

Instituto Tecnológico de Costa Rica

Carrera de Ingeniería Mecatrónica



Robótica Inteligente: Reconocimiento gestual para determinar cuando un conductor utiliza el celular mientras conduce

Informe de Proyecto de Graduación para optar por el título de Ingeniero en Mecatrónica con el grado académico de Licenciatura

Lugar de ejecución del proyecto:

Universidade de São Paulo, São Carlos, São Paulo, Brasil

Profesor Asesor: Dr. Juan Luis Crespo Mariño

**Diana Esquivel González
201121714**

Diciembre de 2015

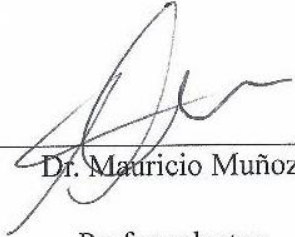
INSTITUTO TECNOLÓGICO DE COSTA RICA
CARRERA DE INGENIERÍA MECATRÓNICA
PROYECTO DE GRADUACIÓN
ACTA DE APROBACIÓN

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Mecatrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.


Estudiante: Diana Esquivel González

Nombre del proyecto: **Reconocimiento gestual para determinar cuando un conductor utiliza el celular mientras conduce**

Miembros del Tribunal



Dr. Mauricio Muñoz
Profesor lector



Dr. Juan Luis Crespo Mariño
Profesor asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Carrera de Ingeniería Mecatrónica

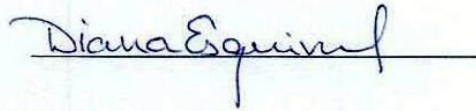
Cartago, Diciembre 2015

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

São Carlos, São Paulo, Brasil,
Diciembre 2015

A handwritten signature in black ink, reading "Diana Esquivel", written over a horizontal line.

Diana Esquivel González

Céd: 114830865

HOJA DE INFORMACIÓN DEL PROYECTO

Datos del estudiante:

Nombre: Diana Esquivel González

Cédula: 1-1483-0865

Carné ITCR: 201121714

Dirección de su residencia: La Uruca, de Central de Mangueras 100 m Este, 125 m Sur, condominio Kandor Casa #9

Teléfono de residencia: 2520-2056

Teléfono celular: 8867-9429

Correo electrónico: dianaesquivel100@hotmail.com

Información del proyecto:

Nombre del Proyecto: Robótica Inteligente: Reconocimiento gestual para determinar cuando un conductor utiliza el celular mientras conduce

Información de la institución:

Nombre: Laboratório de Robótica Móvel- ICMC- Universidade de São Paulo

Actividad Principal: Centro de Investigación

Zona: São Carlos, São Paulo, Brasil

Dirección: Avenida Trabalhador São-carlense, 400 – Centro

Teléfono: +55 (16) 3373-9700

Información del encargado/asesor en la empresa:

Nombre: Dr. Fernando Santos Osorio

Puesto que ocupa: Profesor investigador

Departamento: Laboratório de Robótica Móvel

Profesión: Profesor e investigador **Grado académico:** Doctor

Correo electrónico: fosorio@icmc.usp.br

Resumen

El presente proyecto fue desarrollado en la *Universidade de São Paulo*, sede *São Carlos* en Brasil; específicamente en el “*Laboratório de Robótica Móvel*” perteneciente a la Universidad. Este laboratorio realiza proyectos en el área de navegación autónoma de vehículos y aplicaciones de asistencia al conductor en prevención de accidentes y supervisión de acciones inseguras.

Las acciones inseguras, como hablar por celular mientras se conduce, son una de las principales causas de accidentes en las vías, produciendo una cantidad considerable al año. Aunque varias de estas acciones se han regulado por ley, son difíciles de supervisar y controlar a nivel práctico. Utilizando un sistema automatizado es posible controlar este tipo de acciones peligrosas, para así incrementar la seguridad en las vías.

El proyecto consistió en realizar el diseño e implementación de una aplicación de reconocimiento de gestos para asistencia al conductor, utilizando un sensor 3D Kinect y substracción de fondo, con la cual se puede detectar el gesto de hablar por celular. En la primera parte se realizó una simulación funcional del sistema utilizando el programa V-REP. La implementación se realizó utilizando datos reales del Kinect procesados en el programa Octave para verificar el funcionamiento del programa desarrollado en la simulación. Se utilizaron dos métodos para hacer la detección: análisis de histogramas y análisis de la zona alrededor de la cabeza. Ambos métodos produjeron resultados deseables.

Palabras claves: Kinect; Substracción de fondo; Reconocimiento gestual; Procesamiento de imágenes; Dispositivo móvil.

Abstract

This project was developed at *Universidade de São Paulo*, campus *São Carlos* in Brazil; specifically at the “*Laboratório de Robótica Móvel*”, a lab that belongs to this University. This laboratory develops autonomous navigation projects and applications assisted driving in topics like accident prevention and unsafe actions supervision.

Unsafe actions, like using a cellular phone while driving, are one of the main causes of accidents on the road, producing a considerable amount per year. Even though many of these actions have been regulated by law, they are quite difficult to supervise and control on a practical level. The use of an automated system makes controlling of dangerous actions possible, thus increasing safety on the roads.

This project includes the design and implementation of a gesture recognition application for assisted driving, using a Kinect 3D sensor and background subtraction, which allowed the detection of gestures like using a cellular phone. In the first part, a functional simulation was developed using the V-REP software. The implementation was done using real data obtained from the Kinect sensor and processed using the Octave software to verify the algorithms developed on the simulation code. The software developed used two distinct methods for gesture detection: histogram analysis and examination of the area around the head. Both methods produced desirable results.

Keywords: Kinect; Background Subtraction; Gesture Recognition; Image Processing; Mobile device

Dedicatoria

Dedico este proyecto a mi familia y a Sergio, quienes me han apoyado en cada paso de mi carrera y me han alentado a dar siempre lo mejor de mí misma y a luchar por mis sueños, nunca dejándome darme por vencida.

Agradecimientos

Un especial agradecimiento a los ingenieros Dr. Fernando Santos Osório y Dr. Juan Luis Crespo Mariño, quienes aceptaron ser mis asesores para este proyecto y compartieron su conocimiento para lograr hacer de este proyecto uno exitoso.

ÍNDICE GENERAL

Resumen.....	v
Abstract.....	vi
Dedicatoria.....	vii
Agradecimientos.....	viii
Capítulo 1: Introducción	1
1.1. Contextualización y Motivación	1
1.2. Entorno del proyecto	1
1.3. Definición del problema	4
1.3.1. Generalidades	4
1.3.2. Síntesis del Problema	5
1.4. Enfoque de la solución:	5
1.5. Diagrama Causa-Efecto	6
1.6. Objetivos.....	7
1.6.1. Objetivo General	7
1.6.2. Objetivos Específicos.....	7
1.7. Metodología de trabajo	7
Capítulo 2: Investigación Previa.....	10
2.1. Generalidades de la tecnología de reconocimiento gestual	10
2.2. Estado del arte en reconocimiento gestual	14
2.3. Marco Teórico.....	19
2.3.1. Cinemática Inversa	19
2.3.2. Substracción de fondo	22
2.3.3. Sensor Kinect.....	25
Capítulo 3: Diseño preliminar	27
3.1. Requerimientos:.....	27
3.2. Propuestas de solución.....	28
3.3. Discusión de las soluciones	29
3.4. Especificaciones.....	33
Capítulo 4: Desarrollo de la simulación.....	35
4.1. Sensor Kinect	36
4.1.1. Toma de imagen de fondo.....	38
4.1.2. Substracción de fondo	40
4.1.3. Identificación de la zona de interés.....	41
4.1.4. Identificación del gesto de hablar por celular.....	49
4.2. Figura Humana (Bill)	57

Capítulo 5: Verificación de funcionamiento	60
5.1. Substracción de fondo	61
5.1.1. Pre-procesamiento de las imágenes	61
5.1.2. Obtención de la máscara	64
5.2. Obtención de la zona de interés	66
5.2.1. Método de análisis de histogramas	67
5.2.2. Método de análisis de profundidad	68
5.3. Detección del gesto de hablar por celular	70
5.3.1. Método de análisis de histogramas	71
5.3.2. Método de análisis de zona alrededor de la cabeza	73
5.3.3. Resultados obtenidos	74
Capítulo 6: Discusión de resultados y limitaciones	83
Capítulo 7: Diseño de la estructura de soporte	86
Capítulo 8: Conclusiones y Recomendaciones	88
8.1. Conclusiones	89
8.2. Recomendaciones	89
Referencias	90
Apéndices	95
Apéndice A.1: Histogramas de pixeles para diferentes distancias y posiciones del sensor Kinect	95
A.1.1. Posición frontal del Kinect con respecto a la persona, a una distancia de 0.5 m	96
A.1.2. Posición frontal del Kinect con respecto a la persona, a una distancia de 0.75 m	97
A.1.3. Posición frontal del Kinect con respecto a la persona, a una distancia de 0.9 m	98
A.1.4. Posición del Kinect con una inclinación de 20° respecto a la persona, a una distancia de 0.75 m en Y y 0.2 m en X	99
A.1.5. Posición del Kinect con una inclinación de 30° respecto a la persona, a una distancia de 0.75 m en Y y 0.3 m en X	100
Apéndice A.2.: Profundidades medidas desde el punto máximo de la cabeza hasta el inicio del cuello	101
Apéndice A.3: Planos de construcción para la estructura de soporte del sensor Kinect	103
A.3.1. Placa #1	103
A.3.2. Pieza #2	104
A.3.3. Pieza #3	105
A.3.4. Pieza #4	106

