esa forma como por ejemplo si se realizara un entrenamiento en el que las presas de arriba del todo no fueran necesarias o solamente con la mitad de la superficie fuera suficiente para realizar dicho ejercicio. En nuestro caso tal y como se puede ver en los videos de demostración, la proyección abarcará toda la superficie al completo ya que para realizar los juegos desarrollados en nuestra aplicación se requiere de la utilización de toda ella. En la siguiente figura se observa un sencillo ejemplo del proceso explicado:

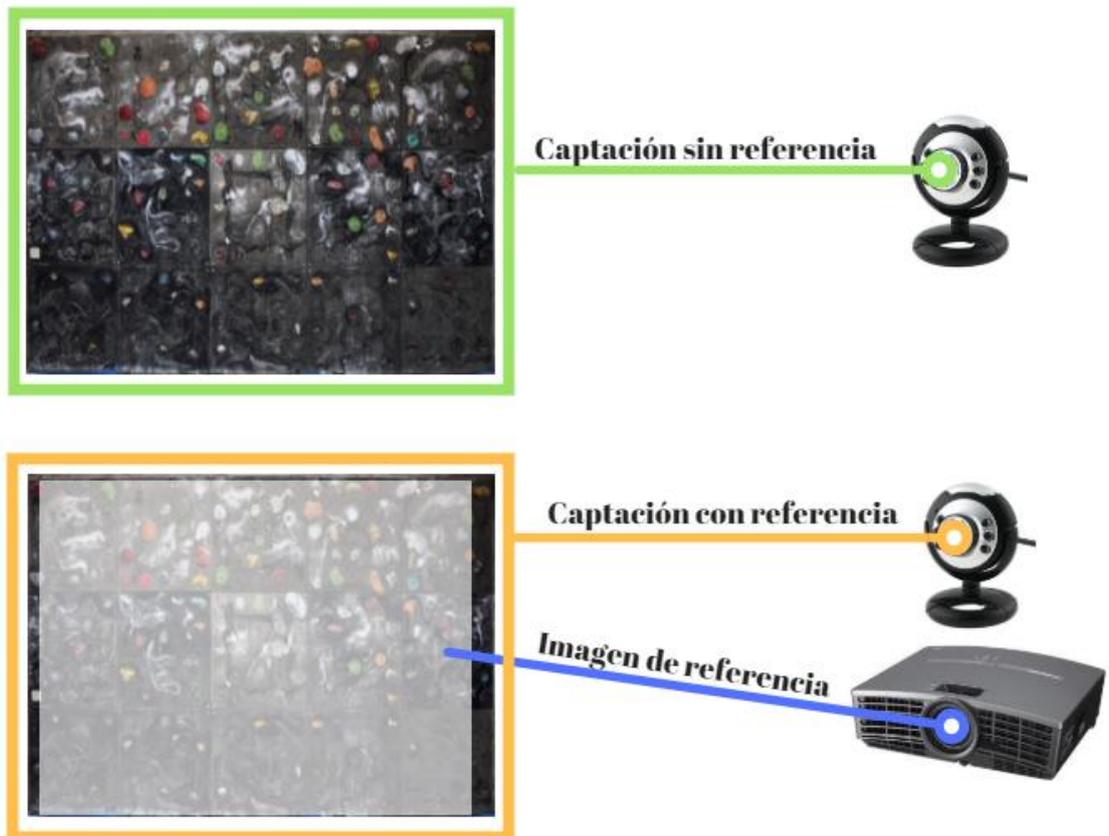


Figura 26. Captación de la imagen de referencia

En este momento en el cual se puede obtener la diferencia entre la captación con la imagen de referencia respecto a la captación, proceso que denominaremos como sustracción de fondo. Esta sustracción resulta de vital importancia para obtener los cuatro puntos correspondientes a cada una de las esquinas de la imagen de referencia que son necesarios para realizar el cálculo de la matriz homográfica, el cual se realiza de forma automática mediante la función `cv2.findHomography()` que nos proporciona OpenCV y que se compone de dos arrays principales; un array con los puntos correspondientes a los cuatro puntos del rectángulo blanco o imagen de referencia tras realizar la sustracción de fondo y por otro lado un array con los puntos correspondientes al tamaño de la imagen de destino tras procesar que se desee. El orden de los puntos del rectángulo blanco tras restar la imagen de referencia se muestra en la siguiente figura:



Figura 27. Puntos de las esquinas de la imagen de referencia

Posteriormente, una vez hemos obtenido la matriz de homografía H , la aplicamos en el proceso de homografía en el que una imagen que es capturada a través de la cámara siempre será ajustada al tamaño de procesamiento que nosotros deseemos y que insertaremos manualmente en el código dentro de la función `cv2.warpPerspective()`. Dentro de ésta función, insertaremos la imagen que se captura con la cámara con la imagen de referencia, la matriz de homografía H calculada anteriormente, y el tamaño deseado de la imagen de procesamiento. Por lo tanto, si los puntos de referencia se han calculado correctamente, la imagen de homografía que será emitida por el proyector cuadrará con la posición original.

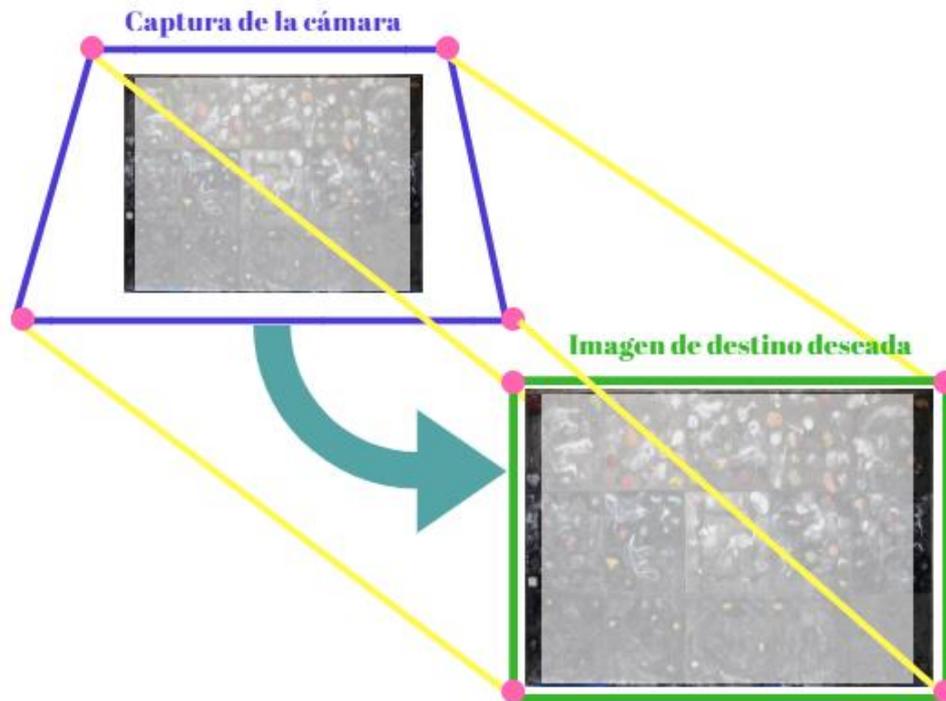


Figura 28. Proceso de homografía y cambio de perspectiva

Una vez hemos realizado todo el proceso de homografía correctamente, el siguiente paso será la realización del tratamiento digital de la imagen para detectar las presas y poder así realizar los juegos interactivos, dicho procesado ya se ha explicado en el capítulo 4.1.

4.3.2. Resultados

En este capítulo se procederá a realizar la proyección de cada uno de los juegos interactivos desarrollados tras haber comprobado su correcto funcionamiento con la realización de múltiples pruebas y corrección de errores a tiempo real.

