

# Implementación de un algoritmo para detección del movimiento en una aplicación de realidad virtual

## *Implementation of an algorithm for motion detection in a virtual reality application*

Fecha de recepción: 2022-07-29 • Fecha de aceptación: 2022-08-22 • Fecha de publicación: 2022-10-10

**Jonathan Marcelo Tipán Espinoza**  
Universidad Central del Ecuador, Ecuador  
[jmtipane@uce.edu.ec](mailto:jmtipane@uce.edu.ec)  
<https://orcid.org/0000-0001-8960-4190>

### RESUMEN

El avance tecnológico en los últimos años es cada vez mayor, su uso repercute de manera directa e indirecta en la vida cotidiana de las personas. Haciendo uso de este progreso se propone una aplicación que permita a personas con cierto grado de discapacidad interactuar mediante la visión computacional, en particular la detección de movimiento con entornos que simulen la realidad. En primer lugar, se realiza una especificación del grupo a quien va dirigida esta propuesta, esto mediante la clasificación internacional del funcionamiento de la discapacidad y de la salud, con el fin de delimitar el público objetivo y poder seleccionar los algoritmos óptimos para esta situación. A continuación, se realiza un análisis de los métodos utilizados para el análisis de video que permitan detectar objetos y realizar un tracking de movimiento, proceso que dará como resultado la interacción de la persona con el software sin necesidad de periféricos extra.

**PALABRAS CLAVE:** algoritmos, seguimiento, interacción, interfaz, programa informático

## ABSTRACT

Technological progress in recent years is increasing, and its use has a direct and indirect impact on people's daily lives. Making use of this progress, we propose an application that allows people with a certain degree of disability to interact through computer vision, in particular the detection of movement with environments that simulate reality. First, a specification of the target group of this proposal is made, this by means of the international classification of disability and health functioning, in order to delimit the target audience and to be able to select the optimal algorithms for this situation. Next, an analysis of the methods used for video analysis that allow object detection and motion tracking, a process that will result in the interaction of the person with the software without the need for extra peripherals.

**KEYWORDS:** algorithms, tracking, interaction, interface, software

## Introducción

Según la Organización Mundial de la Salud, más de mil millones de personas en todo el mundo tienen alguna discapacidad, y, de estas, 200 millones de personas aproximadamente tienen alguna dificultad considerable en su funcionalidad (González et al., 2012). El término discapacidad es usado de manera genérica para referirse a deficiencias, limitaciones de actividad y restricciones para interactuar; esto incluye discapacidades sensoriales, cognoscitivas, intelectuales y psíquicas (Alban, 2018). Lograr que estas personas puedan comunicarse o interactuar en distintos ámbitos de su día a día, mejoraría su calidad de vida.

Para solucionar este problema hay que tomar en cuenta cómo los avances tecnológicos impactan en nuestra vida, por lo que el uso de dispositivos electrónicos se ha hecho prácticamente indispensable. De modo que este despliegue permite brindar soluciones robustas y económicas en diferentes ámbitos (Castillo, 2014). Sobre la base de esto se plantea una aplicación que, mediante el procesamiento de imágenes, permita detectar el movimiento de la persona. Esto permite que un usuario con discapacidad pueda interactuar e incluso comunicarse mediante una interfaz.

La premisa con la que se busca resolver este problema en principio cumple con el objetivo deseado, pero frecuentemente el tiempo de ejecución de los algoritmos y el coste de los equipos representan el principal problema (Gómez-Vega et al., 2015). Además, el procesado digital de imágenes demanda el manejo de un alto número de operaciones a nivel de bit, las cuales se desea ejecutar en el menor tiempo posible (Bravo, 2007).

Según lo descrito anteriormente se propone una aplicación basada en algoritmos que traduzcan el movimiento en tiempo real y sin retraso, a una interfaz en la cual el usuario pueda interactuar o comunicarse.