ÍNDICE DE FIGURAS

Figura 1: Vehículo autónomo CaRINA II del Laboratorio de Robótica Móvel de la USP (Tomado de http://irm.icmc.usp.br/web/index.php?n=Port.ProjCarina2Videos)	3
Figura 2: Diagrama Causa-Efecto (Creado por el autor)	7
Figura 3: Reconocimiento de dirección a la que está viendo el conductor utilizando diferentes métodos (Murphy-Chutorian, Doshi y Trivedi, 2007)	15
Figura 4: Ejemplo de los gestos reconocibles en la aplicación de aritmética por reconocimiento de gestos (Ren, Meng, Yuan, y Zhang, 2011)	16
Figura 5: Juego de piedra, papel, tijera, utilizando reconocimiento de los gestos de la mano (Ren, Meng, Yuan, y Zhang, 2011)	16
Figura 6: Entrada de la cámara de color, cámara de profundidad y sensor radar (de arriba hacia abajo) (Molchanov, Gupta, Kim y Pulli, 2015).....	17
Figura 7: Cámara de color, cámara de profundidad y point cloud (Ohn-Barr y Manubhai, 2014).....	17
Figura 8: Demostración del esqueleto dibujado al sostener un celular cerca de la oreja (Solomon y Wang, 2015)	18
Figura 9: Obtención de una figura en movimiento utilizando sustracción de fondo y comparación de imágenes en tiempos diferentes. Tomado de http://www.ics.uci.edu/~dramanan/teaching/cs117_spring13/lec/bg.pdf	24
Figura 10: Componentes internos del sensor Kinect para Windows (Zeng, 2012)	26
Figura 11: Escena de simulación inicial (Creado por el autor en programa V-REP)	36
Figura 12: Campo de visión del Kinect (Tomado de https://msdn.microsoft.com/en-us/library/hh973074.aspx) ..	37
Figura 13: Diagrama de flujo para almacenar los datos del fondo (Creado por el autor en Visio)	38
Figura 14: Imagen de fondo en la simulación (Creado por el autor en V-REP).....	39
Figura 15: Diagrama de flujo para la sustracción de fondo (Creado por el autor en Visio)	41
Figura 16: Diagrama de flujo para encontrar el punto máximo de la cabeza. (Creado por el autor en Visio).....	44
Figura 17: Diagrama de flujo para encontrar el punto de inicio del cuello utilizando el método de histogramas. (Creado por el autor en Visio).....	45
Figura 18: Diagrama de flujo para encontrar el punto de inicio del cuello y los extremos laterales de la cabeza utilizando el método de medición de profundidades (Creado por el autor en Visio)	47
Figura 19: Resultados al calcular los puntos de inicio del cuello y final de la cabeza utilizando el método de número de píxeles por fila (Creado por el autor en programa V-REP)	48
Figura 20: Resultados al calcular el punto de inicio del cuello, el extremo superior de la cabeza, y los extremos laterales de la cabeza utilizando el método de diferencia de profundidades (Creado por el autor en el programa V-REP).....	49
Figura 21: Diagrama de flujo para detectar el gesto de hablar por celular utilizando el método de número de píxeles por fila (Creado por el autor en Visio)	52
Figura 22: Diagrama de flujo para la detección del gesto de hablar por celular utilizando el método de análisis de franjas laterales a la cabeza (Creado por el autor en Visio)	53
Figura 23: Resultado de detección de gesto por el método de análisis de conteo de píxeles por fila (Creado por el autor en V-REP)	54
Figura 24: Resultado de activación de alarma al reconocer el gesto por más de 5 segundos seguidos (Creado por el autor en V-REP)	55
Figura 25: Resultado de detección de regreso a posición neutral después de haber detectado el gesto (Creado por el autor en V-REP)	56
Figura 26: Posición inicial y final de Bill al presionar el botón que define la posición del brazo al hablar por celular (Creado por el autor en V-REP)	59
Figura 27: Dos imágenes distintas con información del fondo (Osório&Berri, 2015).....	62
Figura 28: Promedio de 20 imágenes con información del fondo (Realizado por el autor en Octave)	63
Figura 29: Promedio de las imágenes con la persona en posición neutral (Creado por el autor en Octave)	63
Figura 30: Recorte de la imagen de fondo y la imagen con la persona en posición neutral (Creado por el autor en Octave).....	64
Figura 31: Máscara resultado de realizar sustracción de fondo (Creado por el autor en Octave)	66
Figura 32: Punto máximo de la cabeza y punto de inicio del cuello graficados sobre la máscara de la figura humana (Creado por el autor en Octave).....	68

Figura 33: Punto máximo de la cabeza, extremos izquierdo y derecho de la cabeza y punto de inicio del cuello graficados sobre la máscara de la figura humana (Creado por el autor en Octave)	70
Figura 34: Substracción de fondo para diferentes casos (a) persona en posición neutral, b) con las manos al frente, c) utilizando el celular con la mano derecha, d) utilizando el celular con la mano izquierda) (Creado por el autor en Octave)	72
Figura 35: Imágenes pertenecientes a la serie con la persona en posición neutral utilizada para probar el algoritmo (Osório & Berri, 2015)	75
Figura 36: Resultado de correr el programa DetecciónHist (método 1) con una serie de imágenes en posición neutral (Creado por el autor en Octave)	76
Figura 37: Resultado de correr el programa DetecciónProf (método 2) con una serie de imágenes en posición neutral (Creado por el autor en Octave)	76
Figura 38: Imágenes pertenecientes a la serie con la persona en posición neutral y sus brazos hacia el frente (Osório & Berri, 2015).....	77
Figura 39: Resultado de correr el programa DetecciónHist (método 1) con una serie de imágenes con la persona en posición neutral y con los brazos hacia el frente (Creado por el autor en Octave).....	78
Figura 40: Resultado de correr el programa DetecciónProf (método 2) con una serie de imágenes con la persona en posición neutral y con los brazos hacia el frente (Creado por el autor en Octave).....	79
Figura 41: Imágenes pertenecientes a la serie con la persona hablando por celular con la mano derecha (Osório & Berri, 2015)	79
Figura 42: Resultado de correr el programa DetecciónHist (método 1) con una serie de imágenes con la persona hablando por celular con la mano derecha (Creado por el autor en Octave).....	80
Figura 43: Resultado de correr el programa DetecciónProf (método 2) con una serie de imágenes con la persona hablando por celular con la mano derecha (Creado por el autor en Octave).....	81
Figura 44: Imágenes pertenecientes a la serie con la persona hablando por celular con la mano izquierda (Osório & Berri, 2015)	82
Figura 45: Resultado de correr el programa DetecciónHist (método 1) con una serie de imágenes con la persona hablando por celular con la mano izquierda (Creado por el autor en Octave)	83
Figura 46: Resultado de correr el programa DetecciónProf (método 2) con una serie de imágenes con la persona hablando por celular con la mano izquierda (Creado por el autor en Octave)	83
Figura 47: Camión autónomo desarrollado por el LRM (Laboratório de Robótica Móvel, 2015).....	87
Figura 48: Estructura de soporte para sostener el sensor (con el Kinect en su posición y sin el Kinect) (Creado por el autor en CREO Parametric) Kinect tomado de (Sintonen, 2014)	88
Figura A.1.1: Histograma de pixeles para el sensor en posición frontal, a una distancia de 0.5m (Creado por el autor en Excel)	96
Figura A.1.2: Histograma de pixeles para el sensor en posición frontal, a una distancia de 0.75m (Creado por el autor en Excel)	97
Figura A.1.3: Histograma de pixeles para el sensor en posición frontal, a una distancia de 0.9m (Creado por el autor en Excel)	98
Figura A.1.4: Histograma de pixeles para el sensor en posición de 20° de inclinación, a una distancia de 0.75m en Y y 0.2m en X (Creado por el autor en Excel)	99
Figura A.1.5: Histograma de pixeles para el sensor en posición de 30° de inclinación, a una distancia de 0.75m en Y y 0.3 m en X (Creado por el autor en Excel)	100
Figura A.3.1: Placa #1 para posicionamiento del sensor Kinect (Creado por el autor en CREO Parametric)	103
Figura A.3.2: Pieza #2 para soporte de la placa #1 (Creado por el autor en CREO Parametric).....	104
Figura A.3.3: Pieza #3 para unión entre pieza #2 y pieza superior (Creado por el autor en CREO Parametric)..	105
Figura A.3.4: Pieza superior para sostener la estructura en el techo del carro (Creado por el autor en CREO Parametric).....	106

ÍNDICE DE TABLAS

Tabla 1:	Metodología de trabajo (Creada por el autor)	7
Tabla 2:	Tabla morfológica para diseño de la solución (Creada por el autor).....	28
Tabla 3:	Descripción de las variables para las ecuaciones 4.5 a 4.7.....	43
Tabla A.2.1:	Valores de profundidad para las filas 30 a 48 con el sensor posicionado a una distancia de 0.5m	101
Tabla A.2.2:	Valores de profundidad para las filas 31 a 42 con el sensor posicionado a una distancia de 0.75m	102

Capítulo 1: Introducción

1.1. Contextualización y Motivación

Los avances tecnológicos han permitido que el ser humano progrese en su desarrollo a lo largo de los años, provocando que las tareas que realiza sean cada vez más eficientes y más seguras. Durante los últimos años, la robótica y la automatización han jugado un papel muy importante en el desarrollo de la tecnología, ya que gracias a ellas se han podido llevar a cabo nuevas aplicaciones que simplifican las tareas de la industria o del quehacer diario; las realiza en menor tiempo, aumenta la producción, e incrementa la seguridad para el operario. Este último punto es una de las principales motivaciones de este trabajo, ya que al aumentar la seguridad en el uso de la tecnología, se protege el componente principal en cualquier área, el factor humano. Es por ello que se debe invertir en la investigación y desarrollo de nuevas aplicaciones que permitan mejorar los mecanismos de seguridad, automatizando procesos que se ven afectados por el error humano, o monitoreando tareas de los usuarios para evitar acciones inseguras que puedan provocar accidentes.

Gran parte de los accidentes que suceden, ya sea en la industria o en las carreteras, se deben a error humano. Este factor puede eliminarse con la automatización, ya que las máquinas y las computadoras no cometen errores y no sufren de características humanas como el cansancio y la distracción. Por esta razón, este proyecto se enfoca en la investigación en el área de la automatización para el desarrollo de una aplicación que permita detectar acciones inseguras a la hora de conducción en carreteras, de modo que pueda reducirse drásticamente la cantidad de accidentes que suceden por esta causa.

1.2. Entorno del proyecto

La *Universidade de São Paulo* es una institución educativa fundada en 1934 en el estado de São Paulo, en Brasil. Hoy, es la universidad más grande y prestigiosa del país, con 7 sedes distintas distribuidas en diferentes ciudades del estado. Su trayecto la ha convertido además en la universidad número uno de Latinoamérica, y se mantiene entre las primeras 50 posiciones del

ranking mundial. (Manzano, 2014) El campus de São Carlos es reconocido en el área de la ingeniería y la tecnología y ha ganado su prestigio por la cantidad y calidad de avances tecnológicos que desarrolla anualmente. Empresas nacionales e internacionales se concentran en esta ciudad debido a la presencia de la universidad y su aporte de profesionales altamente calificados. Además, esta ciudad cuenta con la mayor cantidad per cápita de profesionales con doctorado en el país.