Dicha proyección resulta de vital importancia en este proyecto para poder referirnos al concepto de realidad aumentada, ya que como bien sabemos, la realidad aumentada es la visión de un entorno físico del mundo real (en nuestro caso, el rocódromo) a través de un dispositivo tecnológico (el ordenador) que añade información virtual a la información física que ya existe. En conclusión, el entorno físico, se combina con el entorno virtual, dando lugar a la creación de la realidad aumentada en tiempo real.

En un primer instante, se implantó la instalación tal y como hemos comentado previamente en el capítulo 3.3, con el proyector apuntando hacia la superficie del rocódromo a una distancia razonable con el fin de cubrirla por completo y poder detectar todas las presas, aunque como se explicará a continuación podríamos proyectar y jugar en una sola parte del rocódromo y no necesariamente abarcar toda la superficie al completo.

Uno de los problemas que aparece en esta etapa de proyección tiene que ver con el tono oscuro que presenta la superficie del rocódromo ya que, como bien sabemos, los tonos negros u oscuros absorben mucha luz al contrario que los blancos que la reflejan. Por ello todas las proyecciones se realizan sobre pantallas blancas, ya que con el mismo número de lúmenes que ofrece el proyector se consigue una visibilidad muchísimo mejor. Por lo tanto, retomando lo anterior, cuando proyectamos sobre el rocódromo apenas se aprecia la proyección debido a la causa explicada previamente sumada de las pésimas condiciones de iluminación ya que tenemos un ventanal al lado. Para solucionar este problema se ha tenido que realizar la proyección con unas condiciones de iluminación nocturnas para poder grabar los videos de demostración del funcionamiento de los juegos.

El escenario que se ha montado se observa en las siguientes ilustraciones:



Figura 29. Montaje del escenario

Una vez hemos establecido la posición de los dispositivos de captura y proyección de forma adecuada, debemos tener en cuenta el principio básico por el cual funciona toda aplicación que es la determinación de la posición relativa entre la cámara y el proyector respecto a la superficie de calibración, el cual se explica detenidamente en el anterior capítulo de proceso de auto-calibración de este proyecto.

Retomando lo anterior, una vez que ya se ha hecho efectivo el proceso de auto calibración, es hora de pasar al ámbito práctico y proyectar cada uno de los juegos en el rocódromo tras la realización de infinidad de pruebas y corrección de errores en el código de cada uno de ellos y por tanto llevar a efecto la correspondiente demostración del correcto funcionamiento de la aplicación en un entorno de trabajo real.

En primer lugar, se ejecuta el *Juego 1: "Elige tu ruta"* en el que hemos seleccionado la opción de crear una ruta personalizada para realizar la demostración del mismo. Las líneas que conforman la trayectoria de la ruta se han seleccionado del mismo grosor, es decir, no existen rutan alternativas. La visualización de estas es bastante notoria en la superficie de escalada, así que, se podría decir que el color amarillo ha sido el adecuado para trazar las trayectorias. A continuación, se muestra en la siguiente figura la demostración de este primer juego:



Figura 30. Resultado de "Juego 1: Elige tu ruta"

Posteriormente, se ejecuta el segundo juego conocido con el nombre de Splatoon. Debemos de tener en cuenta la detección del color de cada una de las pulseras era buena en un principio cuando se han realizado las pruebas en un primer plano, pero el procesado ha resultado ser un poco lento debido a que la distancia de la cámara a la superficie de escalada era bastante lejana respecto a la calidad de captación que ésta nos ofrecía, aunque ha sido suficiente para demostrar el funcionamiento.



Figura 31. Resultado de "Juego 2: Splatoon"

Por último, se hace correr la última aplicación desarrollada que se corresponde con el juego multijugador: Pingpong. Al igual que en el caso anterior, el procesado ha sido un poco costoso a la hora de la detección de los colores de las pulseras, pero hemos podido disfrutarlo y demostrar el funcionamiento. Por otro lado, y en el mismo acto de proyección de este juego, se planteó la idea de aumentar la velocidad de la pelota modificando el código manualmente en ese mismo instante para incrementar el nivel de dificultad y hacer el juego un poco más variado.