## Metodología

El proyecto será realizado con una metodología cualitativa y está dividido en 4 etapas:

1. Primera etapa: delimitar las personas a las que beneficiaría la aplicación. En esta etapa se investigará de manera general los grados de discapacidad descritos por la clasificación internacional del funcionamiento, con base en los cuales se determinará a qué grupos se puede ayudar con la aplicación.
2. Segunda etapa: determinar qué algoritmo se va a usar. En esta etapa se describirá el algoritmo utilizado para la detección de movimiento.
3. Tercera etapa: desarrollo del recorrido virtual. En esta etapa se elegirá el escenario para el recorrido, así como el software para su desarrollo.
4. Cuarta etapa: unificación del recorrido virtual con el algoritmo de detección de movimiento. Esta es la última etapa, en la que se completará la aplicación y se realizarán las respectivas pruebas.

Para implementar un algoritmo de detección de movimiento o *tracking* sobre una aplicación de realidad virtual es importante definir algunas consideraciones: ¿Para quién está dirigida esta aplicación? ¿Qué algoritmo se va a utilizar? ¿En qué consiste la aplicación de realidad virtual? ¿De qué manera interactúa el algoritmo con la aplicación?

Estas preguntas permiten obtener información y definir las bases sobre las cuales se puede desarrollar la aplicación. A continuación, se responde cada una de estas preguntas.

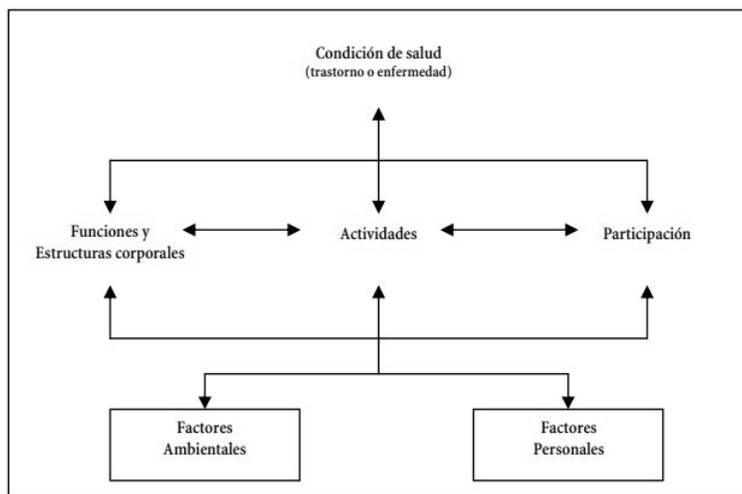
## 2.1 ¿Para quién está dirigida esta aplicación?

La aplicación está orientada a personas con cierto grado de discapacidad, esta sentencia puede resultar algo general o incluso ambigua, de modo que no resulta nada trivial definir e indagar a fondo sobre distintos grados de discapacidad y condiciones, con el fin de delimitar el público objetivo de una mejor manera; la CIF (clasificación internacional del funcionamiento, de la discapacidad y de la salud) aborda esta problemática explicando que, el funcionamiento de un individuo en un dominio específico se entiende como una relación compleja entre su condición de salud y el contexto de su entorno, es decir, factores ambientales y personales (Organización Mundial de la Salud, 2001). Esto ya da un indicio de que la discapacidad altera esta relación persona-ambiente; sin embargo, sigue siendo bastante amplia y se debe delimitar.

Para delimitar el grupo en la *Figura 1* se muestra lo implementado, lo cual permite apreciar cómo las actividades de una persona se ven afectadas por diferentes elementos.

**Figura 1**

*Interacciones Entre los Componentes de la CIF*



Entonces, para definir el público objetivo de esta aplicación se dejan de lado los factores ambientales (p.ej., las actividades de la sociedad, las características arquitectónicas, el sistema legislativo), ya que son extrínsecos a la persona, y se toman en cuenta factores personales como la edad, estilos de vida, antecedentes sociales, patrón global de conducta y tipo de personalidad.