En los últimos años, la USP (*Universidade de São Paulo*) ha estado trabajando en la navegación autónoma e inteligente de vehículos y en el desarrollo de aplicaciones inteligentes para mejorar la seguridad de conductores y pasajeros en su Laboratorio de Robótica Móvel, en donde tienen varios ejemplares de automóviles a los que se les realiza modificaciones para que puedan manejarse de manera autónoma y detectar o evitar situaciones de peligro. Su proyecto más reciente se basa en un camión autónomo que es capaz de navegar por las vías de la universidad sin la necesidad de un conductor, siendo capaz de detectar obstáculos y reconocer el camino navegable, siguiendo una trayectoria indicada por el usuario. Se utilizan sensores muy avanzados de proximidad, sistemas de visión, sensores ultrasónicos, radares, y GPS de alta precisión para este propósito. El objetivo es poder utilizarlos en algunos años para la navegación en ambientes urbanos sin la necesidad de un conductor humano.

El proyecto principal que tiene la universidad en este tema es el “Projeto CaRINA”, el cual combina la navegación robótica, visión computacional, fusión de sensores y control automático para crear un automóvil completamente autónomo. Dicho automóvil ya realizó una prueba en el año 2013 en las vías públicas de São Carlos, pudiendo identificar obstáculos, caminos de la carretera, rotondas, intersecciones, reductores de velocidad, señales de tránsito y peatones. Con esto consiguió navegar de manera 100% autónoma. CaRINA se convirtió en el primer vehículo autónomo de América Latina en ser utilizado en las vías públicas (Laboratório de Robótica Móvel, 2015). Sin embargo, aún se deben realizar más modificaciones y pruebas para que el automóvil sea completamente seguro, y cumpla con regulaciones estatales necesarias para poderse comercializar y utilizar a gran escala. Para la navegación autónoma en el laboratorio, se realizan de forma separada los distintos componentes del automóvil (reconocimiento de vías, planeamiento de rutas, evasión de obstáculos, identificación de otros

automóviles en la vía, lectura de señales, etc) y posteriormente se integran para poder hacer un sistema funcional.



Figura 1: Vehículo autónomo CaRINA II del Laboratorio de Robótica Móvel de la USP (Tomado de <http://irm.icmc.usp.br/web/index.php?n=Port.ProjCarina2Videos>)

Paralelo a la navegación autónoma, en el laboratorio se realizan investigaciones que puedan mejorar los sistemas de seguridad en vehículos existentes, de modo que se puedan automatizar procesos que generalmente se ven afectados por el error humano. En esta área se desarrollan aplicaciones utilizando sensores 3D y cámaras de alta resolución para monitorear el comportamiento del conductor de modo que se detecten fuentes de distracción y acciones que se consideren inseguras. De esta forma se puede asistir al conductor en su labor para evitar accidentes. Este proyecto pretende agregar un componente adicional en esta área, que puede utilizarse dentro del vehículo para identificar acciones inseguras del conductor, en este caso la acción de hablar por celular durante la conducción. Se pretende explorar el uso de los sensores 3D con tecnología “point clouds” (en este caso un Kinect) para la detección del cuerpo humano y sus partes, e identificación de los gestos, y poder utilizar esta información para el control de mecanismos de seguridad del vehículo que se utiliza. Para este caso, se pretende activar una alarma para alertar al conductor del peligro que causa la acción que está realizando.

1.3. Definición del problema

1.3.1. Generalidades

La conducción inapropiada e insegura de automóviles genera millones de accidentes al año en todo el mundo. Según la Organización Mundial de la Salud, alrededor de 1.3 millones de personas mueren cada año en accidentes de tránsito (Organización Mundial de la Salud, 2009). A esta problemática se suma la inseguridad por secuestros, robos y demás acontecimientos que puedan suceder en las vías o que puedan ser realizados por conductores de vehículos de transporte público. La asistencia al conductor por medio de la tecnología parece una solución viable para esta problemática, pero debe ser sometida a una serie de pruebas para asegurar tanto la seguridad del sistema como la facilidad de uso para el usuario. Además, debe desarrollarse de manera robusta para asegurar el funcionamiento con cualquier usuario y ante una variedad de circunstancias y situaciones.

Un problema concreto relacionado a esta problemática es la distracción del conductor del vehículo. Esta acción insegura provoca que se pierda el enfoque a la hora de conducir e incrementa de manera importante la probabilidad de sufrir un accidente; poniendo en riesgo la vida no solo del conductor sino de las personas que viajan dentro del vehículo y las personas en los vehículos alrededor o los peatones. Una acción muy común que ha sido regulada por la ley pero que sigue siendo un problema es la de utilizar dispositivos móviles, como el celular, mientras se está conduciendo. Este acto es peligroso debido a que desvía la atención del conductor y evita que esté alerta a las situaciones de riesgo que suceden a su alrededor. A pesar de ser un acto penalizado por ley, es difícil poder detectar cuando se da una infracción y aún más difícil poder probar que efectivamente se estaba haciendo uso del dispositivo en el momento en que la persona iba conduciendo. La cantidad de personas que han sido amonestadas por quebrantar esta ley en comparación con la cantidad que en efecto hace uso del dispositivo móvil mientras maneja es muy reducida, a pesar de que el incumplimiento de esta norma representa una de las principales causas en accidentes de tránsito.

A pesar de que se han desarrollado aplicaciones automatizadas para detectar acciones inseguras y para prevenir accidentes de tránsito, aún no se cuenta con una solución comercial y

confiable para detección del acto de hablar por celular mientras se conduce. Muchos de los proyectos que se han trabajado en este tema (los cuales se resumen posteriormente en la sección de revisión bibliográfica) lo ven como un problema secundario al que abordan de manera superficial, por lo que no se ha logrado encontrar una solución viable al problema que se plantea.

1.3.2. Síntesis del Problema

Las acciones inseguras y las distracciones a la hora de conducir un vehículo, especialmente utilizar dispositivos móviles como el celular, provocan una considerable cantidad de accidentes de tránsito y presentan un problema ya que son difíciles de detectar sin utilizar un sistema automatizado.

1.4. Enfoque de la solución:

La Mecatrónica, al combinar distintas ramas de la tecnología, permite realizar soluciones muy completas y efectivas a problemáticas como la que se plantea. El propósito del proyecto es poder crear una aplicación para automóviles que utilice los principios de la automatización y el procesamiento de imágenes y que sea capaz de solventar un problema de inseguridad para usuarios de vehículos.

Para la solución del problema, se trabajó con cámara de visión 3D o Kinect en donde se utilizó la tecnología “point clouds” y la substracción de fondo, para extraer los objetos de interés y poder analizarlos. Inicialmente, se realizó una simulación de la aplicación utilizando el programa V-REP que permite hacer simulaciones muy exactas de los sistemas con los que se trabajan.

Para poder cumplir el objetivo, primero debió desarrollarse un algoritmo para poder restar la información que compone el “fondo” de la imagen, para así poder trabajar únicamente con la información de interés: el cuerpo humano. Luego se desarrolló una aplicación en la cual se logró identificar la figura humana y sus partes para así poder determinar si el conductor está utilizando el celular mientras maneja, lo cual activa una alarma. Una vez que se tuvo la simulación funcional, se trabajó con los sensores en físico, lo cual permitió la verificación de los datos para la solución planteada. Finalmente, se diseñó una estructura para ser colocada en el

vehículo que se tiene disponible, que permite posicionar el sensor de modo que se pueda utilizar la aplicación diseñada.

El desarrollo de la aplicación integra las diferentes áreas de conocimiento para obtener un resultado satisfactorio. Es por esto que el proyecto a desarrollar es afín a los objetivos de la Ingeniería Mecatrónica y ofrece una solución novedosa y factible a un problema actual y relevante.

1.5. Diagrama Causa-Efecto

El siguiente diagrama de Ishikawa resume las principales causas que llevan a la inseguridad en carretera y en uso de automóviles. Estos se pueden resumir en tres categorías: Acciones Inseguras, Situaciones de peligro y Error Humano. Las acciones inseguras y el error humano llevan a accidentes vehiculares en carretera, mientras que las situaciones de peligro se relacionan a secuestros y robo de vehículos. No todas las causas se pueden evitar con la aplicación que se quiere realizar, pero varias de ellas se podrían solventar con el uso de este tipo de tecnología.

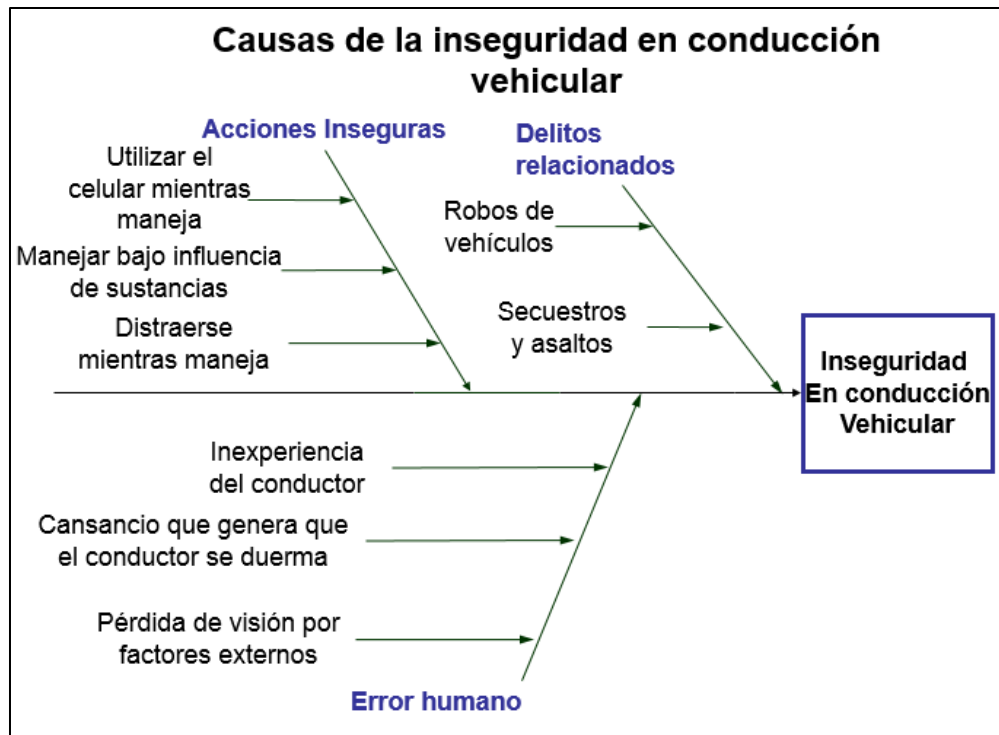


Figura 2: Diagrama Causa-Efecto (Creado por el autor)

1.6. Objetivos

1.6.1. Objetivo General

Realizar el diseño e implementación de una aplicación de reconocimiento gestual utilizando un sensor de profundidad, para determinar si un conductor se encuentra utilizando el celular mientras conduce, para ser utilizado en aplicaciones de seguridad.

