Tecnológicas

ISSN-p: 0123-7799
ISSN-e: 2256-5337

Vol. 24, nro. 50, e1265

Recibido: 30 julio 2020
Aceptado: 09 octubre 2020
Disponible: 13 noviembre 2020

Interfaz humano-computador basada en gestos faciales y orientada a la aplicación WhatsApp para personas con limitación motriz de miembros superiores

Human-Computer Interface Based on Facial Gestures Oriented to WhatsApp for Persons with Upper-Limb Motor Impairments

Carlos Ferrin-Bolaños[1];
José Mosquera-DeLaCruz[2];
John Pino-Murcia[3];
Luis Moctezuma-Ruiz[4];
Jonathan Burgos-Martínez[5];
Luis Aragón-Valencia[6];
Humberto Loaiza-Correa[7]

[1]Universitaria Católica Lumen Gentium, Cali-Colombia, cdferrinb@unicatolica.edu.co
[2]Universitaria Católica Lumen Gentium, Cali-Colombia, jhmosquerad@unicatolica.edu.co
[3]Universitaria Católica Lumen Gentium, Cali-Colombia, john.pino01@unicatolica.edu.co
[4]Universitaria Católica Lumen Gentium, Cali-Colombia, luis.moctezuma01@unicatolica.edu.co
[5]Universitaria Católica Lumen Gentium, Cali-Colombia, jonthan.burgos01@unicatolica.edu.co
[6]Universidad del Cauca, Cali-Colombia, fernandoaragonva@gmail.com
[7]Universidad del Valle, Cali-Colombia, humberto.loaiza@correounivalle.edu.co

Cómo citar / How to cite

C. Ferrin-Bolaños; J. Mosquera-DeLaCruz; J. Pino-Murcia; L. Moctezuma-Ruiz; J. Burgos-Martínez; L. Aragón-Valencia; H. Loaiza-Correa, "Interfaz humano-computador basada en gestos faciales y orientada a la aplicación WhatsApp para personas con limitación motriz de miembros superiores", *TecnoLógicas*, vol. 24, nro. 50, e1722, 2021. https://doi.org/10.22430/22565337.1722

Resumen

En el caso de personas con limitación motriz de miembros superiores, los gestos faciales son la principal forma de comunicarse con el mundo. Sin embargo, las interfaces actuales basadas en gestos no tienen en cuenta la reducción de movilidad que la mayoría de las personas con limitación motriz experimentan durante sus periodos de recuperación. Como alternativa para superar esta limitación, se presenta una interfaz humana-computador basada en técnicas de visión por computador sobre dos tipos de imagen: la imagen del rostro capturada mediante webcam y la captura de pantalla de una aplicación de escritorio en primer plano. La primera imagen es utilizada para detectar, seguir y estimar la pose del rostro con el fin de desplazar y ejecutar comandos con el cursor; la segunda imagen es utilizada para lograr que los desplazamientos del cursor sean realizados a zonas específicas de interacción de la aplicación de escritorio. La interfaz es programada totalmente en Python 3.6 utilizando bibliotecas de código abierto y se ejecuta en segundo plano dentro del sistema operativo Windows. El desempeño de la interfaz se evalúa con videos de personas utilizando cuatro comandos de interacción con la aplicación WhatsApp versión de escritorio. Se encontró que la interfaz puede operar con varios tipos de iluminación, fondos, distancias a la cámara, posturas y velocidades de movimiento; la ubicación y el tamaño de la ventana de WhatsApp no afecta la efectividad de la interfaz. La interfaz opera a una velocidad de 1 Hz y utiliza el 35 % de la capacidad de un procesador Intel Core i5 y 1,5 GB de RAM para su ejecución lo que permite concebir esta solución en equipos de cómputo personales.

Palabras Claves

Interfaz humano-computador, detección de rostros, visión por computador, tecnología de asistencia.

Abstract

People with reduced upper-limb mobility depend mainly on facial gestures to communicate with the world; nonetheless, current facial gesture-based interfaces do not take into account the reduction in mobility that most people with motor limitations experience during recovery periods. This study presents an alternative to overcome this limitation, a human-computer interface based on computer vision techniques over two types of images: images of the user's face captured by a webcam and screenshots of a desktop application running on the foreground. The first type is used to detect, track, and estimate gestures, facial patterns in order to move and execute commands with the cursor, while the second one is used to ensure that the cursor moves to specific interaction areas of the desktop application. The interface was fully programmed in Python 3.6 using open source libraries and runs in the background in Windows operating systems. The performance of the interface was evaluated with videos of people using four interaction commands in WhatsApp Desktop. We conclude that the interface can operate with various types of lighting, backgrounds, camera distances, body postures, and movement speeds; and the location and size of the WhatsApp window does not affect its effectiveness. The interface operates at a speed of 1 Hz and uses 35 % of the capacity a desktop computer with an Intel Core i5 processor and 1.5 GB of RAM for its execution; therefore, this solution can be implemented in ordinary, low-end personal computers.

Keywords

Human-computer interface, face detection, computer vision, assistive technology.

1. INTRODUCCIÓN

En la actualidad, el dominio de computadoras se ha vuelto una de las necesidades apremiantes de nuestra sociedad moderna, permitiendo no solo para acceder a información relevante, sino también como una herramienta comunicativa y de socialización [1]. Es decir, poder manejar dispositivos computacionales permite interactuar con el resto de la humanidad. Sin embargo, toda esta revolución informática presenta barreras limitantes que interfieren en el acceso directo a este mundo moderno, un ejemplo de ello lo constituyen la población con alteraciones motoras, especialmente las que presentan limitación motriz de miembros superiores [2]. En Colombia, aproximadamente un 2,6 % de la población padece algún tipo de limitación según [3]. El diseño de Interfaces Humano-Computador (IHC) para facilitar la interacción de este tipo de personas con aplicaciones orientadas a internet constituye todavía un problema abierto dentro de la comunidad científica [4]. Dentro de las tipologías de IHC más investigadas se encuentran las basadas en gestos, exactamente las basadas en gestos faciales, ya que los miembros superiores no son un mecanismo idóneo en este tipo de personas para la generación de gestos.

Se han identificado varios trabajos en la literatura científica que se constituyen en trabajos referentes al aquí desarrollado. En Colombia, por ejemplo, en 2005, en la Universidad Javeriana se reportó el desarrollo de un sistema electrónico capaz de reconocer, en tiempo real, doce gestos realizados por un interlocutor en una escena con iluminación y con fondo controlados. Aun cuando el sistema constituye un gran aporte a la línea aquí investigada, el sistema se limita únicamente a gestos de mano. La interfaz es robusta a rotaciones, translaciones y cambios de escala de la mano del interlocutor en el plano de la cámara, estas características son importantes para tener en cuenta en el caso de utilizar gestos faciales [5].

En 2009, en la Universidad Nacional Sede Manizales, se trabajó una interfaz hombre máquina que permitía ofrecer una "mano adicional" para controlar la laparoscopia a un cirujano cuando se encuentra desarrollando una intervención quirúrgica de este tipo, lo que a menudo se hace muy necesario ya que la mayor parte del tiempo los cirujanos tienen ambas manos e incluso ambos pies ocupados manipulando instrumentos quirúrgicos. La interfaz utilizaba gestos del rostro, específicamente de la postura de los labios para generar el comando. Se demostró que un cirujano podía, de manera fácil y precisa, controlar el brazo de un robot haciendo gestos faciales sin tener que usar interruptores o comandos de voz para iniciar la secuencia de control. Este tipo de ideas constituye un gran aporte si se piensa en pacientes con capacidad de producir gestos faciales, pero con limitaciones de miembro superior [6]. En 2017, en la Universidad del Valle se desarrolló una interfaz audiovisual (audio y voz) para comandar varias aplicaciones orientadas a la Web (Google, Facebook y Gmail). En la parte gestual utilizaron algoritmos convencionales de detección de rostros para la manipulación del cursor y utilizaron la detección de guiño para la emulación de clic derecho.

La propuesta se caracteriza por estar centrada en el usuario y ser independiente de la aplicación, obteniéndose resultados sobresalientes, principalmente con personas sin limitaciones motrices de miembros superiores [7].

Por otra parte, en el contexto internacional se encuentran varios desarrollos que en la actualidad son productos comerciales consolidados en la industria de los dispositivos de asistencia tecnológica. Por ejemplo, *Facial Mouse* es un sistema emulador de ratón basado en el movimiento facial del usuario. Se coloca una cámara web frente al usuario, enfocándose en la cara del mismo. Luego, se utiliza un algoritmo de extracción de movimiento que es independiente del usuario, para extraer el movimiento facial del video. Este movimiento se usa para mover el puntero del mouse que se controla de una manera relativamente similar a los dispositivos de mouse estándar [8]. *FaceMouse* es otro sistema que utiliza una cámara web

estándar y técnicas de visión por computadora para rastrear la nariz de la persona y usar esto para mover el puntero del mouse (de acuerdo con la dirección del movimiento de la nariz) [9].

Estudiantes del Departamento de Matemáticas e Informática de la Universidad de Las Islas Baleares, en el año 2008, desarrollaron un sistema que permite a las personas con discapacidad motriz acceder a la computadora a través de los movimientos de la cabeza del usuario. El sistema no requiere calibración y detecta automáticamente la cara usando el algoritmo Viola y Jones. A continuación, el método divide la cara en regiones: ojos, cejas, boca y nariz. Se utiliza también un método Gaussiano 3D para detectar la región del color de la piel. Para determinar las regiones de los ojos y las cejas se realiza el umbral de imagen.

El único gesto facial a tener en cuenta es el parpadeo. El movimiento del mouse se realiza mediante la posición de la nariz, y el parpadeo del ojo puede tener diferentes funciones.

De esta manera, las personas sin movimiento en los miembros superiores pueden controlar la computadora [10]. Uno de los sistemas de IHC para personas con tetraplejia que más ha crecido en la industria de tecnologías de asistencia es *Camera Mouse*, un software comercial que rastrea los movimientos del usuario con una cámara de video y los traduce a los movimientos del puntero del mouse en la pantalla. Se puede simular un clic izquierdo del mouse al pasar el puntero sobre el icono que se va a seleccionar [11]. Nose Tracking [12], propuesto en 2017, demostró, mediante un experimento controlado, que se desempeña mejor que Camera Mouse; sin embargo, el tiempo promedio para hacer clic es mejor en Camera Mouse que con Nose Tracking.

La mayoría de los trabajos encontrados en la revisión de la literatura hasta ahora asumen que el usuario puede hacer movimientos, aunque sean pequeños, con la cabeza, la mano o los ojos y así controlar la computadora. Desafortunadamente, en algunos casos de limitaciones severas, no hay movilidad de los miembros superiores y en el mejor de los casos se puede contar con movimientos leves del rostro e inclusive solo ciertos músculos faciales pueden moverse, lo que lleva a que la mayoría de los sistemas de interfaz existentes no sean lo suficientemente robustos para que un usuario con limitación motriz de miembros superiores pueda usar para tareas básicas de interacción, como la selección de una zona especial dentro de una interfaz gráfica de usuario. En casos de restricciones extremas de movimiento, las expresiones y movimientos leves faciales son la única alternativa para interactuar con la computadora [13].

Por lo anteriormente expuesto, en este trabajo se describe el diseño de una interfaz humano-computador que permita a personas con limitación de miembros superiores realizar tareas básicas en una aplicación orientada a internet mediante gestos faciales. La hipótesis de nuestro trabajo se centra en que es posible facilitar la interacción con el computador a las personas con limitación motriz de miembros superiores si, además de los gestos faciales, se identifican zonas de interacción de la aplicación con la que se interactúa, permitiendo un posicionamiento más rápido del cursor.