Pero aún queda pendiente la condición de salud, las funciones y estructuras corporales de la persona; para esto se hace uso de la clasificación dada por la CIF. En primer lugar, no debe existir un grado de afectación en las funciones corporales, pero sí se va a enfocar en personas que posean algún grado de afectación en estructuras relacionadas con el movimiento, en particular estructura de la extremidad inferior, tronco o extremidad superior, estructuras musculoesqueléticas adicionales, relacionadas con el movimiento y problemas con la movilidad. Si bien esta es una aproximación general al tema, ayuda a definir qué personas pueden beneficiarse de esta aplicación.

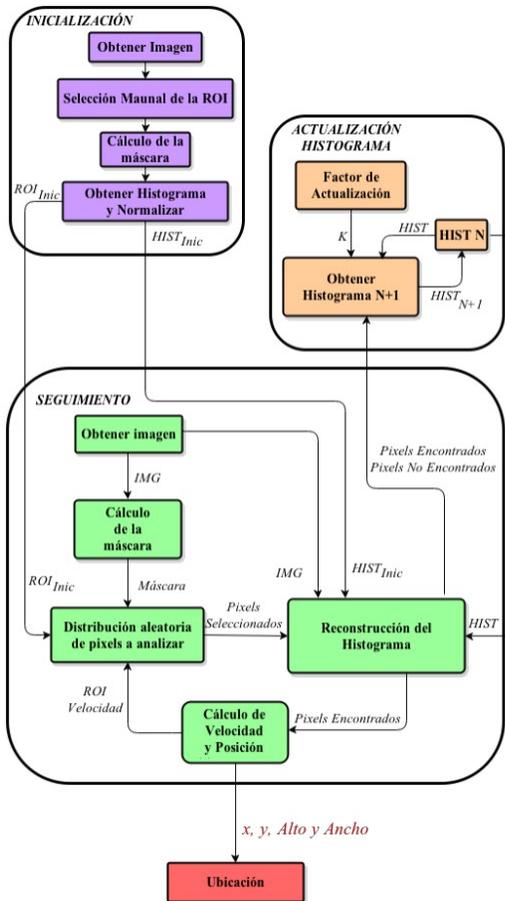
## 2.2 ¿Qué algoritmo se va a utilizar?

Este apartado en particular es muy importante; por un lado, permite establecer la relación que tiene el usuario con la aplicación, es decir, dependiendo de si se elija un algoritmo u otro, hará que esta interacción sea más fluida o por el contrario más torpe; además, el seguimiento de objetos es una tarea relevante dentro del campo de la visión por computador y con el desarrollo de nuevas tecnologías, disponibilidad de cámaras de video de alta calidad y bajo costo, y la creciente necesidad de analizar un video de forma automatizada se ha conseguido un avance muy grande en este campo (Prieto et al., 2014). Por lo tanto, se disponen de varias opciones que varían en su dificultad de implementación o incluso el tamaño necesario.

La estructura de estos algoritmos suele ser en bloques divididos en detección, seguimiento y actualización, en la *Figura 2* se grafica este apartado.

**Figura 2**

*Esquema del Algoritmo de Seguimiento*



Para detectar qué movimientos realiza el usuario se propone una cámara apuntando hacia él, es decir, recibe un *input* con el video de lo que está haciendo; el video recibido es separado en imágenes, las cuales reciben un tratamiento (procesamiento de imágenes) mediante la librería *opencv*. En primer lugar se restan las imágenes contiguas, lo que permite resaltar el objeto en movimiento; luego se aplica una dilatación a la imagen para facilitar los cálculos, y, por último se procesa esta diferencia entre las imágenes para obtener el movimiento realizado (Kalal et al., 2011). Este proceso se puede apreciar en la *Figura 3*.