Indicador: Simulación y verificación de datos que cumplen con los objetivos planteados

1.6.2. Objetivos Específicos

- Diseñar una simulación de la aplicación que se quiere realizar
 - Indicador: Diseño de la aplicación con alcances, requerimientos y especificaciones que se deben cumplir
- Realizar la simulación del sistema y los sensores utilizando el programa V-REP para aplicaciones en robótica
 - Indicador: Simulación funcionando
- Diseñar un soporte para posicionar el sensor en una implementación física del sistema simulado
 - Indicador: Diseño del soporte con planos de construcción
- Realizar un prototipo del sistema funcionando con los sensores reales
 - Indicador: Pruebas exitosas realizadas con el sensor que respalden el funcionamiento de la aplicación creada

1.7. Metodología de trabajo

Tabla 1: Metodología de trabajo (Creada por el autor)

Objetivo	Actividades	Herramientas	Resultados Esperados
----------	-------------	--------------	----------------------

<p>Diseñar una simulación de la aplicación que se quiere realizar</p>	<ul style="list-style-type: none"> - Investigar acerca de las propiedades y el manejo apropiado de los sensores 3D que utilizan la tecnología “point cloud” - Realizar un planeamiento de la aplicación que se va a realizar con respecto a los objetivos del laboratorio y las capacidades de los sensores. - Plantear los alcances, requerimientos y especificaciones de la aplicación que se va a realizar 	<ul style="list-style-type: none"> - Manual de uso del programa V-REP y libro de programación en lenguaje LUA - Material educativo proporcionado por el profesor guía 	<ul style="list-style-type: none"> - Identificar las herramientas disponibles y sus limitaciones para el desarrollo de la aplicación - Plan de trabajo con especificaciones de la simulación que se va a desarrollar.
<p>Realizar la simulación del sistema y los sensores utilizando el programa V-REP para aplicaciones en robótica</p>	<ul style="list-style-type: none"> - Crear un entorno simulado que se asemeje a una situación real de conducción de vehículos - Identificar las propiedades de los elementos que componen el ambiente simulado - Definir los parámetros necesarios para poder realizar lecturas apropiadas en la simulación 	<ul style="list-style-type: none"> - Manual de usuario de V-REP - Modelos importados predefinidos (personas, sillas, sensor Kinect, elementos del ambiente) 	<ul style="list-style-type: none"> - Lograr realizar una substracción del fondo que permita trabajar únicamente con los pixeles que pertenecen a una persona en la escena - Segmentar el área de interés de donde se va a realizar la lectura - Identificar los gestos que corresponden a una persona hablando por

			celular
Diseñar un soporte para posicionar el sensor en una implementación física del sistema simulado	<ul style="list-style-type: none"> - Identificar las características del lugar donde se debe posicionar el sensor en una situación real - Realizar mediciones para el diseño del soporte - Elaborar planos de construcción para la pieza que se debe construir 	<ul style="list-style-type: none"> - Acceso a los vehículos disponibles del laboratorio - Herramientas de diseño CAD 	<ul style="list-style-type: none"> - Obtener un diseño apropiado para construcción de un soporte para el sensor Kinect, que permita el desempeño adecuado de la aplicación realizada
Realizar un prototipo del sistema funcionando con los sensores reales	<ul style="list-style-type: none"> - Utilizar el sensor real para hacer mediciones en un ambiente que se asemeje a la aplicación real - Verificar el funcionamiento adecuado de la aplicación realizada - Realizar ajustes si se obtienen resultados no deseables 	<ul style="list-style-type: none"> - Sensor Kinect y el programa de Microsoft Kinect Studio SDK - Ambiente controlado similar al asiento del conductor de un automóvil 	<ul style="list-style-type: none"> - Mediciones que comprueben el funcionamiento adecuado de la aplicación desarrollada - Prototipo funcional en un ambiente controlado

Capítulo 2: Investigación Previa

2.1. Generalidades de la tecnología de reconocimiento gestual

El reconocimiento gestual es la tecnología encargada de interpretar gestos humanos por medio de sensores y sistemas computacionales. Se utilizan algoritmos matemáticos para poder identificar los gestos e interpretar los datos que se reciben. Algunas de las áreas de aplicación para el reconocimiento gestual son la lectura de lenguaje de señas, interpretación de la postura, estudio de la proxémica (organización del espacio en la comunicación), lectura de señas de la mano, e interpretación de las emociones (gestos faciales). El propósito de emplear esta tecnología es poder estudiar el comportamiento humano, de modo que se pueda crear interfaces entre persona y máquina que sea más “natural” para el individuo. De este modo, se elimina la necesidad de dispositivos (como el “mouse” y el teclado) para interactuar con una computadora, haciendo que el contacto con la tecnología se torne más amigable para el usuario, convirtiéndola así en algo más atractivo y más efectivo. (Braffort, Gherbi, Gibet, Richardson, & Teil, 1999)

Hay diversos pasos que se deben seguir para realizar un adecuado reconocimiento de gestos. El primero es tener un modelo de gestos, para el cual se debe estudiar las características del gesto que se quiere reconocer. Este modelo depende de la aplicación que se quiera realizar con dicho reconocimiento. Una vez que se tiene el modelo, se debe realizar un análisis para definir los parámetros que se tomarán en cuenta. Cuando se tienen los parámetros bien definidos, se procede a la etapa de reconocimiento, en donde se reconocen los distintos gestos que fueron predefinidos. Finalmente, se utilizan los gestos reconocidos para alguna aplicación o sistema. Para este último paso se le da un uso concreto al reconocimiento de gestos, con el cual se llevará a cabo una acción que está ligada a la información analizada. (Premaratne, 2014)

La complejidad en reconocimiento gestual se debe a la amplia variedad de gestos que pueden existir, y al hecho de que un mismo gesto no producirá la misma serie de imágenes en

dos ocasiones distintas. Es por ello que se debe trabajar con modelos de gestos que puedan definir claramente lo que un gesto específico significa en términos de posiciones y movimientos de las manos y de expresiones faciales. Para simplificar la tarea de discernir entre una infinidad de gestos, se puede dividir en categorías. Distintos autores han definido sus propias categorías de acuerdo a las necesidades que se les presenta. Sin embargo, una de las más adecuadas para aplicaciones de HCI ("*Human-Computer Interactions*" o Interacciones Hombre-Máquina) es dividir los gestos en acciones voluntarias e involuntarias, de modo que se estudien únicamente las acciones voluntarias para identificar los gestos. Las acciones voluntarias a su vez se pueden clasificar en gestos manipulativos y gestos comunicativos. Los gestos manipulativos se utilizan para aplicaciones de control, como por ejemplo indicar rotación o hacer una señal con el dedo para activar un sistema; mientras que los gestos comunicativos son los que asisten en la comunicación. Estos últimos son los que se utilizan para interpretación de emociones y de estados de ánimo, y generalmente se acompañan del habla. Los gestos comunicativos también pueden dividirse en actos y símbolos. Los símbolos son representaciones de algún objeto o acción, mientras que los actos se relacionan directamente con la interpretación del movimiento. Los actos pueden imitar y representar comportamientos, por lo que son utilizados para aplicaciones en donde se estudia el quehacer humano y las acciones comunicativas. (Pavlovic, Sharma, & Huang, 1997)

Inicialmente, el reconocimiento de gestos de la mano se dio por medio del uso de una interfaz basada en un guante de recepción de datos. Esta interfaz cuenta con una serie de sensores táctiles posicionados en la punta de los dedos o en las articulaciones de las manos, que se conectan a una computadora por medio de cables. En las articulaciones también se posicionan interruptores táctiles o sensores resistivos para determinar si la mano se encuentra abierta o cerrada, o si alguno de los dedos se encuentra doblado. Los resultados medidos por el sistema, se mapean para identificar ciertos gestos y se analizan por medio de una computadora. La ventaja de este tipo de dispositivo es que no se requiere pre-procesamiento para hacer la lectura de datos, y se puede realizar con muy baja potencia de procesamiento. Sin embargo, resulta poco eficaz en la lectura de gestos "naturales" ya que incomodan al usuario y lo restringen de la fluidez de movimiento. La alternativa a estos guantes con cables, es utilizar guantes con colores. Éstos, a diferencia de los primeros, no contienen cables de ningún tipo ya que no utilizan sensores, sino

que se realiza la lectura por medio de procesamiento de imágenes que interpreta la posición de los colores (uno diferente para cada dedo) para reconocer los gestos de la mano.

En 1977 se desarrolló el “Sayre Glove” el cual utiliza tubos finos en cada dedo (Premaratne, 2014), en los cuales se posiciona una fuente de luz LED en un extremo y un fotodiodo en el otro, pudiendo medir el grado de flexión del dedo dependiendo del voltaje medido en el diodo. Posterior a éste, se comenzaron a desarrollar guantes con múltiples sensores, en donde se incluyen sensores táctiles para determinar si los dedos tocan otras partes de la mano, sensores de flexión de los dedos, sensores de inclinación para determinar la orientación de la mano, y sensores inerciales para determinar la inclinación y flexión de la muñeca. Este tipo de dispositivos permitió una lectura mucho más precisa de los gestos realizados con la mano. Los guantes más recientes utilizan acelerómetros para determinar la orientación de cada dedo y de la palma de la mano, pudiendo así realizar lecturas muy exactas. Este tipo de guantes aún se utiliza para investigación en biomecánica, control de robots, industria de videojuegos, animación y rehabilitación. (Premaratne, 2014)

Como alternativa al uso de guantes con sensores o colores para detección de gestos, surgió la opción de realizar la detección por medio de procesamiento de imágenes. Para ello, se utiliza la información proveniente de una cámara y se analiza los datos de color, profundidad, posición, o intensidad para determinar distintas situaciones que se pueden estar dando. Este tipo de lectura de datos permite una amplia variedad de aplicaciones que se pueden realizar para análisis de los datos de la imagen recibida. Entre ellos, se pueden reconocer gestos para aplicaciones como: control de dispositivos por medio de gestos faciales, interacción en realidad virtual o interacción con robots, lectura de lenguaje de señas u otros símbolos, control remoto por medio de gestos, etc. Generalmente, las cámaras utilizadas para este tipo de aplicaciones incluyen la cámara de profundidad, que mide el tiempo de retorno de una señal luminosa, o una cámara estéreo, que mide la relación entre imágenes tomadas desde dos puntos distintos cuya distancia de separación es conocida. (Srilatha & Saranya, 2014) En el reconocimiento por medio de lectura de imágenes, la interpretación de los datos de la imagen, realizada por medio del software, debe ser bien desarrollada para poder producir resultados adecuados, ya que no se cuenta con un modelo estándar que se pueda utilizar para realizar reconocimientos efectivos.