2. REVISIÓN DE LA LITERATURA

El desarrollo de dispositivos de asistencia tecnológicas basados en sistemas de visión por computador aumenta día a día. Estos dispositivos de asistencia se pueden utilizar para que las personas con discapacidades físicas interactúen con muchas aplicaciones, tales como la comunicación (teléfonos inteligentes, tabletas, portátiles), el control del entorno en el hogar (control inteligente de electrodomésticos), la educación y el entretenimiento (videojuegos).

En particular, para diseñar un dispositivo de asistencia tecnológica basado en gestos faciales, es fundamental revisar los trabajos de investigación relacionados con los métodos de

detección de rostros. En los últimos años, los sistemas de visión por computador para detección de rostros se han orientado más hacia aplicaciones como sistemas biométricos, de seguridad, etc. Los sistemas de detección de rostros se están probando e instalando en aeropuertos para proporcionar un nuevo nivel de seguridad [14]. De hecho las IHC, basadas en la expresión facial y los gestos corporales, se están explotando como una forma de reemplazar las interfaces tradicionales como el mouse y el teclado en los computadores [1].

Las expresiones de la cara desempeñan un papel importante y complementario o suplementario al que desempeñan por ejemplo las manos en los actuales sistemas de IHC [15].

La obtención de patrones faciales a partir de imágenes permite detectar y localizar rostros automáticamente en escenas complejas con fondos no controlados [16]. Las limitaciones por las rotaciones fuera del plano de imagen se pueden abordar utilizando diferentes tipos de técnicas de registro rígido y no rígido de imágenes [17]. En este enfoque, las posiciones centrales de los patrones faciales distintivos como los ojos, la nariz y la boca se consideran puntos de referencia para normalizar las caras de prueba de acuerdo con algunos modelos faciales genéricos o de referencia [18]. Los cambios de escala de los rostros pueden abordarse escaneando imágenes en varias resoluciones para determinar el tamaño de los rostros presentes, que luego pueden normalizarse lo suficiente para permitir la detección exitosa de los mismos [19].

La detección de rostros [20], [21] es un caso específico de la detección de objetos. La detección de caras por computador es un proceso por el cual el ordenador ubica los rostros presentes en una imagen o un vídeo. Generalmente, la imagen se procesa en escala de grises y el resultado del algoritmo es similar al mostrado en la Figura 1.

Figura 1. Proceso de detección de rostro a partir de imagen en escala de grises proveniente de una imagen RGB. Fuente: elaboración propia.

Este proceso no es tan sencillo como lo haría el sistema visual humano. Según las condiciones en la que se encuentre la imagen durante el proceso de detección puede suponer algunos problemas. En muchos casos la luminosidad no es la adecuada, aparecen elementos extraños, las caras están de perfil, se encuentran tapadas por algún elemento o por alguna otra cara o en un ángulo complicado. Actualmente existen varios métodos para detección de rostros.

Métodos basados en el conocimiento: estos métodos representan las técnicas de detección de rostros que se basan en una serie de reglas previas definidas por la persona que quiere hacer la detección. Se definen una serie de características sobre las caras a detectar (forma de la cabeza, dos ojos, una nariz). Esto puede suponer un problema, y es que, si estas reglas son muy generales, el resultado de una búsqueda en imágenes donde no hay rostros, seguramente el resultado dirá que sí hay rostros y, además, en una cantidad elevada. En el caso en que las reglas establecidas sean muy específicas posiblemente también aparezcan problemas ya que el resultado de la detección será muy bajo [4].

Métodos basados en características invariantes: estos métodos utilizan como punto de referencia el color de la piel y la textura. El problema radica en que, si en la imagen aparece ruido o diferentes condiciones de iluminación, el algoritmo aplicado no funcionará correctamente. Si se utiliza el color de la piel, los algoritmos que utilizan toda la gama de colores tienen mejor resultado que los que utilizan una escala de grises [22], [23].

Métodos basados en plantillas: estos métodos modelan geométricamente la forma del rostro. Una vez están definidas las plantillas (círculo, elipse, etc.) se evalúa la correspondencia entre la cara y la plantilla. Las principales técnicas son las plantillas deformables y los contornos activos [24].

Métodos basados en apariencia: esta técnica en un principio no necesita el conocimiento de las características de la cara de la imagen que se quiere detectar. En los algoritmos utilizados en estos métodos aparecen los conceptos de entrenamiento y de aprendizaje, diferentes métodos para poder realizar la detección de caras por ordenador [25], [26].

Uno de los algoritmos más populares en la actualidad para la detección de rostros es el algoritmo de Viola Jones [27]. Este algoritmo que está enmarcado dentro de los métodos basados en apariencia tiene un coste computacional muy bajo, y consta de dos partes principales: clasificador en cascada, que garantiza una discriminación rápida y un entrenador de clasificadores basado en Adaboost [28]. El algoritmo de Viola Jones tiene una probabilidad de verdaderos positivos del 99,9 % y una probabilidad de falso positivos del 3,33 %, y a diferencia de otros algoritmos utilizados en métodos de características invariantes procesa sólo la información presente en una imagen en escala de grises. No utiliza directamente la imagen, sino que utiliza una representación de la imagen llamada imagen integral. Para determinar si en una imagen se encuentra un rostro o no, el algoritmo divide la imagen integral en subregiones de tamaños diferentes y utiliza una serie de clasificadores (clasificadores en cascada), cada una con un conjunto de características visuales. En cada clasificador se determina si la subregión es un rostro o no. El uso de este algoritmo supone un ahorro de tiempo considerable ya que no serán procesadas subregiones de la imagen que no se sepa con certeza que contienen un rostro y sólo se invertirá tiempo en aquellas subregiones que posiblemente si contengan uno. Este detector se ha hecho muy popular debido a su velocidad a la hora de detectar rostros en imágenes y por su implementación en la biblioteca OpenCV [29], la cual a su vez permite su uso en diferentes lenguajes de programación como C/C++, Java y Python 3.

Por otro lado, para el módulo de control del cursor, la conversión de los parámetros del movimiento del rostro humano a la navegación del cursor del mouse se puede clasificar en modo directo, modo de joystick y modo diferencial. Para el modo directo, se establece un mapeo uno a uno desde el dominio de los parámetros de movimiento hasta las coordenadas de la pantalla mediante una calibración fuera de línea o mediante un diseño basado en el conocimiento previo sobre la configuración del monitor [23]. En el modo Joystick, la posición del cursor se establece por la dirección (o el signo) de los parámetros de movimiento. Y la velocidad del movimiento del cursor está determinada por la magnitud de los parámetros de movimiento [11]. En el modo diferencial, la acumulación del desplazamiento de los parámetros de movimiento impulsa la navegación del cursor del mouse, y algún parámetro de movimiento adicional enciende o apaga el mecanismo de acumulación para que el parámetro de movimiento se pueda restablecer sin influir en la posición final cursor. Este modo es muy similar al modo del mouse manual: el usuario puede levantar el mouse y regresar al origen en la superficie plana donde reposa el mouse después de realizar una operación de arrastre del mismo. Una vez la posición del cursor se ha establecido, la ejecución de eventos del cursor, como los clics que dispararán los botones del mouse, se desencadenan mediante la detección

de gestos del rostro. El desencadenante más sencillo se obtiene mediante algunos parámetros geométricos de partes del rostro cuando superan ciertos umbrales específicos, por ejemplo, la relación de aspecto cuando se hace apertura o cierre de boca u ojos. Otro desencadenante se obtiene mediante algunos parámetros de movimiento que superan umbrales específicos temporales. En [30], un evento de clic del mouse se activa por "tiempo de permanencia", por ejemplo, se activa un clic del mouse si el usuario mantiene el cursor quieto durante 0,5 s.

En [12], la confirmación y cancelación de las operaciones del ratón se transmite mediante un movimiento vertical y un movimiento horizontal del rostro. Una máquina de estado finito temporizada está diseñada para detectar estos tipos de movimientos después de posicionarse el cursor.

El objetivo de este trabajo es utilizar la expresión facial como una forma alternativa de controlar el movimiento del cursor. Es un trabajo muy útil para las personas con limitaciones de miembros superiores. Para ello se utilizará, como en otros trabajos, una cámara web interconectada con la computadora portátil del usuario. El sistema detectará automáticamente el rostro y estimará su orientación en el espacio 3D del usuario mediante el algoritmo de Viola-Jones y un modelo flexible. Mediante un criterio de mínima distancia entre *frames* consecutivos de las imágenes provenientes de la webcam se seguirán automáticamente los movimientos del rostro y se dará una salida al sistema de operación del cursor. Para hacer más rápido el posicionamiento del cursor y la generación de clic, este sistema operará únicamente sobre un aplicativo orientado a redes sociales y se detectarán mediante técnicas de visión por computador elementos gráficos claves de interacción lo que facilitará su uso.

3. MATERIALES Y MÉTODOS

Para concebir la solución de interfaz gestual, primero se definió el software WhatsApp de Escritorio como aplicación objetivo para comandar, esto con base en una encuesta preliminar sobre la preferencia de aplicación de las personas con limitaciones motrices para comunicarse con el mundo exterior. Las condiciones de interacción se establecen en la Figura 2. Para tener una buena composición en la imagen capturada mediante la cámara integrada en un laptop sobre una mesa, se define una distancia promedio entre 60 a 70 cm medidos desde la pantalla del computador, esta debe estar inclinada no más de 20° respecto a la normal de la superficie de apoyo del computador.

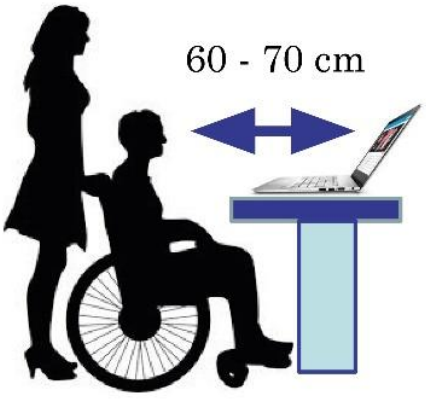

Figura 2. Condiciones de disposición geométrica para interacción con la interfaz gestual
Fuente: elaboración propia.

La primera opción para interactuar con WhatsApp es la de controlar la posición del cursor con ayuda de los movimientos del rostro dado que estos pueden ser estimados utilizando la técnica de detección rostro descrita en la sección 3.2; sin embargo, esto puede resultar tedioso para una persona con limitación motriz, ya que debe estar moviendo constantemente la cabeza para seleccionar con el cursor las zonas de interés.

Para hacer más fácil la interacción se propone realizar desplazamientos específicos del cursor aprovechando que en la aplicación WhatsApp existen dos zonas de interacción claramente diferenciadas: zona de contactos, y la zona de chat (o conversación). Por esta razón se propone no solamente procesar la imagen del rostro capturada mediante una webcam sino también la captura de pantalla del computador donde la persona está visualizando WhatsApp.

Adicionalmente, se propone como comandos de interacción: i) activar o desactivar la interfaz gestual con la apertura de la boca, ii) ubicar el cursor en la zona de conversación o en la zona de contactos mediante movimientos laterales y sostenidos del rostro, iii) hacer scroll up/down en cualquiera de estas zonas mediante movimientos verticales y sostenidos del rostro. La activación del scroll up/down en la zona de conversación permite subir y bajar en la conversación que se ha desarrollado entre dos personas o en un grupo personas. En caso de que el scroll up/down se realice en la zona de contactos permitirá seleccionar un contacto a la vez, esto implica emular 'Ctrl+TAB' o 'Ctrl+Shift+TAB' cada que vez que el cursor se desplaza de forma discreta entre los diferentes contactos.