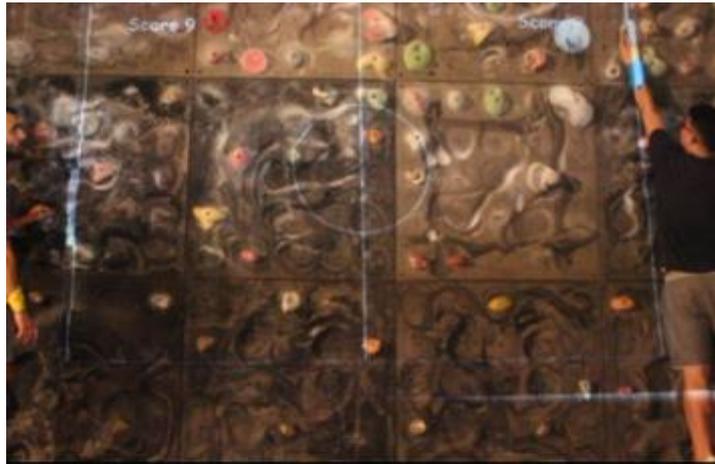


Figura 32. Demostración de "Juego 3: Multijugador Pingpong"

Como conclusión, cabe mencionar que en el video de demostración que forma parte del anexo de este proyecto se pueden observar de manera mucho más animada la realización de estos tres juegos y verificar todos los resultados comentados, así como todo el proceso previo para llegar a ellos.

4.4. Interfaz de la aplicación

Para finalizar este proyecto, se ha creado una interfaz gráfica de usuario bastante sencilla y accesible a cualquiera que desee experimentar este sistema interactivo. En primer lugar, tenemos el menú principal donde tenemos dos botones: uno para dar paso a el menú de selección de cada uno de los juegos y otro para leer las instrucciones necesarias previas para realizarlos correctamente.