El reconocimiento gestual se ve rodeado de muchas dificultades, debido a la infinita variedad de gestos reconocibles y a la poca uniformidad en relación a un mismo gesto en instantes diferentes. La información de color (piel, ojos, cabello), la posición de las partes del cuerpo, las dimensiones de la cara y de la mano, las proporciones faciales, cambian mucho entre diferentes personas, por lo que realizar una aplicación que sea capaz de reconocer un gesto para cualquier usuario es bastante complejo. Es por ello que este tema sigue estando en investigación y desarrollo. Temas como el estudio de emociones se ve afectado por la variedad de posibles resultados obtenibles, ya que además de los estados de ánimo más tradicionalmente estudiados, como felicidad, enojo, desagrado, tristeza, miedo, y sorpresa, se tiene una gran cantidad de expresiones faciales espontáneas que requieren ser reconocidas para un estudio más realista del comportamiento humano. Es por ello que se han desarrollado bases de datos con información recolectada acerca de los posibles resultados que se pueden obtener, las cuales permiten realizar aplicaciones más robustas y aplicables a la realidad. (Zhang, y otros, 2013) Esto contribuye a que se dé un avance en los trabajos realizados, de modo que no se requiera estudiar cada posible resultado a la hora de hacer una nueva aplicación, sino que se pueda usar la información disponible para ampliar en el tema o para aportar una nueva función.

Un problema importante en el área de reconocimiento de gestos es las variaciones en iluminación, poses, expresiones, obstrucciones y movimientos. Una aplicación realizada en un ambiente controlado funciona bastante bien. Sin embargo, no puede utilizarse en aplicaciones reales en donde se estudie el comportamiento natural de las personas. Para poder realizar este tipo de sistemas más complejos, se debe poder contar con modelos matemáticos más avanzados, que logren realizar segmentaciones y reconocimientos en ambientes difíciles o en situaciones cotidianas, sin necesidad de tener que realizar poses forzadas. Para ello se puede recurrir a herramientas como redes neuronales y sistemas de aprendizaje con inteligencia artificial, que logren tomar la información y utilizarla para “aprender”, de forma que la aplicación se torne más eficiente con cada caso encontrado. (Yan, Zhang, Lei, Yi, & Li, 2013)

Además de esto, algunas aplicaciones que realizan reconocimiento de gestos incluyen partes del cuerpo que son muy difíciles de detectar, como por ejemplo la mano, ya que poder discernir entre los dedos es bastante complejo. Asimismo, un modelo realista de una mano puede incluir hasta 35 articulaciones, por lo que se torna sumamente complejo a la hora de poder

identificar cada una de ellas. Por esta razón, cuando se trabaja con la mano se debe utilizar información en 3D (con datos de profundidad) ya que un análisis en dos dimensiones daría resultados deficientes debido a la uniformidad en características y a la variedad de propiedades. Para ello se debe hacer un estudio de las propiedades 3D de la mano o utilizar bases de datos existentes para este propósito. (Zhang, Yang, & Tian, 2013) Utilizar la información de color o de intensidad es difícil debido a que es bastante uniforme a lo largo de la mano y gran parte de las poses de ésta se ven afectadas por la obstrucción de ciertas secciones, lo cual causa que no se pueda obtener información adecuada con el uso de una cámara. Como solución, se requiere poder inferir la posición de las partes de la mano, de acuerdo con modelos pre-establecidos y con análisis adecuado de la información. Una aplicación realizada en este ámbito debe ser resistente a ruidos, de modo que un cambio leve en la pose no afecte los resultados de lectura, y debe realizar una segmentación robusta, que evite detecciones erróneas. (Thippur, Ek, & Kjellstrom, 2013)

2.2. Estado del arte en reconocimiento gestual

El reconocimiento gestual se ha utilizado en los últimos años para una variedad de aplicaciones de control y seguridad. Se utiliza la substracción de fondo y la segmentación el cuerpo humano para poder determinar los diferentes gestos que realiza una persona y así tomar una decisión de accionamiento. La ventaja de una aplicación como esta es que permite bastante flexibilidad a la hora de realizar lectura de datos pero debe ser programado de forma que tome en consideración todos los diferentes escenarios que se puedan presentar.

Por ejemplo, en una publicación de la IEEE redactada por Murphy-Chutorian, Doshi y Trivedi en 2007, se expone el uso de un sistema de reconocimiento gestual que determina la dirección a la que el conductor está viendo, basado en la posición de la cabeza. Esto se realiza con el fin de poder monitorear cuando el conductor se encuentra distraído. Esta aplicación, al combinarse con los sistemas de seguridad como sensores láser y cámaras estéreo permitiría alertar al conductor cuando haya una situación de peligro de la que no se ha percatado, como por ejemplo al girar a la derecha mientras ve hacia la izquierda y se atraviesa un peatón. El sistema permitiría que el vehículo detecte al peatón y reconozca que el conductor no lo ha visto, para luego tomar la decisión de alertarlo del peligro. (Murphy-Chutorian, Doshi, & Trivedi, 2007)

Aplicaciones como esta son útiles para incrementar la seguridad en el vehículo, así sea autónomo o no, para poder reducir la cantidad de accidentes de tránsito que se puedan provocar.

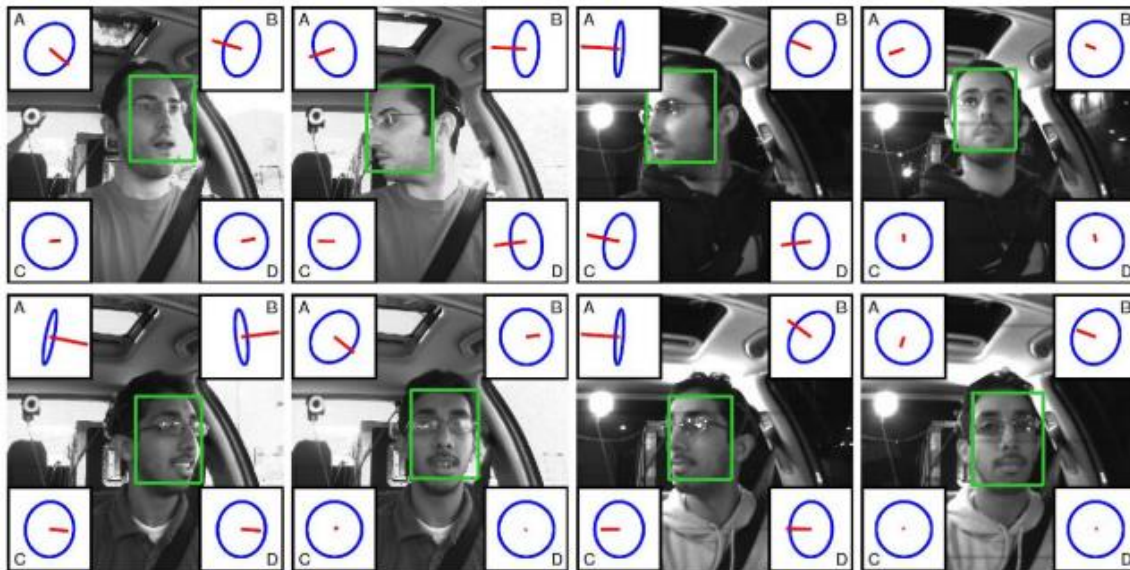


Figura 3: Reconocimiento de dirección a la que está viendo el conductor utilizando diferentes métodos (Murphy-Chutorian, Doshi y Trivedi, 2007)

El reconocimiento de gestos utilizando Kinect ha sido una de las aplicaciones más exploradas en los últimos años, debido a la simplicidad de uso del dispositivo y a la riqueza de información que se puede recibir. El Kinect permite la lectura de dos tipos de datos: información de profundidad y cámara de color. Esto incrementa la cantidad de aplicaciones que se le puede dar al dispositivo, ya que se puede combinar lo que se obtiene en ambas cámaras para hacer lecturas más precisas. La cámara de profundidad permite hacer segmentación de objetos grandes como lo es el cuerpo humano, mientras que la cámara de color facilita la detección de segmentos más pequeños como la mano. En la Universidad Tecnológica de Nanyang, se publicó un trabajo de Ren, Meng y Yuan el cual utiliza el Kinect para poder hacer lectura de las manos e identificar gestos de dos tipos: cálculos aritméticos y el juego de piedra, papel y tijera. Para ello realizan una substracción del fondo y luego un reconocimiento de la mano. Finalmente, determinan el gesto que la mano está realizando. (Ren, Meng, Junsong, & Zhang, 2011)

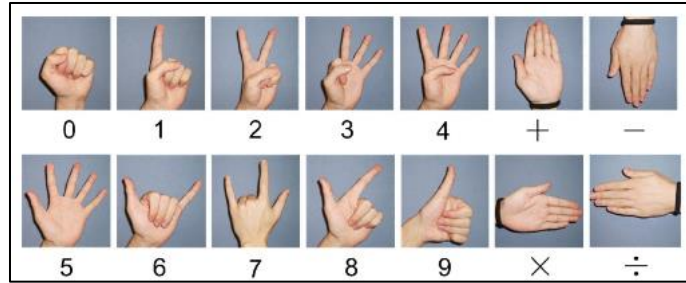


Figura 4: Ejemplo de los gestos reconocibles en la aplicación de aritmética por reconocimiento de gestos (Ren, Meng, Yuan, y Zhang, 2011)

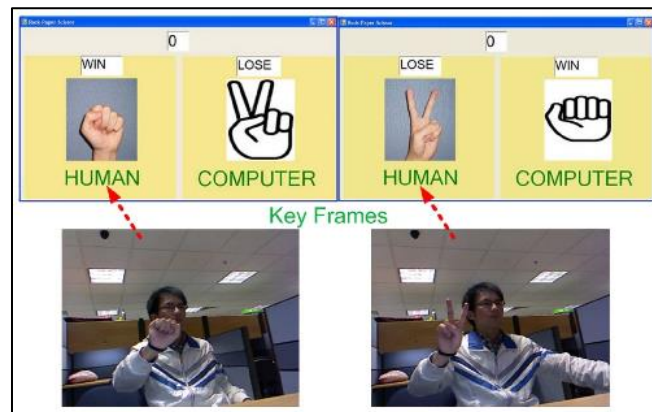


Figura 5: Juego de piedra, papel, tijera, utilizando reconocimiento de los gestos de la mano (Ren, Meng, Yuan, y Zhang, 2011)