La solución propuesta contempla tres etapas: preprocesamiento, análisis y generación de eventos (ver Figura 3). Las técnicas utilizadas en cada una de estas etapas se pueden ver en detalle en la Figura 4.

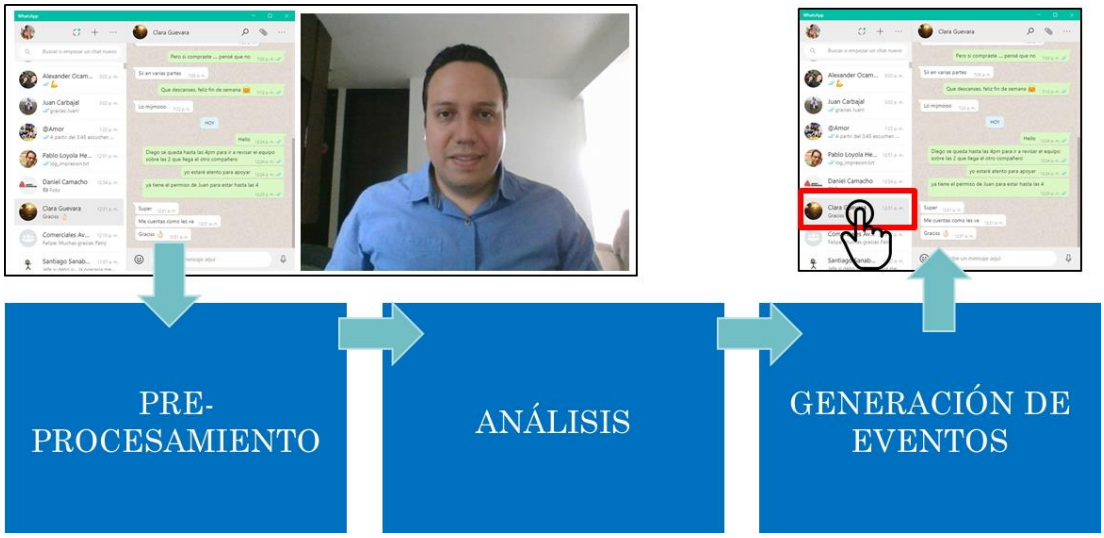

Figura 3. Diagrama de bloques principales de la solución propuesta
Fuente: elaboración propia.

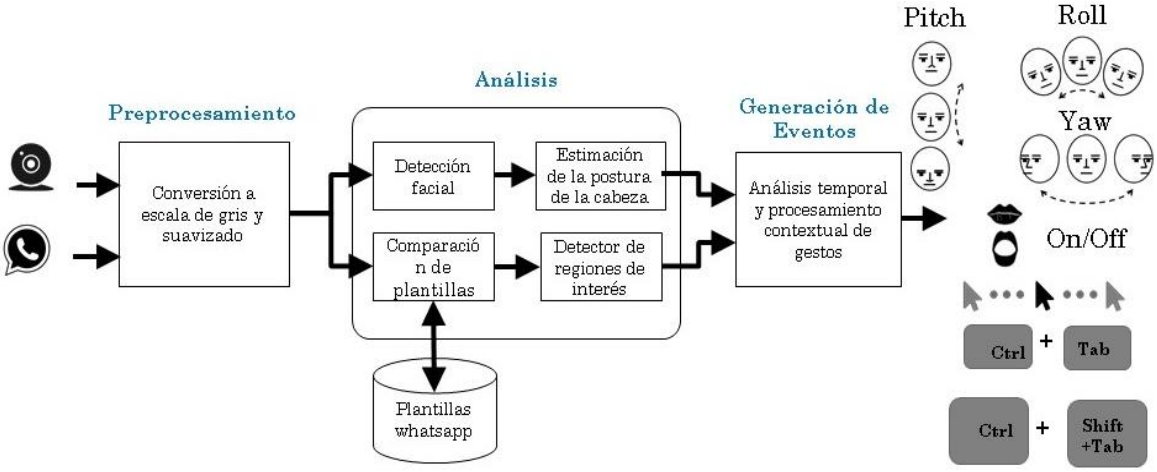

Figura 4. Técnicas implementadas en cada una de las etapas de la solución propuesta
Fuente: elaboración propia.

3.1 Preprocesamiento

La etapa de preprocesamiento tiene dos propósitos fundamentales: reducir la dimensionalidad de los datos y mejorar la relación señal a ruido de las imágenes [31].

La reducción de dimensionalidad consiste en pasar de imágenes de tres canales, RGB, a imágenes de un solo canal en escala de grises. Esto se realiza tanto en la imagen adquirida por webcam como en la captura de pantalla de la aplicación WhatsApp.

La mejora de la relación señal a ruido se lleva a cabo con la aplicación de un filtrado no lineal tipo mediana estadística el cual conserva los bordes únicamente en la imagen adquirida por la webcam (ver Figura 5). El propósito es mejorar la tasa de detección de rostros, principalmente cuando las condiciones de iluminación se reducen. Se probaron diferentes tamaños de kernel (3x3, 5x5, 7x7 y 9x9) para el filtro de mediana. Para cada uno se estimó que el tiempo promedio de ejecución por imagen son los siguientes: 3x3 (151 ms), 5x5 (185 ms), 7x7 (243 ms) y 9x9 (272 ms). Se seleccionó, finalmente, el de tamaño 3x3, porque se obtuvo la misma efectividad en la detección de rostros comparándola con los demás tamaños, y a que su velocidad de ejecución es la más rápida comparada con el resto.

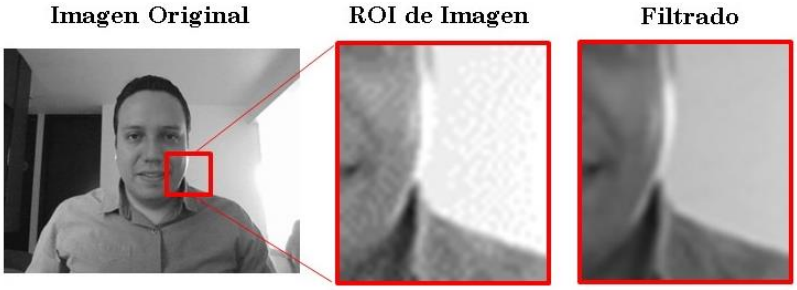

Figura 5. Efecto de aplicación de filtrado de mediana. Fuente: elaboración propia.

3.2 Análisis

La detección de rostros [20], [21] (del inglés *face detection*) es un caso específico de la detección de objetos. La detección de rostros es un proceso por el cual el computador ubica los rostros presentes en una imagen o un vídeo. En la actualidad uno de los algoritmos más representativos para la detección de objetos o, más específicamente de rostros, es el algoritmo

Viola & Jones, debido a sus múltiples factores como su bajo coste computacional, ahorro de tiempo considerable y la facilidad en la detección de objetos, se convierte para los programadores de hoy en día en una de las herramientas más utilizadas. Por otro lado, a pesar de ser un algoritmo supremamente robusto, consta de dos partes fundamentales que facilitan su comprensión, una de ellas es el clasificador cascada que representa una probabilidad de verdaderos positivos del 99,9 % y una probabilidad de falso positivo del 3,33 %. Esta probabilidad de verdaderos positivos es alta a comparación de otros algoritmos debido a que utiliza una representación de la imagen llamada imagen integral, donde se utilizan unos clasificadores en cascada que facilita la detección de cada subregión y así determina si son caras o no. Por otro lado, posee un entrenador de clasificadores basado en Adaboost [11], el cual se enfoca en los datos que fueron erróneamente clasificados y así obtener mejores tiempos en la entrega de información. Este detector se ha hecho muy popular debido a su rapidez a la hora de detectar las caras en imágenes y por su uso en la biblioteca OpenCV [29].

Como resultado de usar está técnica implementada en esta biblioteca de programación, en cada instante de tiempo se tiene un conjunto de ROI asociadas a todos los rostros detectados en la imagen.

Para realizar el seguimiento de un solo rostro el problema puede simplificarse y lograr una mayor velocidad de ejecución entre frames (imágenes en un video) consecutivos si se fija un ROI con etiqueta cero, su etiqueta puede asociarse al ROI más cercano en el siguiente frame utilizando una distancia euclidiana (1) para ello:

$$d_i(p_t, p_{t+1}^i) = \sqrt{p_t \cdot p_{t+1}^i} \tag{1}$$

Donde p_t es el centroide de coordenadas (x_t, y_t) del ROI de etiqueta cero en el instante de tiempo t y p_{t+1}^i es el centroide del i-ésimo ROI detectado en el instante de tiempo t+1. La etiqueta cero se pasará al i-ésimo ROI cuya distancia d_i sea la mínima entre las posibles.

Para estimar la pose del rostro, es decir, encontrar los seis números que determinan la ubicación y orientación que tiene el rostro en el espacio se utiliza la implementación encontrada en la biblioteca DLIB [33], la cual trabaja con un modelo flexible de 68 puntos característicos (ver Figura 6).

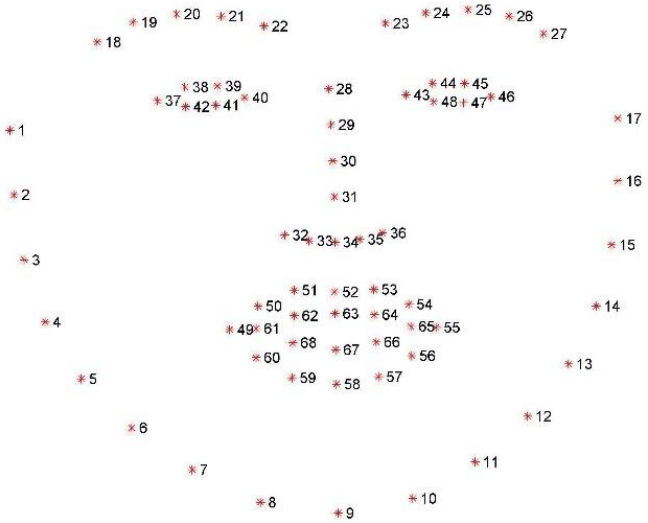

Figura 6. Sesenta y ocho puntos característicos faciales del modelo flexible utilizado en la biblioteca DLIB
Fuente: Adaptado de [33].

A partir de este modelo es posible definir algunas relaciones de aspectos que pueden contribuir a determinar por ejemplo si se ha producido un guiño del ojo o una apertura de boca. Para el caso de la boca, el coeficiente de relación de aspecto se denomina MAR (*Mouth Aspect Ratio*) y está definido a partir de los puntos 61, 62, 63, 64, 65, 66, 67 y 68, ver Figura 7, mediante (2):

$$MAR = \frac{\|p_{61} - p_{68}\| + \|p_{63} - p_{67}\| + \|p_{64} - p_{66}\|}{2\|p_{65} - p_{61}\|} \quad (2)$$

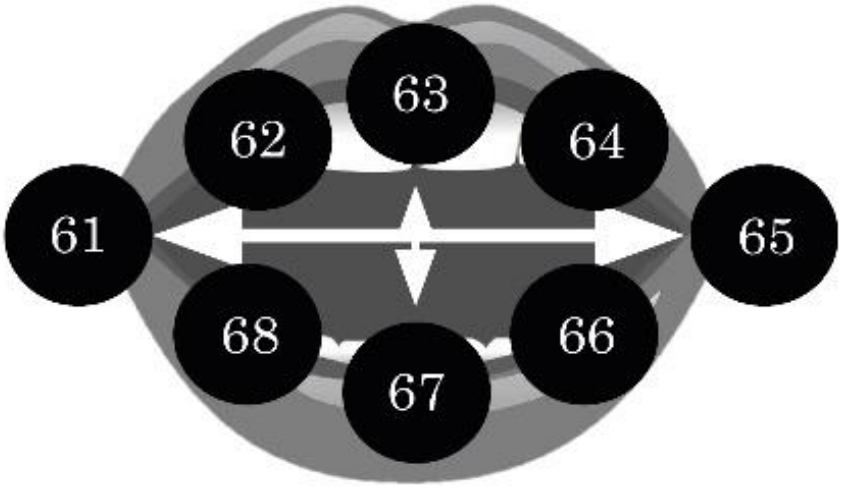

Figura 7. Puntos de interés para definir la relación de aspecto de la boca con base en el modelo flexible de la biblioteca DLIB. Fuente: elaboración propia.