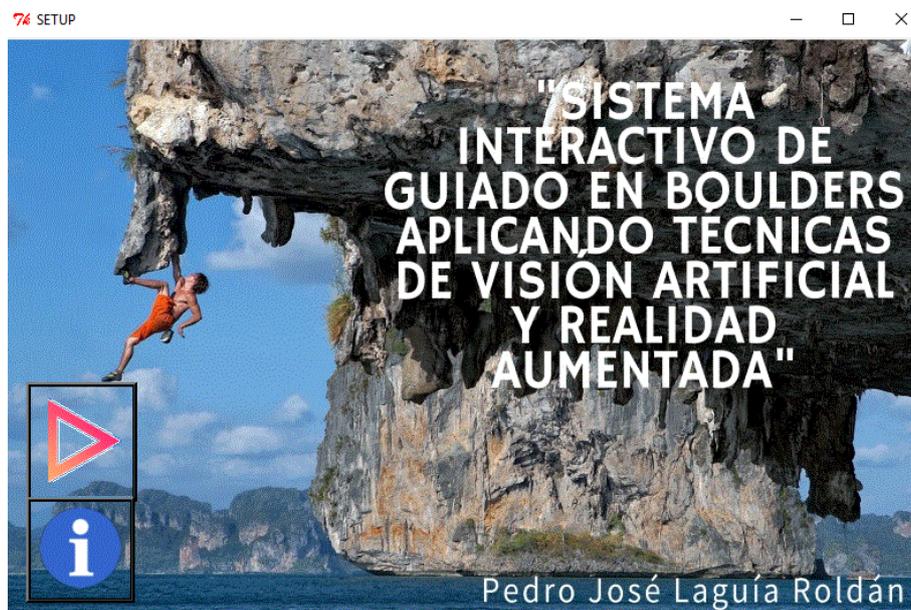


Figura 33. Menú principal de la aplicación

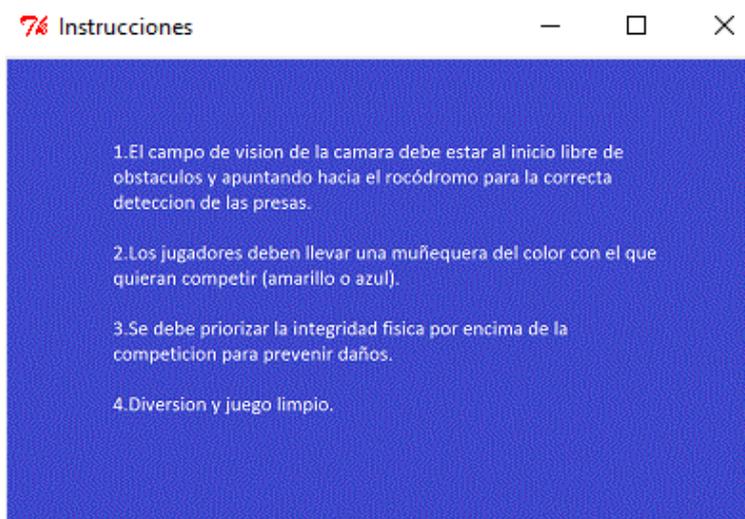


Figura 34. Instrucciones de juego

En el menú principal tenemos tres botones a los cuales les hemos asignado una imagen para que sea mucho más gráfica la selección de los tipos de juegos desarrollados:



Figura 35. Menú de juegos

Cuando pulsamos el primer juego nos salta una nueva ventana para seleccionar el nivel de dificultad de la ruta a escalar, o bien, para realizar una ruta personalizada a gusto de cada usuario.

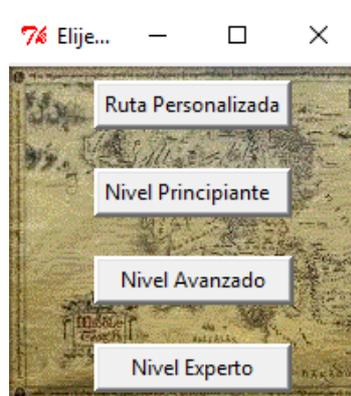


Figura 36. Menú de selección de ruta

Por último, cuando se seleccionan tanto el botón de Splatoon como el de Pingpong, simplemente se ejecutan los ficheros .exe y se da paso a la realización del juego directamente.

5. Conclusiones y trabajo futuro

En última estancia, debemos de darle mera importancia al capítulo actual en el cual verificamos el cumplimiento de los objetivos propuestos en este proyecto, ante todas las dificultades que se han presentado a lo largo del mismo. Con estos objetivos hacemos referencia tanto a la creación de una aplicación bastante accesible a nivel de usuario que permita seleccionar diferentes tipos de juego relacionados con la escalada y que posteriormente se proyectarán en el rocódromo, así como a la aprobación de la buena sensación de haber experimentado la realidad aumentada por parte del usuario sin una experiencia previa necesaria en dicho deporte para poder disfrutarla.

Resulta oportuno tener en cuenta varias ventajas con las que hemos podido desarrollar el proyecto. En primer lugar, la accesibilidad al recinto polivalente del campus de Gandía para realizar pruebas ha sido bastante cómoda y en muchas ocasiones el recinto estaba vacío por lo que el tiempo para realizar detenidamente los experimentos no ha supuesto ningún inconveniente. En segundo lugar, la solidaridad para la prestación del material por parte de la escuela, como es el caso del proyector. En tercer lugar, debemos destacar el potencial que posee la librería OpenCV, que nos permite desarrollar infinidad de aplicaciones tanto de visión artificial como de realidad aumentada con el aliciente de escribir dicho código con lenguaje Python, el cual es muy intuitivo y legible. En tercer y último lugar, hay que destacar que podemos disfrutar de la aplicación en cualquier superficie con la realización del proceso de calibración previo a la realización de los juegos.