Otro trabajo importante en el área de reconocimiento gestual se realizó en Santa Clara, California en el Laboratorio de Investigación NVIDIA, en donde Molchanov, Gupta, Kim y Pulli desarrollaron una aplicación para reconocimiento de gestos en un automóvil utilizando fusión de sensores. En el informe argumentan que, al combinar una cámara de color, una cámara de profundidad y un sensor Radar se puede tener un sistema más robusto que permita que funcione ante diferentes condiciones de iluminación. Utilizando una red neuronal se combina la información de los tres sensores utilizados para poder identificar entre diez gestos distintos. Esta aplicación fue desarrollada para uso en automóviles, permitiéndole al usuario utilizar gestos de la mano para controlar diferentes funciones en el interior del automóvil, eliminando las distracciones que se generan al utilizar pantallas táctiles o botones. (Molchanov, Gupta, Kim, & Pulli, 2015)

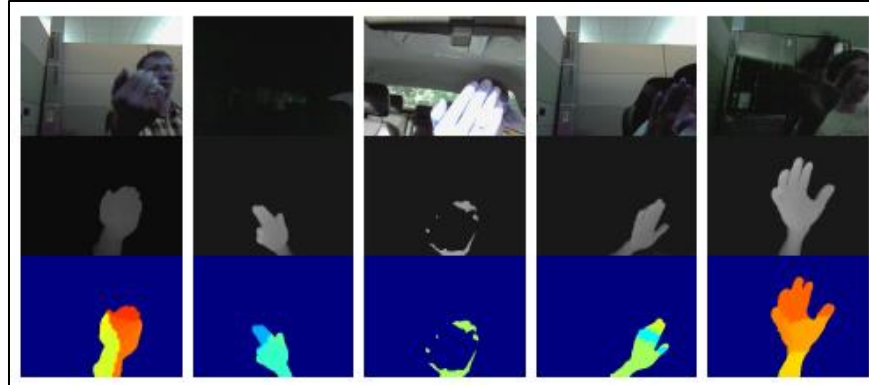


Figura 6: Entrada de la cámara de color, cámara de profundidad y sensor radar (de arriba hacia abajo) (Molchanov, Gupta, Kim y Pulli, 2015)

Varios otros trabajos se han realizado para el reconocimiento gestual en el control y seguridad de automóviles, generalmente utilizando únicamente cámara de color y cámara de profundidad. Ohn-Bar y Manubhai de la IEEE realizaron una aplicación de reconocimiento gestual de la mano para interfaces de automóviles utilizando un Kinect. Este trabajo pretende utilizar los gestos reconocidos para control de sistemas dentro del vehículo utilizando la información de color y profundidad. (Ohn-Bar & Trivedi, 2014)

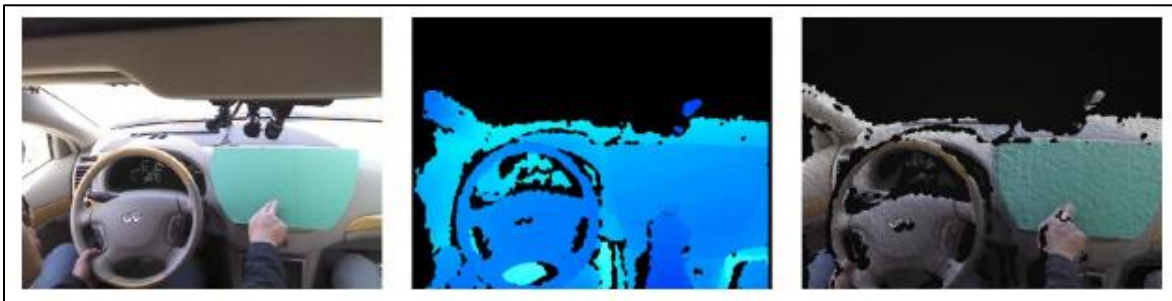


Figura 7: Cámara de color, cámara de profundidad y point cloud (Ohn-Barr y Manubhai, 2014)

En la Universidad de Virginia, en 2013, se desarrolló una aplicación de detección de gestos específicamente para detectar distracciones del conductor. Esta aplicación es capaz de detectar cuatro tipos de distracciones: alcanzar objetos con el vehículo en movimiento, distracciones relacionadas a la higiene personal (acomodarse repetidas veces el cabello), distracción por un objeto externo al vehículo, y como objetivo secundario, hablar por celular.

Para su detección se utilizan algoritmos que calculan distancias relativas entre las juntas del cuerpo y algoritmos para detectar la inclinación de la cabeza. Se determina que el conductor está distraído si se mantiene alguna de estas conductas por al menos dos segundos. Para alertar al usuario, se produce una señal auditiva que incrementa su frecuencia con el tiempo. (Gallahan, et al., 2013)

Por último, un trabajo más reciente de Solomon y Wang, explora la detección del comportamiento de un conductor utilizando un Kinect. Este sistema es capaz de detectar la fatiga y distracción del conductor, así como el uso del celular como propósito secundario. Utilizando la distancia relativa entre las articulaciones y los ángulos de la cabeza, se determina si el conductor se está quedando dormido o está viendo hacia una dirección distinta a la que debería. Los ángulos de la articulación de la cabeza determinan la dirección de la mirada del conductor y se activa una alarma si se mantiene una dirección no adecuada por un tiempo significativo. El sistema también es capaz de detectar si el conductor bosteza, la frecuencia con la que lo hace, y la cantidad total de bostezos. Esto permite determinar el nivel de fatiga. La posición relativa entre la mano y la cabeza determina si el conductor está sosteniendo un objeto cerca de su oreja, presumiblemente un celular. El sistema dibuja un esqueleto simplificado en donde se puede apreciar la posición de las uniones del cuerpo y la distancia relativa entre ellas. Se utiliza un temporizador para activar la alarma, con el fin de evitar reconocimientos erróneos y detecciones falsas. (Solomon & Wang, 2015) Estos últimos dos trabajos, son los que más se aproximan al proyecto que se quiere desarrollar.

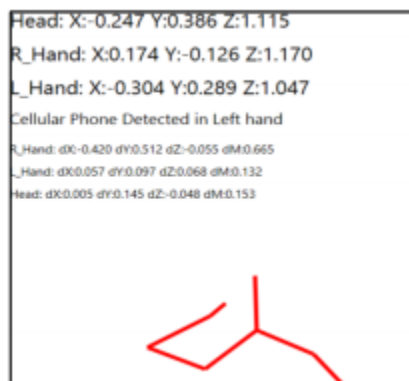


Figura 8: Demostración del esqueleto dibujado al sostener un celular cerca de la oreja (Solomon y Wang, 2015)

2.3. Marco Teórico

Antes de poder implementar la aplicación que se desarrolló, se debió hacer una simulación que utilice los sensores necesarios y que demuestre el correcto funcionamiento del sistema. Para ello, se utilizó el programa V-REP, el cual cuenta con modelos predeterminados de sensores, actuadores, robots y demás elementos utilizables en una simulación en el área de robótica. Además, cuenta con la posibilidad de agregar nuevos elementos, importar modelos desarrollados en un programa CAD, y modificar los parámetros de los componentes utilizados. Esto permite que los resultados obtenidos en la simulación se aproximen bastante a los resultados que se van a obtener en la implementación real.

Debido a que se debe simular el movimiento del cuerpo humano (movimiento de brazo), se trabajó con cinemática inversa para poder obtener el movimiento apropiado de las articulaciones del brazo que permitan llegar a la posición final deseada y regresar a la posición de reposo. Esto permite imitar el movimiento real de un brazo humano a la hora de hablar por celular, para que el sistema pueda detectarlo de manera correcta.

2.3.1. Cinemática Inversa

En la actualidad se está utilizando robots industriales que imitan el movimiento del brazo humano para realizar gran parte de las tareas industriales de manera más eficiente, rápida y con pocos errores. Para ello, se debió estudiar la naturaleza de la capacidad de movimiento de un brazo, imitando las diferentes partes para obtener un sistema altamente funcional. Un sistema como estos, al que se le conoce como “sistema multi-cuerpo”, cuenta con partes rígidas unidas por articulaciones o juntas que permiten el movimiento relativo entre las partes. Un sistema multi-cuerpo puede contar con diferentes tipos de articulaciones, incluyendo principalmente rotacionales y prismáticas (traslacionales). También se utilizan juntas esféricas, principalmente para simulación del esqueleto humano en animación. (Buss, 2004)

Para poder controlar el movimiento de las articulaciones, y consecuentemente el desplazamiento de los enlaces rígidos, se utiliza la cinemática inversa, en donde se supone que

para cada articulación se tiene una posición definida por ángulos y distancias, los cuales dependen de la posición y orientación de un punto definido del sistema llamado “efector final”. Al tener un sistema de coordenadas global y predefinido, se puede llevar al efector final a un punto determinado y con una orientación determinada. El movimiento y rotación de las articulaciones que se debe realizar para alcanzar esa posición y orientación meta, se obtiene mediante la cinemática inversa. El problema que se presenta, es que para una determinada posición del efector final, se puede obtener una cantidad finita o infinita de soluciones, o puede no haber solución alguna. Esto quiere decir, que los ángulos a los que se debe mover cada articulación varían mucho y no siempre se puede obtener el resultado que se desea. Por ello, se utilizan métodos computacionales para poder llegar a una solución (si existiera), y evaluar cuál de los resultados obtenidos es el más adecuado. Además, se pueden establecer los límites de las articulaciones para reducir la cantidad de soluciones posibles.

Algunos métodos utilizados para resolver problemas de cinemática inversa incluyen: métodos de coordenadas cíclicas descendientes, métodos pseudo-inversos, métodos del Jacobiano transpuesto, método de mínimos cuadrados amortiguados de Lavenberg-Marquardt, métodos de gradiente conjugado y quasi-Newtonianos, y métodos de inteligencia artificial y redes neuronales. (Buss, 2004)

Cada uno de los métodos produce una cantidad distinta de resultados posibles, por lo que se debe elegir entre ellos para encontrar la solución que sea más óptima. En el caso de la simulación de movimientos humanos, se debe poder encontrar la combinación de ángulos que produzca un movimiento real y alcanzable del cuerpo. Al tener un método adecuado, con solo especificar la posición y orientación de las manos, pies y cabeza, se puede recrear el movimiento de todo el cuerpo de una manera realista. El método debe ser lo suficientemente robusto para poder encontrar una solución a casi cualquier posición meta y retornar una aproximación bastante cercana cuando se tenga un punto que es inalcanzable con cualquier combinación de ángulos. Algunos métodos, como el método de Jacobiano transpuesto y el método pseudo-inverso, pueden oscilar significativamente ante posiciones meta que son inalcanzables, mientras que el método de los mínimos cuadrados presenta un comportamiento aceptable ante estas situaciones. (Kulpa & Multon, 2005)