En la Figura 8 se muestra el resultado de detección y estimación de pose obtenida con las técnicas descritas anteriormente.

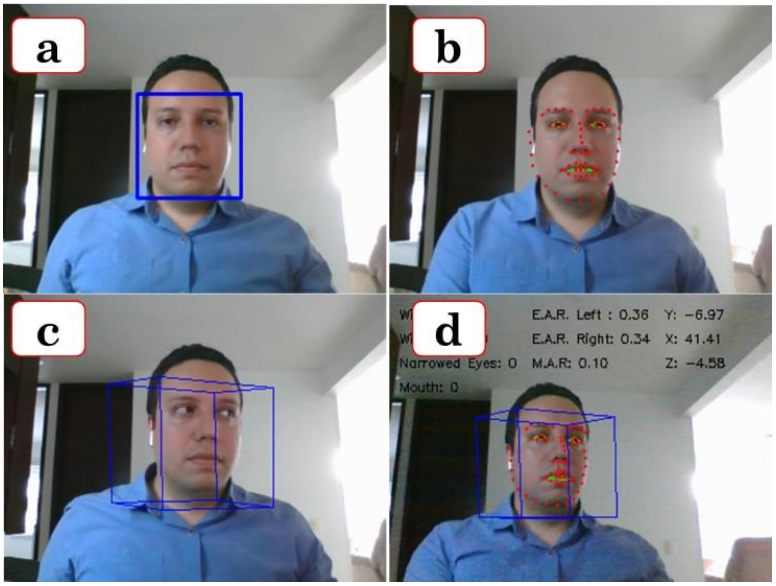

Figura 8. Etapas del proceso de análisis de imagen adquirida por webcam: a) detección de rostro, b) detección y correspondencia de puntos de interés del rostro, c) estimación de pose y d) fusión de resultados de detección de rostro, puntos característicos y estimación de pose. Fuente: elaboración propia.

La detección de zonas de interés se realiza primero obteniendo la región de interés de toda la aplicación de WhatsApp en la captura de pantalla del escritorio del computador. Para esto se utiliza una técnica basada en correspondencia de plantillas (*template matching*). En el procesamiento digital de imágenes la correspondencia por plantilla [24] se utiliza para encontrar pequeñas partes de una imagen que coincidan con una imagen de plantilla. Se puede utilizar en la fabricación como parte del control de calidad, una forma de navegar por un robot móvil o como una forma de detectar objetos en las imágenes. La biblioteca PyAutogui [34] utiliza esta técnica para obtener la región de interés, ROI, que contiene la ventana de una aplicación en la captura de pantalla del escritorio del computador.

Para localizar las zonas de interacción, primero se obtienen las coordenadas de la ventana de WhatsApp, esquina superior de la ventana (X_w, Y_w), y su ancho y alto (W_w, H_w). Estos parámetros son el punto de partida para tener acotado el espacio de detección. Después de analizar los elementos dentro de la ventana, se encontró que el botón de menú podía ser el punto de referencia para separar la ventana en dos zonas y así delimitar las regiones donde se encuentran las zonas de contacto y de conversación. Para encontrar los datos de la posición del botón de menú se toma una muestra del mismo y utilizando la correspondencia por plantillas se obtiene su posición con respecto a la pantalla completa, esto se referenciará al origen de coordenadas de la ventana de WhatsApp para ubicar su posición dentro de la misma, a saber (X_d, Y_d, W_d, Hd) (ver Figura 9).

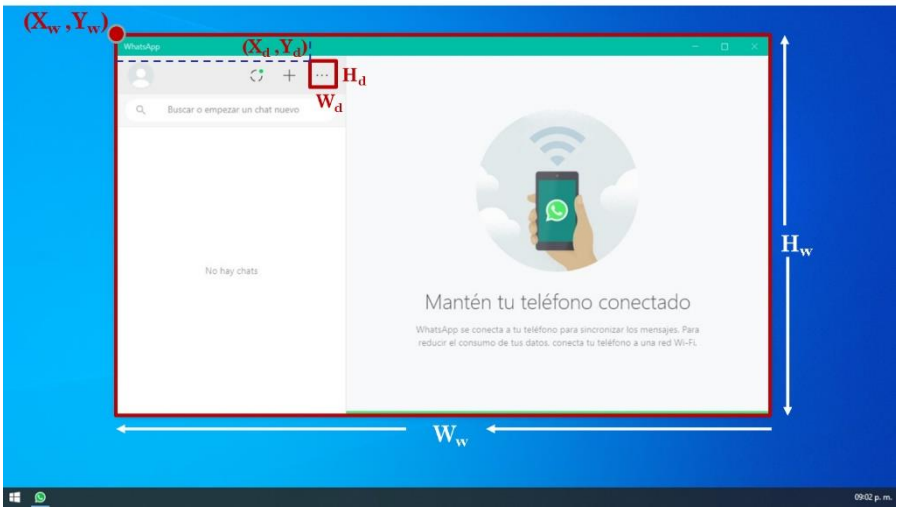

Figura 9. Detección de ROI de la ventana de WhatsApp y del ROI asociado al botón menú
Fuente: elaboración propia.

Con base en los datos del botón menú, se logra separar la ventana de WhatsApp en dos zonas, esto se logra a partir del cálculo de la posición de la esquina superior derecha del icono del menú, la cual se obtiene mediante (3):

$$E_{tr} = Y_d + W_d \tag{3}$$

Se determinó de forma heurística que al valor obtenido anteriormente deben sumarse 16 pixeles para determinar la coordenada en X de la zona Derecha (X_{zd}), y el borde derecho de la zona izquierda, de esta forma se tiene que ($X_{zd} = E_{tr} + 16$). Lo anterior permite calcular el ancho de ambas zonas, W_{zi} y W_{zd}, de la siguiente forma: ($W_{zi} = X_{zd} - X_w$) y ($W_{zd} = W_w - X_{zd}$), Figura 10.

Figura 10. Primera división de la ventana de WhatsApp utilizando criterios geométricos. Fuente: elaboración propia.

Una vez separadas las dos zonas se procede a identificar dos elementos más en la aplicación de WhatsApp para obtener los ROI finales asociados a la zona de contactos y la zona de interacción. Utilizando nuevamente correspondencia de plantillas se localiza el icono de búsqueda de contactos y con esto se logra delimitar mejor la zona de contactos. Para la zona derecha se localiza el icono de emoticones y seguidamente la zona de conversaciones. Si el ROI del icono de búsqueda es (X_b, Y_b, W_b, H_b) y el del icono de emoticón es (X_m, Y_m, W_m, H_m), se puede estimar la posición de las otras dos zonas de interacción de la siguiente forma.

Primero, para la zona de contactos se calcula la coordenada Y_c sumando del área de búsqueda la coordenada Y_b más la altura del área H_b y para la altura de la zona de contactos H_c se usa la altura de la ventana menos la recién calculada coordenada Y_c. Con esto se completan los parámetros de la zona de contactos ya que la posición en X (X_c) es igual a X_{zi} y el ancho es igual a W_{zi}. Por otro lado, para la zona de conversación se empieza tomando las coordenadas ya conocidas e igualándolas con las que se necesita, ya que las coordenadas en X e Y se igualan de la siguiente forma, X_{zc} será igual a X_{zd}, Y_{zc} será igual a Y_b y W_{zc} será igual a W_{zd}. Para calcular la altura habrá que tomar Y_m y restarle Y_{zc}, de esta forma ($H_{cz} = Y_m - Y_{zc}$), y así se obtienen las coordenadas de la zona de conversación ver Figura 11.

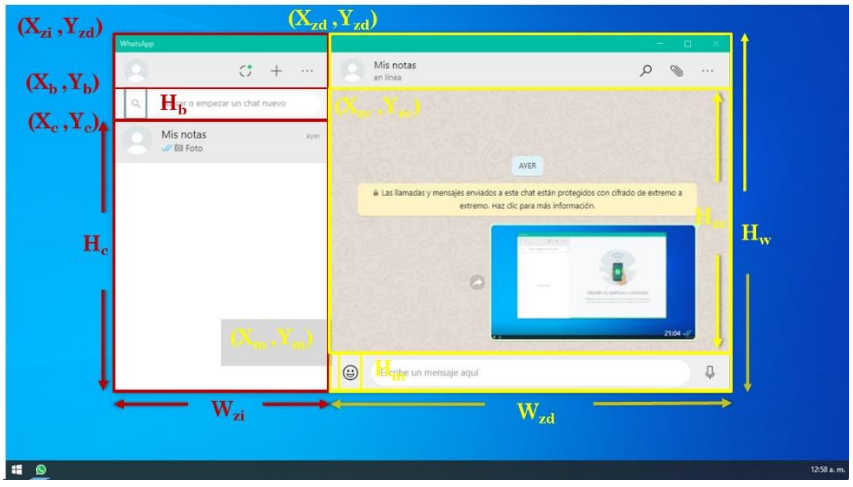

Figura 11. Identificando nuevas plantillas para mejorar la estimación de las zonas de contacto y de conversación. Fuente: elaboración propia.

En la Figura 12 se puede observar un ejemplo del resultado en la detección de zonas de interés utilizando la técnica propuesta anteriormente.

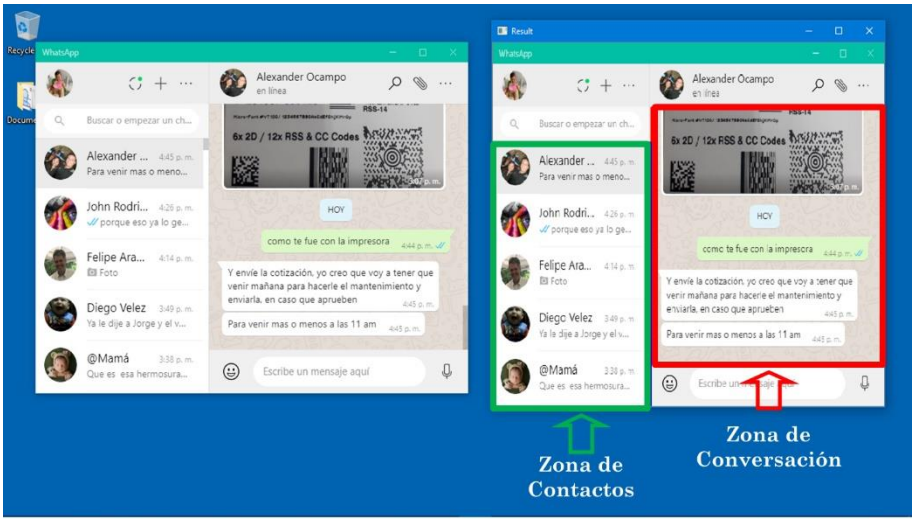

Figura 12. Detección de zonas de interés en aplicación de WhatsApp. Fuente: elaboración propia.