Por otro lado, teniendo en cuenta las consideraciones ventajosas previamente comentadas, es evidente por lo tanto tener en cuenta los impedimentos o inconvenientes que nos han dificultado hacer este proyecto algo más profesional. Para empezar, la superficie rocosa del rocódromo ha sido una de las principales dificultades de este proyecto para poder detectar las presas con efectividad junto con el gran ventanal que se situaba justo al lado y que nos introducía el problema de la iluminación no uniforme. Luego, con el fin de solucionar este problema hemos tenido que implantar etiquetas en las presas, por lo que ha sido necesario manipular las instalaciones del campus para conseguir la detección mediante máscaras de color con umbrales muy estrictos debido al problema de la iluminación comentado. Finalmente, la cámara responsable de realizar la captación no tenía calidad suficiente, a ello se ha de agregar el coste computacional que la aplicación ha requerido.

Como conclusión de este proyecto, hemos de comentar que los resultados han sido satisfactorios y éstos los cuales se pueden observar en un video de demostración. Además, se ha de valorar la importancia de la docencia recibida de asignaturas previamente impartidas en la carrera como Tratamiento Digital de la Imagen fundamentalmente y todas las relacionadas con la programación.

Como propuesta de futuro para la ampliación y mejora de este proyecto, se propone hacer uso de una Kinect para reconocer la silueta del usuario a la perfección y que se permita jugar con todo el cuerpo, elevar la velocidad de procesamiento de la aplicación, y por último la posibilidad de realizar más variedad de juegos.

6. Bibliografía

1. Valo Motion Oy 2018. “ValoClimb, The Original Augmented Climbing Wall”.
<http://valomotion.com/valoclimb/>
2. Weyne. “Instalación de Python y OpenCV en Windows”. 15 de Mayo,2017.
<https://youtu.be/7fEguWHxnkg>
3. Pablo Aguilar García, “TFG Realidad aumentada aplicada a la escalada”.
8 de Septiembre,2017.
<https://riunet.upv.es/bitstream/handle/10251/91788/AGUILAR%20-%20Realidad%20aumentada%20aplicada%20a%20la%20escalada.pdf?sequence=1>
4. Iván F. Lobo, “Escalar montañas como si fuera un videojuego”, Universidad de Indiana. 5 de Febrero,2017
<http://elfuturoesapasionante.elpais.com/escalar-montanas-videojuego/>
5. Anónimo, “Blog: Tutor de programación”. 2017.
 - a. “OpenCV: Operaciones morfológicas”
 - b. “Uso del mouse y trackbar OpenCV”.
 - c. “Funciones de dibujo: OpenCV y Python”
<http://acodigo.blogspot.com>
6. OpenCV by Doxygen license: Copyright by Dimitri van Heesch. “OpenCV modules”. 24 de Octubre, 2017.
<https://docs.opencv.org/3.3.1/>
7. PyImageSearch by Adrián Rosebrock , “OpenCV: Center of contours”.
Febrero,2016.
<https://www.pyimagesearch.com/2016/02/01/opencv-center-of-contour/>
8. Blog Programafacil creado por del Luis del Valle : “Detector de bordes Canny cómo contar objetos con OpenCV y Python”: 11 de Abril,2017.
 - a. “Capitulo 6: Buscar los contornos de una imagen”.
 - b. “Capitulo 7: Dibujar los contornos en una imagen con OpenCV”
https://programafacil.com/blog/vision-artificial/detector-de-bordes-canny-opencv/#Buscar_los_contornos_de_una_imagen
9. Sergio Canu: “Object tracking using homography opencv 3 4 with Python 3 tutorial 34”. 5 de Junio,2018.
<https://pysource.com/2018/06/05/object-tracking-using-homography-opencv-3-4-with-python-3-tutorial-34/>

7. Anexos

Anexo 1: “Código comentado del proceso de homografía.pdf”

Anexo 2: “Código comentado del Juego 1: Elige tu ruta.pdf”

Anexo 3: “Código comentado del Juego 2: Splatoon.pdf”

Anexo 4: “Código comentado del Juego 3: Multijugador Pinpong.pdf”

Anexo 5: “Códigos y ejecutables.zip”

Anexo 6: “Figuras del proyecto.zip”

Anexo 7: “Video de demostración del proyecto.mp4”