**Figura 3**

*Detección de Movimiento con Algoritmo Opencv*



Es importante aclarar que este algoritmo funciona con una cámara estática, la cual se colocará frente al usuario donde permita captar sus movimientos. Según Luo et al. (2021) este algoritmo se divide en tres etapas:

5. Inicialización: se adquiere la imagen inicial y se le da un tratamiento.
6. Seguimiento: se analizan las imágenes procesadas para determinar una posición.
7. Actualización: se genera un bucle para comparar las imágenes generadas por el video y determinar el movimiento realizado.

### 2.3. ¿En qué consiste la aplicación de realidad virtual?

Para responder esta pregunta es necesario entender a qué se refiere el término de realidad virtual, si bien su nombre ya es bastante descriptivo, puede asociarse simplemente a una simulación digital con aspectos más superficiales y espectaculares (Levis, 1997), pero se puede ir un poco más allá y decir que es una tecnología que permite la creación de espacios tridimensionales por medio de un ordenador; es decir, permite la simulación de la realidad, pero con una ventaja aparente, se puede introducir en el ambiente virtual elementos y eventos considerados útiles (Botella et al., 2006), de acuerdo con el objetivo establecido.

Esta inclusión de eventos y elementos representa los cimientos de la realidad virtual para generar una experiencia sintética haciendo que el usuario sustituya la realidad física por un entorno ficticio (Pérez, 2011).

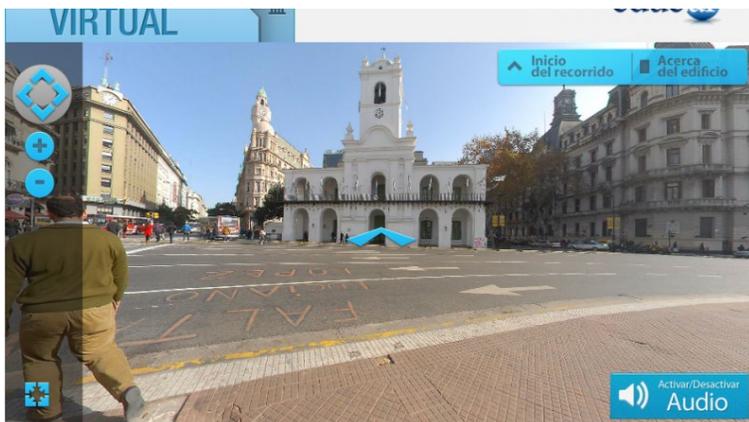
Llegados a este punto se comprende que esta técnica genera una experiencia única, al crear la ilusión de estar en el espacio generado por ordenador (Botella, et al., 2006).

Sin embargo, esta relación hombre-máquina cobra un mayor significado al proyectarse a personas con cierto grado de discapacidad, dado que en su diario vivir existe una interacción dinámica entre la persona y su medio, es decir, no siempre se dan en una relación recíproca predecible y no siempre van a ser iguales (Organización Mundial de la Salud, 2001).

Aquí es donde entra esta aplicación, la cual busca que la interacción entre el usuario y el medio ambiente virtual esté cargada de este dinamismo que describe la realidad; por lo cual se propone un recorrido virtual interactivo en el cual, si bien se implementan movimientos automáticos debidos a las limitaciones del usuario, se espera generar movimientos individuales que busquen interactuar con el medio virtual; es decir, la aplicación permite al usuario moverse a su gusto e interactuar con el medio virtual.

#### Figura 4

*Ejemplo de Ambiente Virtual*



#### 2.4. ¿De qué manera interactúa el algoritmo con la aplicación?

Una vez definido el público hacia quien va dirigido la aplicación, el algoritmo que se va a usar y el medio en el cual se presenta la aplicación, es necesario entender cómo se fusiona el reconocimiento del movimiento y la realidad virtual, además de que beneficios aporta al usuario.