3.3 Generación de eventos e Implementación

En esta etapa se utilizan las señales de roll, pitch y yaw, así como los ROI de interacción en WhatsApp detectados y la relación de aspecto de la boca (MAR), por sus siglas en inglés *Mouth Aspect Ratio*, seguidos en el bloque de análisis.

Se determinó de forma heurística que la mejor forma de posicionar el cursor en el centro del ROI de la zona de conversación debe ocurrir cuando el usuario realice un movimiento en yaw mayor de 15°. Para posicionar el cursor en el centro del ROI de la zona de contactos, el usuario deberá hacer un movimiento en yaw menor de -6°. En caso que el cursor se encuentre en la zona de conversación, con un movimiento en pitch mayor de -5° se interpretará como scroll up y un movimiento en pitch menor de -12° será interpretado como scroll down. En caso que el cursor se encuentre en la zona de contactos, se emulará en 'Ctrl+TAB' (siguiente abajo) o 'Ctrl+Shift+TAB' (siguiente arriba) para activar la conversación con el contacto inmediatamente superior o inferior, respectivamente.

Para detectar la boca abierta, se estableció de forma heurística un umbral del MAR para la boca cerrada de 0,6, al abrir la boca este valor aumentará. Se determinó que, si el usuario permanece con la boca abierta con el umbral superior a 0,6 y durante 3 frames consecutivos, esto se interpretará como una orden para desactivar (cerrar) la interfaz gestual.

La implementación de los diferentes módulos se realizó utilizando el lenguaje Python 3.6.8; el código fuente quedó disponible en [35] bajo licencia MIT y se creó tanto un instalador del aplicativo como un manual de usuario para su instalación, configuración y operación. En síntesis, se debe instalar y ejecutar WhatsApp versión de escritorio en el sistema Operativo Windows 10 y después se debe ejecutar el aplicativo **hciVisualGesture** en el mismo computador (requerimientos mínimos, procesador de 2.0 GHz y RAM 8 GB). Este aplicativo aparecerá en las aplicaciones disponibles en el menú inicio de Windows 10 una vez se haya instalado el programa por primera vez. **hciVisualGesture** se ejecutará en segundo plano como un servicio de Windows y aparecerá en el área de notificaciones en la barra de tareas. Para las próximas versiones del aplicativo se crearán instaladores para dos distribuciones de Linux (Ubuntu 18.04 y Fedora 32) y para Mac OS Mojave.

4. PRUEBAS Y RESULTADOS

Se realizaron pruebas de eficiencia computacional con la interfaz gestual en tres equipos de cómputo de baja (Intel Pentium, 4 GB RAM), media (Intel Core i5, 8 GB RAM) y alta gama (Core i7, 16 GB RAM). También se comparó el tiempo promedio en seleccionar una conversación en la zona de contactos de la aplicación WhatsApp con otras interfaces desarrolladas en otros trabajos. Por otro lado, para evaluar la robustez se estudió el comportamiento de la interfaz frente a cambios de iluminación, presencia de otros usuarios y casos de oclusión tanto del rostro como en la aplicación de WhatsApp. Finalmente, para evaluar la usabilidad en un escenario controlado de pruebas, se solicitó a diez personas que usaran la interfaz durante cinco minutos ejecutando los comandos que fueron previamente enseñados. Durante los dos primeros minutos se instruyó a los sujetos experimentales respecto a los comandos básicos que la interfaz soporta y seguidamente se dejó que los usuarios activaran/desactivaran la interfaz gestual, seleccionaran contactos y navegaran en sus conversaciones. En todos los casos, mediante consentimiento informado, se pidió a los sujetos que abrieran su propia cuenta de WhatsApp para que la navegación fuera más familiar para los mismos.

Es importante mencionar que una evaluación rigurosa de la interfaz desarrollada en este trabajo bajo estándares internacionales es posible siguiendo los criterios establecidos en la ISO 9241-940 [36]. Sin embargo, este tipo de evaluación se proyecta realizar y publicar ante la comunidad científica cuando la interfaz aquí desarrollada sea integrada a otro módulo software que complementa su capacidad de generación de comandos aprovechando las señales de voz en un esquema multimodal de interfaz humano-computador.

4.1 Desempeño

Para determinar si la velocidad de ejecución de la interfaz gestual es independiente de la gama del computador que puede utilizar eventualmente una persona con limitación motriz, se seleccionaron tres equipos con características de baja, media y alta gama. Además, se estudió si la velocidad de ejecución de la interfaz es igual ejecutando solo WhatsApp (además del aplicativo de interfaz gestual) y WhatsApp junto con tres aplicaciones más (Microsoft Word, Mozilla Firefox, Reproductor de Windows Media Player).

Se guarda el tiempo ejecución de cada par de imágenes (webcam y captura de pantalla) que pasa por el flujo de ejecución (preprocesado, análisis, generación de eventos) de la interfaz gestual. Este proceso se realiza 525 veces en cada uno de los escenarios descritos arriba, a saber, WhatsApp ejecutándose solo y con tres aplicaciones más. Los valores promedios para cada uno de los tipos de computadores se muestran en la Tabla 1. Se observa que la rapidez de ejecución de la interfaz gestual es aproximadamente 2 segundos más rápida en el computador de media y alta gama comparado con el de baja gama. Se encontró también que en cualquier tipo de computador la interfaz gestual requiere 1,5 GB en RAM para su ejecución y aproximadamente un 35 % de porcentaje de ocupación de la CPU en los computadores de media y alta gama y de 65 % en el computador de baja gama.

Para determinar si existe una diferencia estadísticamente significativa entre los valores promedios de ejecución por frame, se comprobó primero la normalidad de los datos mediante una prueba de t-Student [37] y se procedió a realizar una prueba de hipótesis sobre los valores promedios de ejecución. En la Figura 13 se observa la distribución de tiempos medidos para cada uno de los tipos de computadores en cada uno de los escenarios mediante diagrama de cajas.

Tabla 1. Tiempo promedio y desviación estándar de ejecución de interfaz gestual. Fuente: elaboración propia.

Características del PC	Tiempo promedio y desviación estándar de ejecución por frame cuando solo se ejecuta WhatsApp (s)	Tiempo promedio y desviación estándar de ejecución por frame cuando se ejecuta WhatsApp y 3 aplicaciones más (s)
Intel Pentium, 1.5 GHz, 4 GB RAM (baja gama)	3.23 (±0.73)	3.52 (±0.98)
Intel Core i5, 2.0 GHz, 8 GB RAM (media gama)	1.14 (±0.41)	1.17 (±0.38)
Intel Core i7, 2.5 GHz, 16 GB RAM (alta gama)	1.05 (±0.25)	1.00 (±0.26)

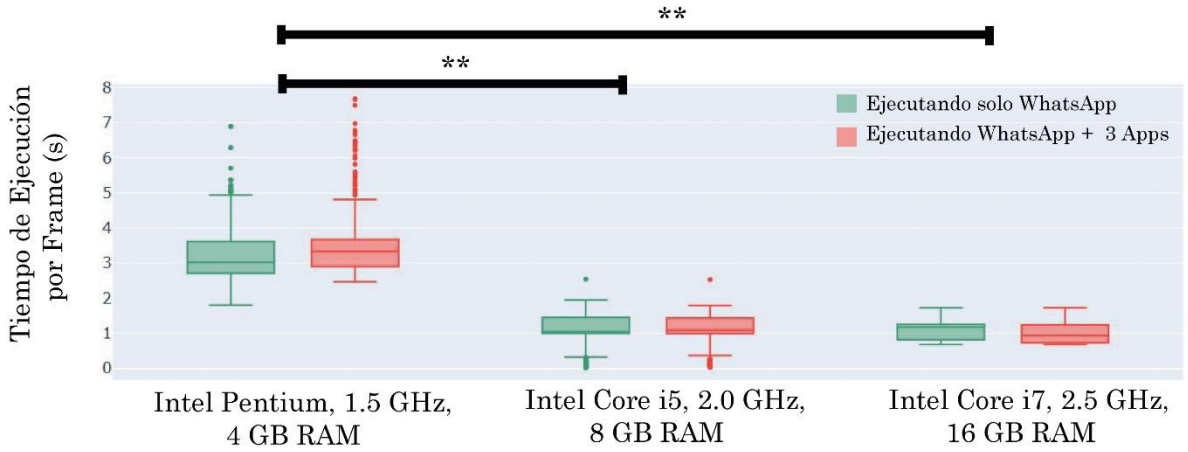

Figura 13. Diagrama de cajas de tiempo de ejecución por frame de la aplicación de interfaz gestual (** p-valor < 0.001). Fuente: elaboración propia.

Se encontró que no es posible establecer diferencia de tiempos promedios entre los escenarios donde se ejecuta solo WhatsApp y WhatsApp más tres aplicaciones. Esto se debe principalmente a que en ningún caso el porcentaje de ocupación de la CPU está al 100 %, lo que significa que la aplicación de interfaz gestual puede ejecutarse junto con otras aplicaciones simultáneamente y no afectar el desempeño de la CPU. Por otro lado, se observa que la diferencia de promedios es estadísticamente significativa (p-valor < 0.001) entre el computador de baja gama y los de media gama y alta gama. No se encontró diferencia de tiempos de ejecución promedio entre el computador de media gama y el de alta gama, lo que indica que es posible operar la interfaz con un tiempo de ejecución promedio por frame de aproximadamente 1,2 segundos desde un computador cuyo precio en la actualidad no supera los 1,6 millones de pesos colombianos (este valor se obtuvo consultando los principales proveedores de la ciudad de Santiago de Cali).

Un experimento adicional, realizado con 5 personas, el cual consistió en lograr hacer clic (100 repeticiones) sobre un contacto (en la zona de contactos de WhatsApp, Figura 12), permitió determinar que en promedio nuestra propuesta (**hciVisualGesture**) es más rápida (p-valor < 0.001) que *Camera Mouse* [11] y *Nose Tracking* [12] (ver Figura 14).

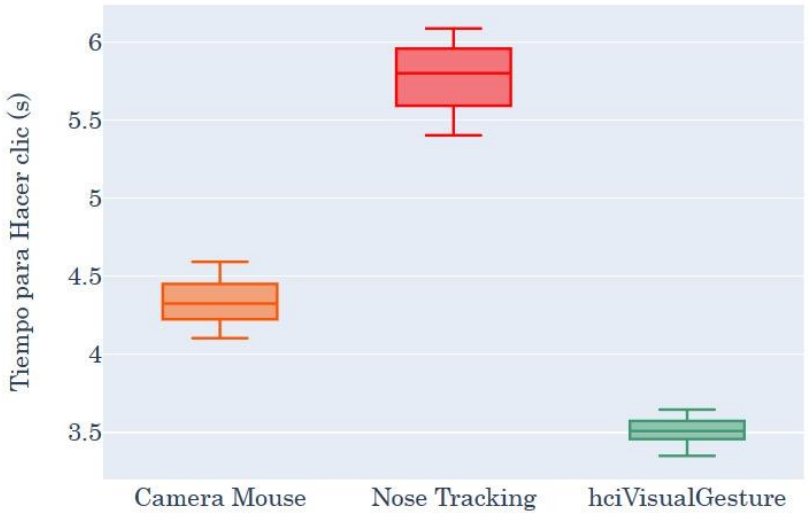

Figura 14. Tiempo para hacer clic: comparación de dos sistemas reportados en la literatura (Camera Mouse [11] y Nose Tracking [12]) y la propuesta de este trabajo denominada hciVisualGesture. Las pruebas fueron realizadas en un PC con Intel Core i7 2.5 GHz y 16 GB RAM. Fuente: elaboración propia.