Se tiene inicialmente una serie de uniones descrita por ángulos (en el caso de juntas rotacionales) representados por el símbolo θ , los cuales definen completamente el sistema multi-cuerpo. Se puede definir puntos específicos pertenecientes al sistema, los cuales serán los puntos de interés o “efectores finales” para los cuales se requiere conocer su posición y orientación. Su posición actual se puede definir con la variable \mathbf{S} y es una función de la configuración de ángulos del sistema. Asimismo, se puede definir la posición deseada o “posición meta” con la variable \mathbf{t} y la diferencia entre la posición meta y la posición actual se puede definir con la variable \mathbf{e} y representa el “error” o la distancia que se requiere recorrer para pasar de la posición que se tiene a la que se quiere llegar. La variable θ puede escribirse como un vector en donde cada entrada representa el valor de ángulo correspondiente a cada junta y las variables \mathbf{S} , \mathbf{t} , \mathbf{e} pueden escribirse como matrices en donde cada fila representa uno de los efectores finales o posiciones metas y las columnas son los valores correspondientes de acuerdo al sistema de coordenadas (generalmente X, Y y Z). Cuando no se puede conseguir una solución por medio de un método cerrado de solución, se utiliza la iteración para poder aproximarse lo mejor posible a una solución aceptable. Para esto se utiliza la matriz del Jacobiano, la cual es función de los ángulos de las juntas:

$$J(\theta) = \left(\frac{\partial \mathbf{s}_i}{\partial \theta_j} \right)_{i,j}. \quad (2.1)$$

Esto con \mathbf{i} siendo los efectores finales del sistema, y \mathbf{j} siendo las juntas que perteneces al sistema multi-cuerpo. Luego, la velocidad de las juntas puede escribirse utilizando la primera derivada con la ecuación:

$$\dot{\mathbf{s}} = J(\theta)\dot{\theta}. \quad (2.2)$$

Eligiendo una modificación en el ángulo de la junta ($\Delta\theta$) tal que el cambio en velocidad de la junta ($\Delta\mathbf{S}$) se aproxime al error (\mathbf{e}), se puede resolver de manera iterativa la ecuación para obtener la combinación de ángulos necesarios para llegar a la posición meta. (Buss, 2004) Esto generalmente se realiza por medio de herramientas computacionales ya que el número de iteraciones y la complejidad puede llegar a ser muy elevado. El simulador utilizado (V-REP) cuenta con herramienta de cálculo de cinemática inversa, a la cual se le deben modificar los parámetros para que se ajuste al resultado que se quiere obtener.

2.3.2. Substracción de fondo

La substracción de fondo es una técnica bastante utilizada en el procesamiento de imágenes para identificar elementos de interés, especialmente cuando se quiere reconocer un objeto en movimiento utilizando imágenes estáticas. Generalmente, el resultado que se obtiene es una máscara binaria que representa los píxeles pertenecientes al objeto de interés.

Como su nombre lo indica, la substracción de fondo resta una imagen pregrabada del fondo, de la imagen actual con la que se está trabajando, para poder obtener los elementos de ella que no pertenecen al fondo. Para hacer la tarea más sencilla, y para eliminar el ruido que se pueda generar por elementos no deseados en la imagen, se puede realizar inicialmente una operación de suavizado. Esto se conoce como pre procesamiento de imágenes. La substracción de fondo es una tarea muy común en visión computacional, ya que reduce la complejidad de reconocimiento de objetos al eliminar todo el ruido que se pueda encontrar en el fondo. Además, permite segmentar de manera más sencilla los píxeles que se van a utilizar para el análisis de la imagen. (Cheung & Kamath, 2007)

Para realizar la substracción de fondo, se debe tomar una imagen inicial que represente el “fondo”. Esta imagen puede contener cualquier cantidad de elementos diferentes, los cuales no son de interés para el análisis de imágenes y serán descartados a la hora de hacer una segmentación o de identificar los objetos. Generalmente, cada cierto tiempo se debe actualizar la imagen de fondo, ya que pueden aparecer nuevos objetos que no son relevantes para el análisis y que no serán reconocidos como “fondo” si no se realiza la actualización. (Vacavant, Chateau, Wilhelm, & Lequière, 2013)

La substracción de fondo puede realizarse sobre imágenes de color o en escala de grises, siendo la primera más compleja que la segunda. Para ello se utiliza el valor de intensidad del pixel y se compara con el pixel correspondiente en la imagen de fondo. Utilizando un valor umbral, se segmenta si éstos son muy diferentes. Esto puede generar un problema si el valor de intensidad del objeto en la imagen es muy similar al valor de intensidad del elemento de fondo. Si esto sucede, la aplicación no es capaz de diferenciar entre ellos y no marca el objeto como tal. Varios otros problemas se pueden presentar a la hora de aplicar una substracción de fondo.

Primero, el sistema se vuelve muy sensible ante movimientos de la cámara, ya que al tener la imagen de fondo pre-grabada, un movimiento causaría que se detecten objetos erróneos que realmente siguen perteneciendo al fondo. Otro problema que puede surgir es cuando hay un objeto nuevo en la escena que se detiene. Este objeto sigue siendo detectado por el programa, por lo que un nuevo objeto que aparezca frente a él no podrá ser detectado de manera adecuada. Además, si algún objeto que se había reconocido como elemento de fondo se llega a mover, éste será reconocido dos veces como objeto de interés (una vez en la parte que está interfiriendo con el fondo, y otra vez como la silueta de donde se encontraba anteriormente). Por último, debido a que se trabaja con niveles de intensidad, un sistema como estos es altamente sensible a cambios en la iluminación y a condiciones climáticas como lluvias y vientos que interfieran con la cámara. Por ello, debe utilizarse en un ambiente controlado en donde no se vaya a dar este tipo de problemas. (Cheung & Kamath, 2007)

Una solución que se puede dar a este tipo de problemas cuando se está analizando objetos en movimiento, es actualizar la imagen de fondo con cada ciclo del programa. Al convertir la imagen actual en la nueva imagen de fondo, todo objeto que entre a la escena y permanezca en ella será considerado como parte del fondo, y solo se podrán detectar los nuevos objetos de interés que llegan a la escena. Además, esto permite que el sistema se adapte muy fácilmente a cambios en la iluminación y a movimientos de la cámara. Sin embargo, debido a que con cada ciclo se cambia la imagen de fondo, solo se podrán detectar los objetos que se encuentren en movimiento en ese momento, marcando los bordes de los mismos (en dirección del movimiento y el borde que deja atrás). Por lo tanto, puede presentarse un problema de detección si se tiene un objeto que se acerca o se aleja de la cámara.

Para evitar este problema, se puede modificar el tiempo que transcurre entre una medición y otra, de modo que se tenga mayor diferencia entre dos cuadros de la imagen. Esto provoca que se tengan más píxeles reconocibles del objeto, pero se tendrá además la figura que el objeto dejó atrás al moverse. Para resolverlo, se toman dos imágenes (una en donde se haya dado el movimiento y se tengan las dos figuras, y otra en un tiempo siguiente en donde se tenga la siguiente figura del movimiento) y se hace un AND entre ellas, obteniendo así únicamente la imagen del objeto en el momento que se movió. Para esto es importante elegir un buen tiempo

entre mediciones, el cual depende de la aplicación que se va a realizar y del tamaño y velocidad del objeto. (Tamersoy & Aggarwal, 2009)

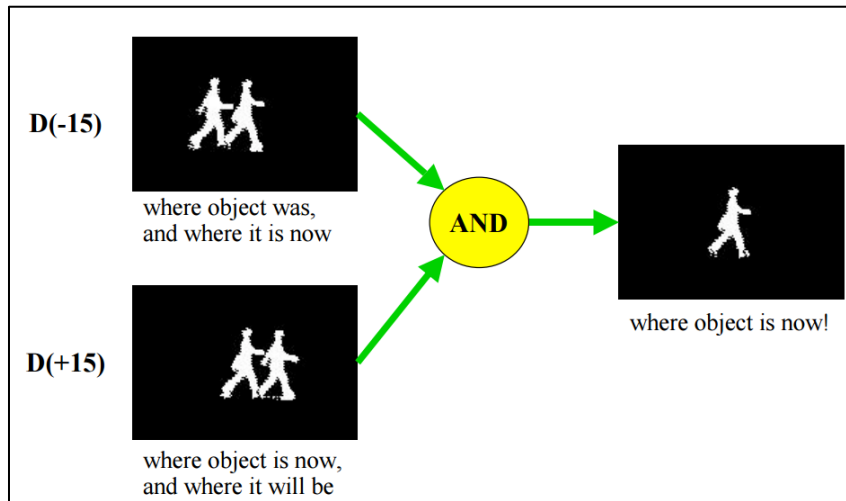


Figura 9: Obtención de una figura en movimiento utilizando substracción de fondo y comparación de imágenes en tiempos diferentes. (Tomado de http://www.ics.uci.edu/~dramanan/teaching/cs117_spring13/lec/bg.pdf)

Se pueden tener otros métodos para obtener la imagen de fondo, cuando no se puede tomar una imagen inicial o cuando se puede ver afectada con los cambios de iluminación o los nuevos objetos que llegan a la escena posteriormente. Algunos de ellos incluyen realizar el promedio de varias imágenes (más de 50) para obtener una imagen que represente satisfactoriamente el fondo. Otra opción es utilizar la mediana de las imágenes. Estas técnicas producen buenos resultados a la hora de hacer substracción de fondo pero requieren mucha memoria computacional, lo cual causa que se requiera un procesador muy robusto. Además, debe manejarse con cuidado dependiendo de la aplicación que se le va a dar, especialmente a la hora de definir el umbral con el que se va a segmentar los objetos, ya que si se tienen objetos pequeños, se cambian las condiciones de iluminación en el tiempo o se tienen objetos viajando a una velocidad muy baja o a una velocidad muy alta, puede que no se detecten si no se elige el valor umbral adecuado. (Tamersoy & Aggarwal, 2009)

Otras formas de identificar el fondo de la imagen se pueden obtener por métodos matemáticos. Utilizando una aproximación heurística en donde se modela cada pixel con una grupo de curvas Gaussianas adaptativas, se puede obtener la probabilidad de que dicho pixel

pertenezca al fondo. Los píxeles que no calzan en la caracterización del fondo, son identificados como elementos de interés, y se agrupan utilizando el principio de conectividad. Los elementos clasificados como fondo son aquellos que tengan alta repetitividad y baja varianza. La ventaja de esto es que no se tiene un único valor de umbral que pueda afectar los resultados obtenidos y puede variar en el tiempo sin afectar los la identificación de objetos. Algunas técnicas más complejas y avanzadas utilizan el factor de tiempo para la detección de fondo. (Stauffer & Grimson, 2007)

2.3.3. Sensor Kinect

El sensor Kinect fue inicialmente desarrollado por Microsoft para utilizarse en la industria de los juegos de video. Su propósito se basa en poder ofrecer al usuario una experiencia más natural al interactuar con la consola. El principio detrás de su funcionamiento se resume en poder entender el comportamiento del usuario (gestos, expresiones, acciones) antes de poder emitir una respuesta. Esto resultaba muy complicado con cámaras 2D sencillas por lo que se decidió agregar el elemento de profundidad para poder realizar tareas como: identificar cuando una persona está hablando, reconocer a la persona que está interactuando con el sensor, o hacer lectura de los gestos que realiza con las partes del cuerpo.