Los resultados obtenidos con el algoritmo de reconocimiento de movimiento se traducen a un sistema de ejes, el cual dirá la dirección del movimiento y de acuerdo con el tiempo con el que esta varía su velocidad; esta información se lleva al módulo de realidad virtual, el cual lo interpreta de acuerdo con lo establecido.

Entonces, esta combinación entre una aplicación que simula la realidad con un algoritmo que capta los movimientos y los traduce a una interacción más fluida, busca ayudar al usuario, ya que las interacciones con el medio funcionan en dos direcciones; la presencia de la discapacidad puede incluso modificar a la propia condición de salud (Organización Mundial de la Salud, 2001), y este es un punto crítico, ya que, si de alguna manera se consigue que la interacción con la aplicación reduzca esta limitación en la interacción puede incluso ayudar a su salud.

La idea general no es nueva, ya que existen otros casos en los que se ha usado la realidad virtual para ayudar con diferentes enfermedades. Según Botella et al. (2006) a continuación se nombran algunos de ellos.

- Realidad virtual en el tratamiento de los trastornos de ansiedad (fobias específicas, fobia social, trastorno de estrés postraumático, trastorno de pánico con agorafobia).
- Realidad virtual en el tratamiento de los trastornos de la conducta alimentaria.
- Realidad virtual en el tratamiento de otros trastornos mentales.

Entonces, esta aplicación propone un nuevo enfoque, orientado en la interacción del usuario con el medio, interacción que busca, no precisamente reemplazar a la real, sino ayudar a quienes han perdido la capacidad de interactuar con partes de su medio físico.

## Resultados

Una vez definida la metodología se pueden comenzar a establecer las opciones óptimas para la aplicación basadas en los resultados expuestos en el apartado anterior. Es decir, se va a dar una respuesta puntual a las preguntas previas. En este artículo se muestran los resultados únicamente de la primera y segunda etapa.

### 3.1. ¿Para quién está dirigida esta aplicación?

Inicialmente hay que referirse a afectaciones en las funciones corporales, tales como estructuras del sistema nervioso, estructuras involucradas en la voz y el habla o sistemas cardiovasculares y respiratorio; en principio, las personas a quienes va orientada la aplicación no deben presentar afectaciones en estas funciones, pues se busca una retroalimentación por parte del usuario, lo cual permitirá hacer cambios que le beneficien.

Entonces, el público hacia quien va la aplicación es para personas que tengan un grado de afectación en su movilidad, es decir, que hayan perdido la capacidad de interactuar total o parcialmente con su entorno; esto permite que la aplicación funcione de manera óptima y genere en el usuario una expectativa para su interacción.

Por último, se deben analizar los patrones de comportamiento y personalidad, puesto que se buscan personas que no reaccionen de manera agresiva; esto con el fin de calibrar el software y quizá luego prestar servicios a un público mayor.

### 3.2. ¿Qué algoritmo se va a utilizar?

Para detectar el movimiento de los objetos se ha optado por utilizar dos algoritmos que realizan funciones específicas, pero trabajan en conjunto. A continuación, se describe de manera general el proceso que realizan estos algoritmos:



- En primer lugar se utiliza un algoritmo para la detección de movimiento, el cual se basa en una variación del algoritmo *K-nearest neighbours* para la extracción del fondo; esto permite crear una imagen de referencia y luego extraer información con cada nuevo frame de video (KaewTraKulPong & Bowden, 2002). De manera general su estructura es la siguiente:
  - Conversión a escala de grises y eliminación de ruido.
  - Operación de sustracción entre planos sucesivos.
  - Aplicar un umbral a la resta de imágenes contiguas.
  - Detección de contornos.
- La segunda parte se basa en un algoritmo de *object-tracking* que permita, con el resultado obtenido en el proceso anterior, cuantificar el movimiento realizado. Su estructura es la siguiente:
  - Obtener la variación en la posición de los píxeles mediante una función de flujo.
  - Generar un sistema de referencia que permita convertir las variaciones de la imagen en magnitudes medibles.
  - Comparar la variación de movimiento y determinar un valor apropiado.