4.2 Robustez

La interfaz gestual depende de dos técnicas claves para su funcionamiento: la detección de rostros y la detección de zonas de interacción. Por esta razón en este proyecto se estudia la robustez de estos bloques de forma cualitativa.

En la Figura 15 se observa que la detección de rostros es efectiva en diferentes condiciones de iluminación, incluso en completa oscuridad utilizando solo la luz emitida por el computador con el nivel de brillo al máximo. Para este caso el sujeto se ubicó a 80 cm de la pantalla del computador. La técnica falla cuando las condiciones de iluminación son muy bajas.

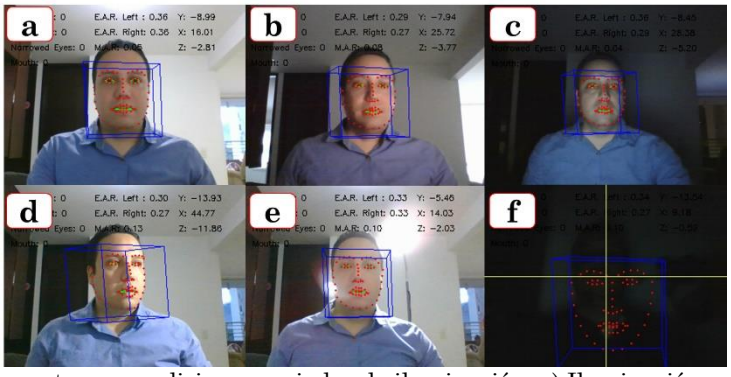

Figura 15. Detección de rostro en condiciones variadas de iluminación. a) Iluminación cuasi-uniforme dentro de habitación durante el día con lámpara superior, b) Sin iluminación de lámpara, c) Durante la noche con el brillo del computador al máximo, d) Lámpara iluminando lateralmente, e) Lámpara detrás del sujeto y de frente a la cámara, y f) Durante la noche con el brillo del computador al mínimo. Fuente: elaboración propia.

El criterio de seguimiento de una sola etiqueta permite que la interfaz no sea comandada por un segundo sujeto en presencia de otras personas en la misma escena del sujeto principal (quien debe operar la interfaz). La Figura 16 muestra este hecho.

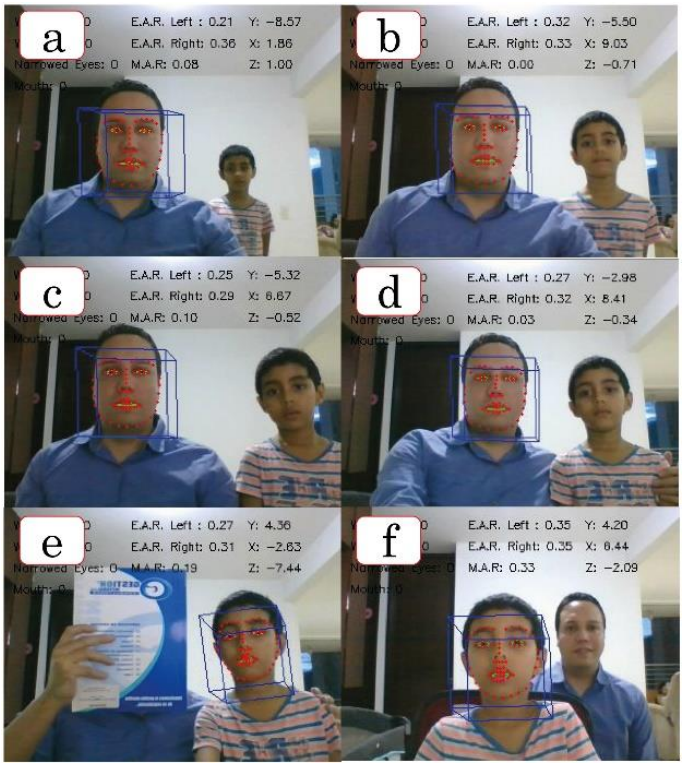

Figura 16. Algoritmo de seguimiento de una sola etiqueta. En a), b), c) y d) se observa que aun cuando hay dos rostros la etiqueta se cede de frame a frame utilizando el criterio de mínima distancia al primer sujeto de izquierda a derecha, e) La etiqueta es cedida al segundo sujeto ya que el primero desapareció de la escena y f) La etiqueta la conservará el segundo sujeto mientras se cumpla el criterio de mínima distancia
Fuente: elaboración propia.

La detección de rostro funciona bien hasta distancias menores de los 90 cm, después de este valor, aun cuando se detecta el rostro, el algoritmo de detección de pose empieza a fallar debido a que la detección de puntos característicos en el rostro no es tan robusta a esta distancia. Finalmente, cuando la distancia es mayor de los 120 cm el algoritmo de detección de rostros falla completamente (ver Figura 17).

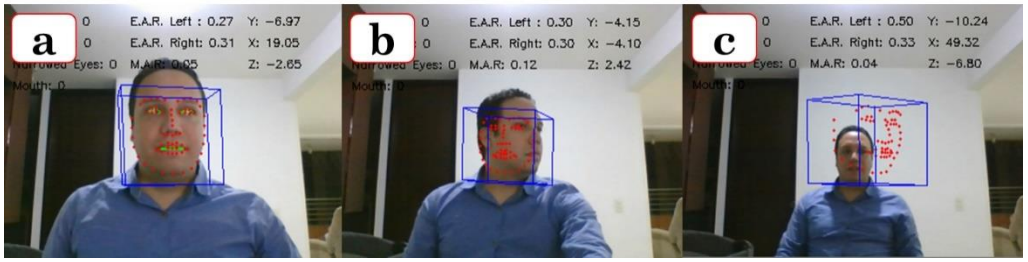

Figura 17. Efecto de la distancia en la efectividad del algoritmo de detección de rostros. a) < 90 cm, b) >90-120 cm y c) > 120 cm. Fuente: elaboración propia.

Es importante aclarar que en este caso y para lograr que la interfaz gestual ejecutara rápidamente la imagen de entrada de webcam es redimensionado siempre a 320x340 antes de ser aplicada dentro del flujo de preprocesamiento, análisis y generación de eventos.

Trabajar a una resolución mayor mejorará la efectividad de la detección en distancias superiores a los 120 cm, sin embargo, hará que el aplicativo de interfaz gestual se ejecute más lentamente.

Las oclusiones son críticas en cualquier aplicación de detección de objetos utilizando imágenes, y la detección de rostro no es la excepción; sin embargo, debe resaltarse que durante las pruebas se pudo observar que casos extremos como el uso de mascarillas, ponerse la mano en el rostro o incluso aparecer a medio cara en la imagen son casos en los que la detección del rostro no falla; no obstante, la detección de puntos característicos sobre el mismo sí afectan directamente la etapa de estimación de pose del rostro (ver Figura 18). Poseer cabello largo, el uso de gafas con lentes transparentes e incluso desaparecer parte del rostro en la imagen, siempre que se alcance a ver la nariz y boca, no afectan la detección y estimación de pose del rostro.

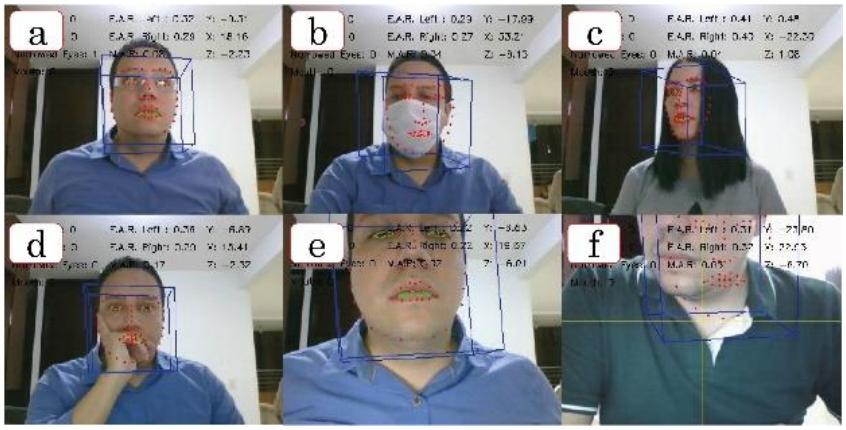

Figura 18. Casos de oclusión sobre el rostro. a) Uso de gafas con lentes transparentes, b) Uso de mascarilla, c) Abundante cabello, d) Mano sobre el rostro, e) Parte del rostro (se alcanza a detectar ojos) por fuera de la imagen, y f) Parte del rostro (no se alcanza a detectar ojos) por fuera de la imagen. Fuente: elaboración propia.

En el caso de la detección de zonas de interés en la aplicación de WhatsApp a partir de la captura de pantalla se observó que la detección de dichas zonas es invariante a escala y traslación (ver Figura 19 y Figura 20).

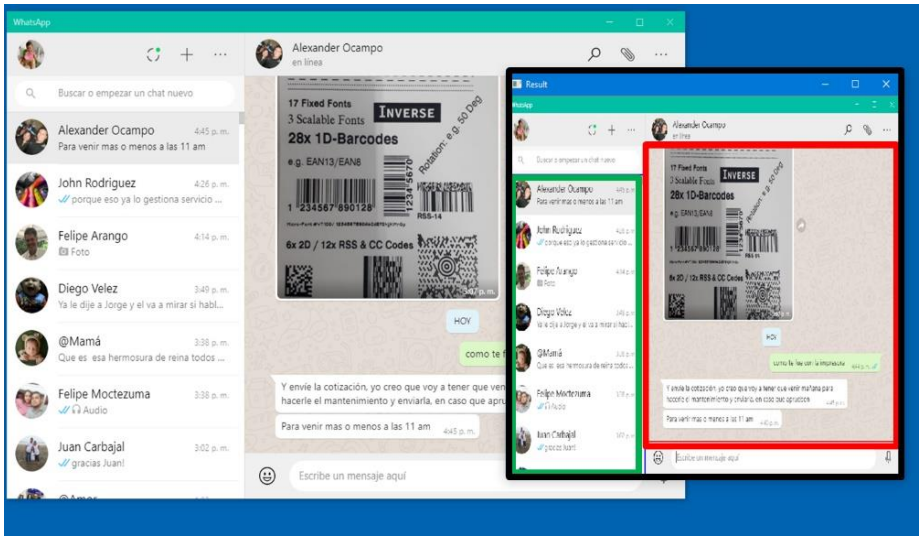

Figura 19. Detección de zonas de interacción con ventana de WhatsApp reducida de tamaño y traslado de origen de coordenadas. Fuente: elaboración propia.