Sobre la base de estos algoritmos se procede a dividir el proceso en tres tareas específicas, con el fin de tener una estructura modular que facilite posibles actualizaciones. A continuación, se detallan estas tareas:

#### a. Detección de objetos

Para detectar los objetos en cámara se procede de la siguiente manera:

- Iniciar la cámara con el método *VideoCapture*.
- Realizar un tratamiento a la imagen obtenida.
- Generar un filtro con base en la resta de imágenes; esto mediante el método *subtract*. De esta manera el fondo o los elementos que se mantengan fijos serán descartados y se tomará en cuenta únicamente a la persona.

#### b. Procesamiento de imágenes

Esta etapa es importante, pues consiste en el tratamiento que se les da a las imágenes obtenidas por la cámara, y de esto dependerá el qué tan fácil es obtener información para detectar el movimiento.

- Se convierte la imagen en blanco y negro para trabajar solo con un canal, método *cvtColor*.
- Después de realizar la resta de imágenes se dilata el resultado para incrementar la precisión, método *cv2.dilate*.

### c. Detección de movimiento

Por último, se determina una medida de movimiento, este valor será devuelto para indicar una magnitud y dirección del movimiento del usuario. Aquí se incluye la librería *numpy* para realizar los cálculos necesarios.

- Utilizar el método *calcOpticalFlowFarneback*, el cual genera una función de flujo de acuerdo con el movimiento obtenido por la cámara.
- Con base en la función de flujo dibujar las aristas con una matriz.
- Comparar las variaciones de movimiento y determinar los valores apropiados de acuerdo con una sensibilidad específica.
- Devolver un valor numérico que indica cuanto se ha movido el usuario.

Con la estructura ya definida se procede a estructurar una clase que mida las variaciones en el movimiento grabado por la cámara. A continuación, en la *Figura 5* se muestra una fracción del código fuente implementado, y en la 6 y 7 se evidencia parte de los resultados.

## Figura 5

### Código Fuente

```

1. """! @brief La clase movimiento controla los parámetros recibidos por la cámara"""
2.
3. ##
4. # @file movimiento.py
5. #
6. # @brief Define la clase Movimiento.
7. #
8. # @section Descripción
9. # Clase encargada de medir las variaciones en el movimiento grabado por la cámara mediante
  OpenCV.
10. # - Movimiento (clase)
11. #
12. # @section librerías/módulos
13. # - OpenCV (https://opencv.org/)
14. # - Acceso a las funciones para el control y procesamiento de imágenes.
15. # - Numpy (https://numpy.org/)
16. # - Acceso a las funciones para el cálculo de las funciones trigonométricas.
17. #
18. # @section Notas
19. # - Permite al usuario interactuar con la raqueta mediante el movimiento de las manos.
20.
21. #import
22. import cv2
23. import numpy as np
24. import argparse
25.
26. # Clases
27. # -----
28. class Movimiento():
29.     """! Clase que mide el movimiento del usuario y lo transforma en una variable
  medible."""
30.
31.     def __init__(self):
32.         """! Se enciende la cámara y se inicializan variables."""
33.
34.         self.webcam = cv2.VideoCapture(0)
35.         # webcam.set(cv2.CAP_PROP_SETTINGS, 1)
36.         self.check, self.frame = self.webcam.read()
37.         ## Variables auxiliares
38.         self.im1 = self.frame
39.         self.im2 = self.frame
40.         ## Se convierte la primera imagen a blanco y negro
41.         self.prevgray = cv2.cvtColor(self.frame, cv2.COLOR_BGR2GRAY)
42.         self.prevgray1 = cv2.cvtColor(self.frame, cv2.COLOR_BGR2GRAY)
43.         ## Variables de ancho y alto
44.         width = int(self.webcam.get(cv2.CAP_PROP_FRAME_WIDTH))
45.         height = int(self.webcam.get(cv2.CAP_PROP_FRAME_HEIGHT))
46.         ## Variables auxiliares
47.         self.bgGray=0
48.         self.i = 0
49.         self.A = 0
50.         self.B = 0
51.         self.C = 0
52.
53.     def deteccion(self):
54.         """! Clase que detecta el movimiento y retorna un valor, que determina cuánto se ha
  movido el usuario"""
55.
56.         ##Se almacenan la información de la cámara.
57.         ret, frame = self.webcam.read()
58.         if ret == False: return
59.         gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
60.
61.         ## Se crea un nuevo filtro, en base a la resta de imágenes
62.         self.im2 = frame
63.         resta = cv2.subtract(self.im2, self.im1)
64.         #cv2.imshow('resta', resta)
65.         self.im1 = frame
66.         # fgMask = backSub.apply(resta)
67.         # cv2.imshow('FG Mask', fgMask)
68.         rest = cv2.cvtColor(resta, cv2.COLOR_BGR2GRAY)
69.
70.         ## Se usa un proceso de dilatación para incrementar la precisión
71.         kernel = np.ones((3, 3), np.uint8)
72.         rest = cv2.dilate(rest, kernel, iterations=3)
73.
74.         #cv2.imshow('resta', rest)
75.         # rest = cv2.Canny(rest, 100, 200)
76.         # cv2.imshow("Detected Edges", rest)
77.
78.         ##Se obtiene una función de flujo en base al movimiento de la mano
79.         flow = cv2.calcOpticalFlowFarneback(self.prevgray1, rest, None, 0.2, 4, 18, 1, 3, 1.1, 0)
80.         #flow = cv2.calcOpticalFlowFarneback(self.prevgray1, rest, None, 0.5, 3, 15, 3, 5, 1.2, 0)
81.         self.prevgray1 = rest
82.
83.         ## Se llama a la función draw flow para obtener un valor de movimiento
84.         aux1, sumal = self.draw_flow(rest, flow, self.A)
85.         aux1 = cv2.cvtColor(aux1, cv2.COLOR_BGR2GRAY)

```

**Figura 6**

*Extracción de Fondo*



**Figura 7**

*Función de Flujo Aplicada Sobre el Video*



## Conclusiones

Se ha establecido un público objetivo, fundamentado en el contenido de la CIF, que ha sido delimitado de acuerdo con las capacidades físicas y mentales de la persona; esto con el fin de que la interacción con el software sea lo más óptima y se pueda obtener retroalimentación por parte del usuario.

Podemos ver que la separación de todo el proceso para detectar el movimiento en dos algoritmos principales y luego, en tres tareas concretas, ha permitido crear módulos específicos que realizan una tarea puntual, lo cual optimiza el software y el flujo de trabajo.

Se concluye entonces que el algoritmo implementado ha funcionado correctamente, permitiendo detectar el movimiento del usuario y transformar este resultado en un valor numérico. El resultado de este proceso permitirá la movilidad dentro del ambiente virtual. Con este proceso finalizado se puede comenzar con el desarrollo de las siguientes etapas.