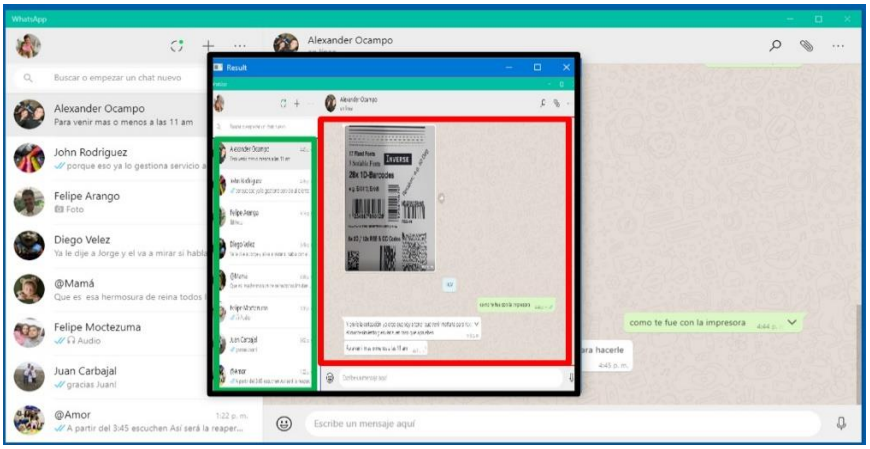

Figura 20. Detección de zonas de interacción con ventana de WhatsApp. Fuente: elaboración propia.

Esto se debe principalmente a que la técnica depende de poder identificar por correspondencia de plantillas los iconos de menú, búsqueda y emoticón. En cualquier caso, que no se logre observar con claridad estos íconos, en la captura de pantalla se tendrá un resultado no deseado, tal y como se evidencia en la Figura 21 y Figura 22.

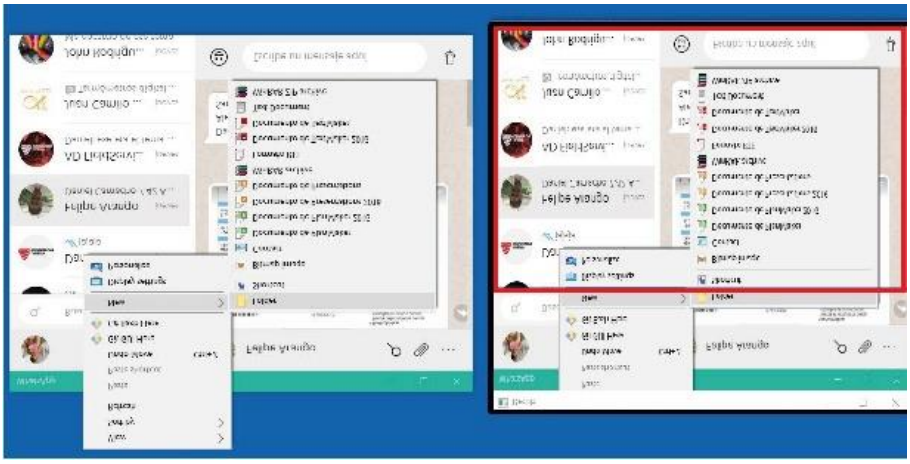

Figura 21. Detección errada de zonas de interacción debido a la presencia de elementos contextuales del sistema operativo cubriendo el icono de menú. Fuente: elaboración propia.

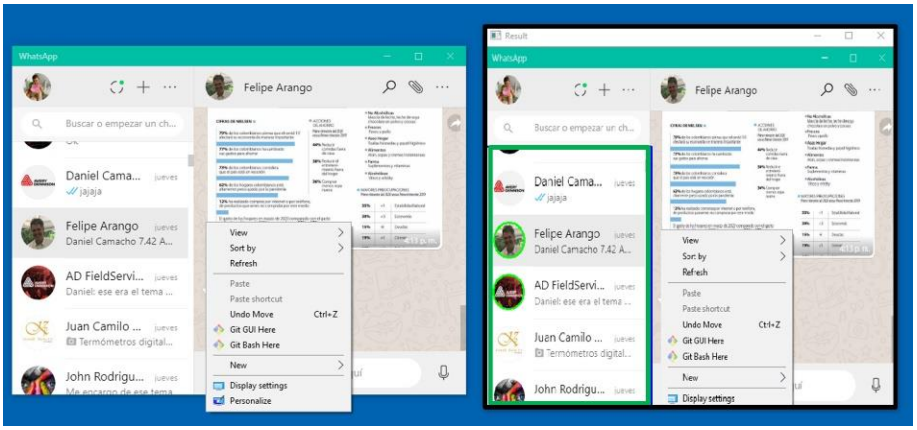

Figura 21. Detección de una sola zona de interacción debido a la presencia de elementos contextuales del sistema operativo cubriendo el icono de emoticón. Fuente: elaboración propia.

La Figura 22 es un caso que soporta la tesis que el algoritmo solo depende de la correcta identificación de los íconos ya mencionados. En este caso la ventana de la aplicación no obstruye ninguno de los íconos y por tal razón se obtiene una correcta identificación de las zonas de interacción. Sin embargo, en caso que la aplicación de WhatsApp se encuentre en segundo plano, los comandos de interacción no tendrán ningún efecto, por el contrario, es posible tener algún efecto no deseado sobre la aplicación que esté en primer plano en ese momento.

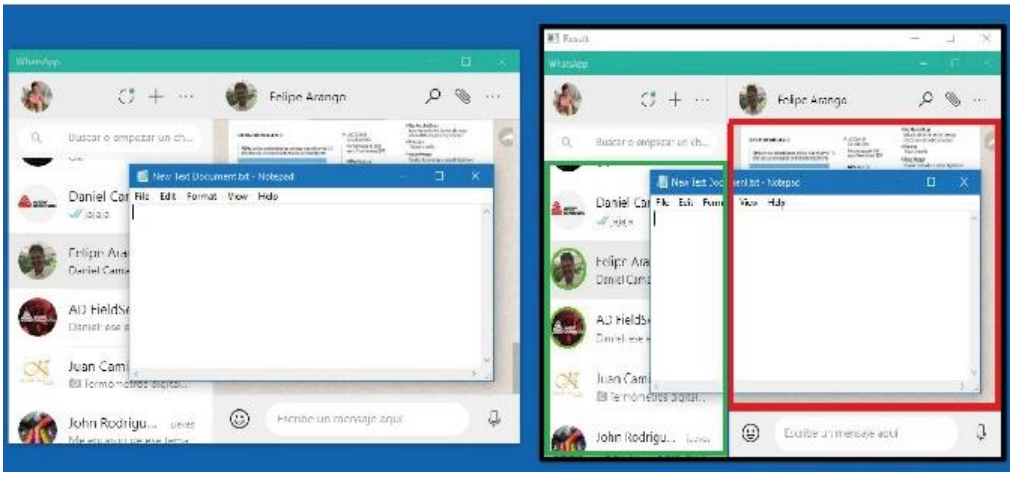

Figura 22. Detección de zonas de interacción con presencia de ventana de aplicación Bloc de Notas de Windows que no cubre ninguno de los iconos claves para la detección de plantillas. Fuente: elaboración propia.

4.3 Usabilidad

Con el fin de explorar la facilidad de uso de la interfaz gestual propuesta se realizó un experimento con once personas (seis mujeres y cinco hombres) cuyas edades varían de los 12 a los 67 años. Con ayuda de un consentimiento informado se daba cuenta del experimento en que los sujetos iban a participar y, una vez firmado el mismo, se procedía con la explicación del procedimiento experimental, el cual consistía durante los primeros 3 minutos en explicar el uso del mismo. Durante las pruebas las personas abrían su cuenta de WhatsApp para estar más familiarizados con los contactos que iban a seleccionar durante las pruebas.

La prueba consistió en utilizar los comandos de interacción desarrollados en la propuesta de solución, a cada usuario se le solicitó seleccionar 10 contactos personales o grupales de la zona de contactos y que navegara de arriba hacia abajo y de abajo hacia arriba en las conversaciones. La supervisión del experimento siempre se llevó a cabo desde la parte trasera del computador para evitar violar la intimidad de las conversaciones de los sujetos experimentales. El tiempo promedio de interacción con WhatsApp mediante la interfaz gestual fue de 10 minutos. Una vez finalizado el ejercicio de interacción con ayuda de un formulario, Tabla 2, se recogía información sobre la percepción de los usuarios sobre la solución de interfaz gestual propuesta. De las respuestas obtenidas en forma porcentual para cada una de las encuestas se observó que en general la percepción de los sujetos experimentales es que la aplicación de la interfaz gestual es amigable, con una curva rápida de aprendizaje y permite interactuar con WhatsApp de forma efectiva bajo los comandos que actualmente permite la misma. Sin embargo, se percibe que la interfaz no es tan rápida en su ejecución, de hecho, algunos sujetos manifestaron que consideran debe mejorarse esta parte dentro de la interfaz para hacerla más intuitiva.

Tabla 2. Formulario de encuesta para las pruebas de usabilidad. Fuente: elaboración propia.

No.	Encuesta
1	La Interfaz detecta el rostro.
2	La interfaz detecta y posiciona correctamente el cursor en las zonas de contactos y mensajes en WhatsApp de escritorio.
3	La interfaz reconoce todos los comandos de navegación (Mov.Izquierda; Mov.Derecha; Mov.Arriba; Mov.Abajo).
4	La interfaz reconoce el comando gestual para salir de la interfaz (boca abierta).
5	La ejecución de los comandos en la interfaz es rápida.
6	Teniendo en cuenta que la interfaz está pensada para personas con tetraplejia, considera que la herramienta ayuda establecer comunicación con WhatsApp de escritorio.
7	La interfaz es de fácil uso para el usuario.
8	La interfaz se puede utilizar en cualquier ambiente de iluminación sin interferencia en la detección del rostro.
9	La interfaz cumple el objetivo de realizar tareas básicas de navegación en la aplicación WhatsApp desktop.
10	Pensando en la necesidad de las personas con limitaciones motrices de miembros superiores, considera que la interfaz es una herramienta que se puede recomendar.

5. CONCLUSIONES

Se desarrolló una interfaz humano-computador basada en gestos faciales y detección de zonas de interés en una aplicación orientada al internet para personas con limitaciones motrices de miembros superiores. La interfaz permite mediante cinco comandos seleccionar contactos, navegar en las conversaciones de la aplicación WhatsApp de escritorio, así como activar/desactivar la interfaz utilizando únicamente gestos del rostro.

Se determinaron las principales características de una interfaz humano-computador para personas con limitaciones motrices de miembros superiores. La interfaz cuenta con principios ergonómicos para facilitar y optimizar su utilización. También posee una combinación de tecnología, conocimiento y recursos que produjeron resultados deseados en equipos de cómputo con precios inferiores a los 1,5 millones pesos colombianos.

Se implementó una técnica de visión por computador para la identificación de gestos faciales y la detección de zonas de interés en una aplicación de escritorio orientada a internet.

Se logró la detección automática, a partir de una captura de pantalla de la aplicación WhatsApp, de las zonas de contacto y conversaciones mediante una combinación de heurísticas y correspondencia por plantilla de imágenes. La detección de gestos se logró utilizando técnicas robustas de detección de rostros y estimación de pose mediante modelos flexibles.

Se construyó una interfaz software de generación de comandos para una aplicación de escritorio a partir de gestos y zonas de interés detectadas. Esta interfaz es capaz de correr en fondo y no interfiere visualmente con la aplicación WhatsApp. La velocidad de ejecución de la misma es de 1 Hz, y en un equipo de media gama ocupa el 35 % de CPU y 1,5 GB RAM.

Se llevaron a cabo pruebas de desempeño computacional, robustez y de usabilidad.

Primero, las pruebas de desempeño permitieron identificar que en un equipo de media gama es posible trabajar la interfaz gestual y que se pueden tener más aplicaciones ejecutándose simultáneamente sin que esto reduzca la velocidad de ejecución de la misma.