## Referencias

- Alban, G. (2018). *Sistema domótico de apoyo para personas con discapacidad motriz mediante tecnología móvil y reconocimiento de voz*. [Tesis de grado, Universidad Técnica de Ambato] <http://repositorio.uta.edu.ec/jspui/handle/123456789/28012>
- Botella, C., García-Palacios, A., Baños, R., Quero, S. y Bretón-López, J. (2006). Realidad Virtual y Tratamientos Psicológicos. *Psicología Conductual*, 14(3), 2006, 491-509 [https://www.behavioralpsycho.com/wp-content/uploads/2020/04/09.Botella\\_14-3oa.pdf](https://www.behavioralpsycho.com/wp-content/uploads/2020/04/09.Botella_14-3oa.pdf)
- Bravo, I. (2007). *Arquitectura basada en FPGAs para la detección de objetos en movimiento, utilizando visión computacional y técnicas* [Tesis doctoral, Universidad de Alcalá] <https://ebuah.uah.es/dspace/handle/10017/1381>
- Castillo, C. (2014). *Sistema de detección automático de movimientos corporales mediante técnicas de procesamiento digital de imágenes* [Tesis de grado, Universidad Nacional de Piura] <http://repositorio.unp.edu.pe/handle/UNP/801>
- Gómez-Vega, C., Ramírez-Medina, V., Méndez-García, M., Alba, A. y Salgado-Delgado, R. (2015). Rastreo de movimiento de roedores usando visión computacional. *Memorias Del Congreso Nacional De Ingeniería Biomédica*, 2(1), 196–199. <https://memoriascnib.mx/index.php/memorias/article/view/92>
- González, S., Tello, J., Silva, P., Lüders, C., Butelmann, S., Fristch, R., Solar, F., Rigo-righi, C., y David, P. (2012). Calidad de vida en pacientes con discapacidad motora según factores sociodemográficos y salud mental. *Revista Chilena de Neuro-Psiquiatría*, 50(1), 23–34. <http://dx.doi.org/10.4067/S0717-92272012000100003>
- KaewTraKulPong, P., & Bowden, R. (2002). An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection. En Remagnino, P., Jones, G., Paragios, N., & Regazzoni, C (Eds) *Video-Based Surveillance Systems* (135–144) Springer. [https://doi.org/10.1007/978-1-4615-0913-4\\_11](https://doi.org/10.1007/978-1-4615-0913-4_11)
- Kalal, Z., Mikolajczyk, K., & Matas, J. (2011). Tracking-learning-detection. En *IEEE transactions on pattern analysis and machine intelligence*, 34(7), 1409-1422. <https://doi.org/10.1109/TPAMI.2011.239>
- Levis, D. (2006). ¿Qué es la realidad virtual? *Realidad Virtual*, 1–28. <https://xdoc.mx/preview/3-la-tecnologia-que-hace-posible-los-simulacros-60669920b139b>
- Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W. & Kim, T. (2021). Multiple object tracking: A literature review. *Artificial Intelligence*, 293. <https://doi.org/10.1016/j.artint.2020.103448>
- Organización Mundial de la Salud. (2001). *Clasificación Internacional del Funcionamiento de la discapacidad y la salud*. 1–1189. Ministerio de Trabajo y Asuntos Sociales. Secretaría de Estado de Servicios Sociales,

Familias y Discapacidad. Instituto de Mayores y Servicios Sociales (IMSERSO).<https://apps.who.int/iris/handle/10665/43360>

Pérez, F. (2011). Presente y Futuro de la Tecnología de la Realidad Virtual. *Creatividad y Sociedad*, 16. <http://creatividadysociedad.com/creatividad-tic-y-sociedad-de-la-informacion>

Prieto, M., Marufo, M., Di Matteo, L., Verrastro, R., Hernández, A., Gómez, J., y Verrastro, C. (2014). Algoritmo de seguimiento de objetos en imágenes mediante reconstrucción iterativa de histograma en tiempo real. En *Jornadas Argentinas de Robótica*, 155. <https://doi.org/10.13140/2.1.4337.0569>

Copyright (2022) © Jonathan Marcelo Tipán Espinoza



Este texto está protegido bajo una licencia internacional [Creative Commons](#) 4.0.

Usted es libre para Compartir—copiar y redistribuir el material en cualquier medio o formato — y Adaptar el documento — remezclar, transformar y crear a partir del material—para cualquier propósito, incluso para fines comerciales, siempre que cumpla las condiciones de Atribución. Usted debe dar crédito a la obra original de manera adecuada, proporcionar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que tiene el apoyo del licenciante o lo recibe por el uso que hace de la obra.

[Resumen de licencia](#) – [Texto completo de la licencia](#)