Además, la propuesta desarrollada en este trabajo demostró, mediante un experimento sencillo, ser más rápido que Camera Mouse y Nose Tracking, dos sistemas vigentes en el estado del arte; sin embargo, el uso de la interfaz desarrollada en este trabajo se orienta a

una sola aplicación, mientras que Camera Mouse y Nose Tracking permiten comandar cualquier aplicación. Segundo, las pruebas de robustez evidenciaron que la interfaz gestual puede trabajar en variadas condiciones de iluminación y con algunos casos de oclusión tanto a nivel de detección de gestos faciales como detección de zonas de interés. Por último, las pruebas de usabilidad permitieron entender que los usuarios reconocen la funcionalidad de la interfaz implementada, así como su potencial para personas con limitación motriz; sin embargo, sugieren que debe ampliarse el número de eventos y de capacidades de la interfaz para ser completamente útil, además de sugerir una mejora en los tiempos de ejecución para que su manejo sea más fluido.

En general, los autores de este trabajo consideran que los hallazgos encontrados en la ejecución de este proyecto son claves para su aplicación en personas con limitaciones motrices de miembro superior y que deben integrarse a la parte de reconocimiento de voz (el cual es otro proyecto que se está realizando en la Fundación Universitaria Lumen Gentium) para complementar su espectro de aplicación.

Finalmente, con el fin ampliar el alcance obtenido y de igual forma superar las limitaciones de la solución propuesta en este trabajo de investigación, se propone a futuro i) Incluir más eventos a partir de la identificación de otros gestos del rostro, por ejemplo, los que involucran los ojos, ii) Mejorar el tiempo de ejecución de la interfaz gestual mediante la re-implementación de algunas técnicas de análisis de imágenes para que escalen mejor las capacidades de la CPU, y iii) Integrar reconocimiento por voz para permitir el ingreso de texto utilizando técnicas de procesamiento del lenguaje natural.

6. AGRADECIMIENTOS

A la Fundación Universitaria católica Lumen Gentium y a su Dirección de Investigaciones por el apoyo económico y administrativo brindado al proyecto.

CONFLICTOS DE INTERÉS DE LOS AUTORES

Declaramos no tener ningún tipo de conflicto de intereses, ninguna relación personal, política, interés financiero ni académico que pueda influir en nuestro juicio.

CONTRIBUCIÓN DE LOS AUTORES

Carlos Ferrin-Bolaños, José Mosquera-De la Cruz y Humberto Loaiza-Correa, contribuyeron significativamente en la conceptualización, diseño y desarrollo de la investigación, así como a la edición y revisión del artículo. John Pino-Murcia, Luis Moctezuma-Ruiz y Jonathan Burgos-Martínez y Luis Aragón-Valencia contribuyeron en la implementación de los diferentes módulos software y en la elaboración y ejecución de las diferentes pruebas experimentales.

7. REFERENCIAS

[1] J. H. Mosquera-DeLaCruz; H. Loaiza-Correa; S. E. Nope-Rodríguez; A. D. Restrepo-Giró, "Human-computer multimodal interface to internet navigation," *Disabil. Rehabil. Assist. Technol.*, pp. 1–14, Jul. 2020. https://doi.org/10.1080/17483107.2020.1799440

[2] C. Ferrin-Bolaños; H. Loaiza-Correa; J. Pierre-Diaz; P. Vélez-Ángel, "Evaluación del aporte de la covarianza de las señales electroencefalográficas a las interfaces cerebro-computador de imaginación motora para pacientes con lesiones de médula espinal," *TecnoLógicas*, vol. 22, no. 46, pp. 213–231, Sep.. 2019. https://doi.org/https://doi.org/10.22430/22565337.1392

[3] Ministerio de Salud y Protección Social Oficina de Promoción Social de Colombia, "Sala Situacional Situación de las Personas con Discapacidad," Junio. 2018. URL

[4] L. Cortés-Rico; G. Piedrahita-Solórzano, "Interacciones basadas en gestos: revisión crítica," *TecnoLógicas*, vol. 22, pp. 119–132, Dic. 2019, https://doi.org/10.22430/22565337.1512

[5] N. Balsero; D. Botero; J. Zuluaga; C. Parra Rodríguez, "Interacción hombre-máquina usando gestos manuales en texto real," *Ing. y Univ.*, vol. 9, no. 2, pp. 101–112, 2005. URL

[6] W. A. Castrillón Herrera, "Implementación de una Interfaz Hombre-Máquina para el Control de un Brazo Robótico Mediante Posturas Labiales," (Trabajo de Grado), Universidad Nacional de Colombia, Manizales, 2009. URL

[7] J. H. Mosquera; H. Loaiza; S. E. Nope; A. D. Restrepo, "Identifying facial gestures to emulate a mouse: navigation application on Facebook.," *IEEE Lat. Am. Trans.*, vol. 15, no. 1, pp. 121–128, Jan. 2017, https://doi.org/10.1109/TLA.2017.7827915

[8] C. Mauri, T. Granollers; J. Lorés; M. García "Computer vision interaction for people with severe movement restriction," *An Interdiscip. J. Humans ICT Environ.*, vol. 2, pp. 38–54, Apr. 2006. URL

[9] E. Perini; S. Soria; A. Prati; R. Cucchiara, "FaceMouse: A Human-Computer Interface for Tetraplegic People," in *Lecture Notes in Computer Science,* Springer-Verlag Berlin Heidelberg, pp. 99–108, 2006. URL

[10] J. Varona; C. Manresa-Yee; F. J. Perales, "Hands-free vision-based interface for computer accessibility," *J. Netw. Comput. Appl.*, vol. 31, no. 4, pp. 357–374, Nov. 2008, https://doi.org/10.1016/j.jnca.2008.03.003

[11] M. Betke; J. Gips; P. Fleming, "The Camera Mouse: visual tracking of body features to provide computer access for people with severe disabilities," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 10, no. 1, pp. 1–10, Mar. 2002. https://doi.org/10.1109/TNSRE.2002.1021581

[12] S. S. Khan; M. S. H. Sunny; M. S. Hossain; E. Hossain; M. Ahmad, "Nose tracking cursor control for the people with disabilities: An improved HCI," en *2017 3rd International Conference on Electrical Information and Communication Technology (EICT)*, Khulna 2017, pp. 1–5. https://doi.org/10.1109/EICT.2017.8275178

[13] A. Matos; V. Filipe; P. Couto, "Human-computer interaction based on facial expression recognition: A case study in degenerative neuromuscular disease," *ACM Int. Conf. Proceeding Ser.*, pp. 8–12, Dec. 2016. https://doi.org/10.1145/3019943.3019945

[14] A. Rabhi; A. Sadiq; A. Mouloudi, "Face tracking: state of the art," in *2015 Third World Conference on Complex Systems (WCCS)*, Marrakech. 2015, pp. 1–8. https://doi.org/10.1109/ICoCS.2015.7483308

[15] P. Premaratne, *Human Computer Interaction Using Hand Gestures*. Singapore: Springer Singapore, 2014.

[16] L. Nanni; S. Brahnam; A. Lumini, "Face Detection Ensemble with Methods Using Depth Information to Filter False Positives," *Sensors*, vol. 19, no. 23, p. 5242, Nov. 2019, https://doi.org/10.3390/s19235242

[17] M. W. Ni, "Facial image registration," (Tesis Doctoral), Electrotechnique, Automatique et ´Traitement du Signal, l'universite de Grenoble, 2017. URL

[18] M. H. Teja, "Real-time live face detection using face template matching and DCT energy analysis," in *2011 International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, Dalian. 2011, pp. 342–346. https://doi.org/10.1109/SoCPaR.2011.6089267

[19] A. Aldhahab; T. Alobaidi; A. Q. Althahab; W. B. Mikhael, "Applying Multiresolution Analysis to Vector Quantization Features for Face Recognition," in *2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*, Dallas 2019, pp. 598–601. https://doi.org/10.1109/MWSCAS.2019.8885188

[20] S. Zafeiriou; C. Zhang; Z. Zhang, "A survey on face detection in the wild: Past, present and future," *Comput. Vis. Image Underst.*, vol. 138, pp. 1–24, Sep. 2015. https://doi.org/10.1016/j.cviu.2015.03.015

[21] A. Kumar; A. Kaur; M. Kumar, "Face detection techniques: a review," *Artif. Intell. Rev.*, vol. 52, pp. 927–948, Agu.2019. https://doi.org/10.1007/s10462-018-9650-2

[22] F. Pujol; M. Pujol; A. Jimeno-Morenilla; M. Pujol, "Face Detection Based on Skin Color Segmentation Using Fuzzy Entropy," *Entropy*, vol. 19, no. 1, p. 26, Jan. 2017. https://doi.org/10.3390/e19010026

[23] E. Perini; S. Soria; A. Prati; R. Cucchiara, "FaceMouse: A human-computer interface for tetraplegic people,"

Lect. Notes Comput. Sci, Berlin, 2006, pp. 99–108. https://doi.org/10.1007/11754336_10

[24] R. Brunelli, *Template Matching Techniques in Computer Vision: Theory and Practice*. JohnWiley & sons ltda. 2009

[25] V. S. R. Middi, K. J. Thomas; T. A. Harris, "Facial Keypoint Detection Using Deep Learning and Computer Vision," Springer International Publishing, 2020, pp. 493–502. https://doi.org/10.1007/978-3-030-16660-1_48

[26] A. Divya; K. B. Raja; K. R. Venugopal, "Face Recognition Based on Windowing Technique Using DCT, Average Covariance and Artificial Neural Network," in *2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, Bangkok 2018, pp. 335–342. https://doi.org/10.1109/ICIIBMS.2018.8549981

[27] J. Huang; Y. Shang; H. Chen, "Improved Viola-Jones face detection algorithm based on HoloLens," Eurasip J. Image Video Process., vol. 2019, no. 1, 2019. https://doi.org/10.1186/s13640-019-0435-6

[28] Y. Freund; R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997, https://doi.org/10.1006/jcss.1997.1504

[29] I. Culjak; D. Abram; T. Pribanic; H. Dzapo; M. Cifrek M, "A brief introduction to OpenCV," en *Proceedings of the 35th International Convention MIPRO, Opatija* 2012. URL

[30] J. H. Mosquera; E. Oliveros, "Interacción Humano-Máquina Audiovisual," (Trabajo de grado), Universidad del Valle, Santiago de Cali, 2011. URL

[31] V. Londoño-Osorio; J. Marín-Pineda; E. I. Arango-Zuluaga, "Introducción a la Visión Artificial mediante Prácticas de Laboratorio Diseñadas en Matlab," *TecnoLógicas*, edición especial, pp. 591-603, Nov. 2013. https://doi.org/10.22430/22565337.350

[32] C. Sagonas; G. Tzimiropoulos; S. Zafeiriou; M. Pantic, "300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge," in *2013 IEEE International Conference on Computer Vision Workshops*, 2013, pp. 397–403. https://doi.org/10.1109/ICCVW.2013.59

[33] X. Ren; J. Ding; J. Sun; Q. Sui, "Face modeling process based on Dlib," in *2017 Chinese Automation Congress (CAC)*, Jinan 2017, pp. 1969–1972. https://doi.org/10.1109/CAC.2017.8243093

[34] A. Sweigart, "Welcome to PyAutoGUI's documentation!," Read the Docs, 2020. URL

[35] C. Ferrin; J. Pino, "hciVisualGesture," 2020. URL

[36] ISO 9241-940:2017, *Ergonomics of human-system interaction - Part 940:Evaluation of tactile and haptic interactions*. Switzerland, 2017. URL

[37] Z. Ali; S. B. Bhaskar, "Basic statistical tools in research and data analysis," *Indian J. Anaesth.*, vol. 60, no. 9, pp. 662-669, 2016. https://doi.org/10.4103/0019-5049.190623