Hoy en día, el Kinect está siendo utilizado en áreas que trascienden la industria de los juegos de video, debido a su simplicidad de uso, su bajo costo, y su potencial en materia de interacción eficiente con el usuario, incluso en áreas como la robótica y la ingeniería. La variedad de aplicaciones que se le puede dar es muy amplia.

El sensor incorpora distintos tipos de hardware que le permiten realizar una variedad de tareas para distintas aplicaciones: sensor de profundidad, una cámara a color, y cuatro micrófonos. Esto permite la medición y detección del cuerpo completo en movimiento en tercera dimensión, así como reconocimiento facial y detección de gestos y de voz.

El sensor de profundidad que se utiliza en el Kinect se compone de un proyector infra-rojo y un sensor CMOS que capta la luz infra-roja. La luz estructurada del proyector se difracta para convertirse en una nube de puntos que impacta los objetos que se están midiendo. Al utilizar un patrón que se tiene en el proyector, se compara con la nube de puntos obtenida y se hace una triangulación para saber la profundidad de cada uno de esos puntos. Luego, se representa en 3D

como una imagen en escala de grises, en donde los valores menores de gris (más oscuros) representan los elementos que se encuentran más próximos a la cámara. Los elementos en negro (valor más bajo de gris) son aquellos que no retornan valor de profundidad, ya sea porque se encuentran muy lejos para ser detectados por el sensor, porque están muy cerca de la cámara (debido a la geometría se tiene una zona indetectable), o porque son superficies que no logran reflejar la luz de manera adecuada (por ser muy finas o por ser superficies especulares). (Zeng, 2012) Esta imagen en escala de grises se puede convertir posteriormente a una en pseudo-color para poderse visualizar más fácilmente.

La utilización de un sensor como este permite la elaboración de una amplia variedad de aplicaciones en distintas áreas, debido a que es un sensor muy completo e incluye una combinación de hardware ideal para distintos tipos de proyectos de visión computacional. Al combinar la información de la cámara de color con la información de profundidad, pueden obtenerse datos esenciales para proyectos como reconocimiento de gestos, identificación de personas, creación de entornos virtuales, aplicaciones médicas, entre otros. Es por ello que continúa siendo elegido como el sensor a utilizar en varias aplicaciones de alta complejidad tecnológica.

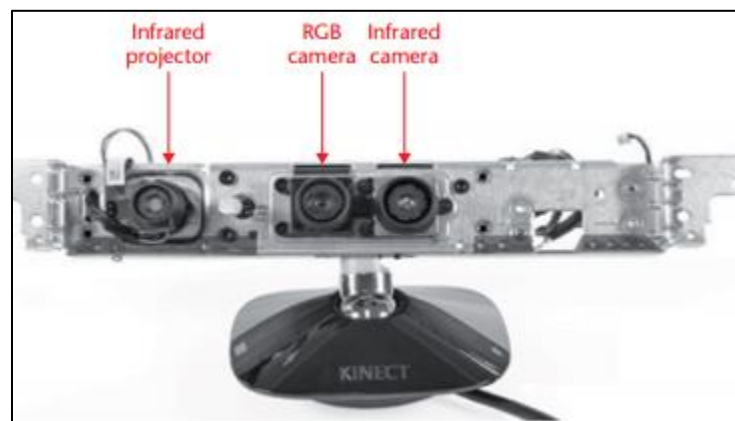


Figura 10: Componentes internos del sensor Kinect para Windows (Zeng, 2012)

Capítulo 3: Diseño preliminar

La simulación y la aplicación que se deben realizar tienen que cumplir ciertos requerimientos para poder alcanzar los objetivos del proyecto y para poder obtener los resultados que se esperan de ellas. Utilizando esta información, junto con la investigación previa que se realizó, se puede determinar los requerimientos y las especificaciones del proyecto.

3.1. Requerimientos:

- 1) El entorno simulado debe semejarse lo mejor posible al ambiente real en el que se va a utilizar la aplicación.
- 2) El programa debe ser capaz de sustraer los elementos que componen el fondo
- 3) La aplicación debe poder identificar los píxeles que pertenecen al cuerpo humano
- 4) El sistema debe poder interpretar la información de profundidad obtenida con un sensor 3D.
- 5) La aplicación debe poder reconocer la zona de interés que se quiere analizar.
- 6) El programa debe poder identificar el gesto de hablar por celular.
- 7) El sistema debe poder activar una “alarma” cuando se identifique que el conductor está hablando por celular.
- 8) El sensor debe contar con un soporte adecuado que permita sostenerlo de manera firme dentro de un vehículo.
- 9) El sistema debe contar con un campo de visión que permita observar las partes necesarias del cuerpo.
- 10) El sistema debe ser automático.
- 11) El sistema debe poder simular el movimiento del brazo humano para imitar la acción de hablar por celular.

A partir de los requerimientos se puede iniciar el diseño del proyecto, tomando en cuenta cada parte. Para ello se desarrolló la siguiente tabla morfológica que explora las diferentes posibilidades para la solución del problema.

3.2. Propuestas de solución

Tabla 2: Tabla morfológica para diseño de la solución (Creada por el autor)

Requerimiento				
1) Simulación de ambiente real	Figura humana en posición sentada	Sillas que simulen los asientos del carro	Posición relativa del sensor similar a donde se pondría en el automóvil	
2) Substracción de fondo	Resta directa de pixeles (comparación de 2 imágenes)	Igualar la imagen de fondo a la imagen actual para cada ciclo	Utilización de valor umbral	Utilizar promedio para determinar la imagen de fondo
3) Identificación de figura humana	Máscara con pixeles pertenecientes a "foreground" (valores 0 o 1)	Utilizar los valores de profundidad más bajos (elementos más cercanos a la escena)	Utilizar valores de proporción relativa	
4) Interpretación de información 3D	Datos obtenidos del Kinect (valores de 0 a 1)	Datos de profundidad por luz estructurada (tiempo de vuelo)	Datos de profundidad por sonares	
5) Reconocimiento de zona de interés (cabeza)	Identificación de articulaciones (hombros, codos, muñecas,	Información de profundidad para determinar donde inicia el cuello y donde	Histogramas de cantidad de pixeles por fila para encontrar el cuello	Información de color para identificar cambio de cabeza a cuerpo

	cabeza) y del esqueleto	termina la cabeza		
6) Reconocimiento del gesto de hablar por celular	Cercanía entre articulación de la muñeca y articulación de la cabeza	Análisis de zonas laterales a la cabeza	Medición de histograma en la zona de interés	Determinar si la mano se encuentra cerrada
7) Sistema de alarma	Alarma sonora	Alarma luminosa	Alerta escrita	
8) Soporte para el sensor	Estructura de aluminio	Estructura de madera	Estructura de plástico	Estructura de fibra de carbono

De la tabla morfológica se puede elegir las mejores opciones para realizar un diseño preliminar del sistema que se quiere desarrollar. Utilizando una combinación de las propuestas se pretende obtener la opción más viable para solucionar el problema.

3.3. Discusión de las soluciones

Para poder determinar cuál de las opciones es la mejor para la aplicación que se quiere desarrollar, se realizó un análisis de cada parte para finalmente poder obtener un diseño preliminar de lo que se va a construir. Para esto se elige la mejor opción de cada una de las partes, tomando en consideración la integración de las funciones y las consideraciones de diseño, manufactura y prototipado.

Para simular un ambiente real se debe considerar todos los componentes que están involucrados en la situación real que se pueda dar. En este caso, el sistema se va a posicionar en la parte interior de un carro. Por lo tanto, lo que va a ver el sensor será los asientos del carro y una persona en posición sentada. Puede haber otros objetos dentro del carro que van a ser considerados como elementos de fondo si están presentes en el momento en que se obtiene la imagen de fondo, antes de que la persona ingrese al vehículo. Debido a que el programa no cuenta con un asiento de automóvil, se puede simular utilizando sillas regulares, y una persona sentada en ellas. Un aspecto que se debe considerar es que la iluminación dentro de un vehículo es muy variada y pueden entrar rayos directos del sol, por lo que el sensor es susceptible a errores que se dan con la luz solar o los cambios de iluminación. Por esta razón, las pruebas de funcionamiento se realizarán en un ambiente de iluminación controlada, y para que funcione en

una situación real se deberán realizar trabajos posteriores. Finalmente, es importante posicionar el sensor en un lugar que asemeje la posición que tendrá dentro del automóvil, sin bloquear la visión del conductor ni obstruir las actividades regulares de transporte vehicular. Para ello, deberá posicionarse con una distancia e inclinación adecuada para simular estas circunstancias.

Para realizar la substracción de fondo existen varias alternativas que se pueden utilizar. Para obtener la imagen de fondo con la cual se va a hacer la comparación, se podría utilizar un promedio de las imágenes si se trata de un análisis de objetos en movimiento. Sin embargo, se estará trabajando con un único objeto principal que realiza movimientos leves, lo cual causa que el promedio de las imágenes no genere el fondo, sino una persona en su posición sentada más frecuente. Tampoco es viable asignar la imagen actual a la imagen de fondo, ya que la substracción generaría un contorno muy leve solo cuando se tienen movimientos importantes. La opción más adecuada es realizar una resta directa de píxeles, utilizando un valor umbral para evitar problemas generados por los cambios leves de iluminación. Para ello, se debe poder tomar una imagen de la escena antes de que la persona entre en ella.

Para discernir la figura de interés (la persona) de los elementos de fondo, la mejor opción es crear una máscara que identifique los píxeles que pertenecen al cuerpo humano. Para ello se utiliza la substracción de fondo discutida anteriormente y se comparan las dos imágenes para decidir si se asigna un valor de 0 o de 1 a la máscara. La opción de utilizar los valores más bajos de profundidad (puntos más cercanos) puede ser utilizada si se conoce la distancia a la que se posiciona el sensor con respecto a los objetos de fondo y se trabaja con un valor umbral que determina si el objeto detectado es una persona. Sin embargo, puede haber otros objetos pertenecientes al fondo que estén más cercanos al sensor (como por ejemplo la manivela), por ello se descarta esta opción. Finalmente, utilizar los valores promedios de proporción relativa produciría resultados indeseables, debido a que la variedad de tipos de cuerpo es infinita y la complejidad en la programación sería muy elevada. Por lo tanto, se decide trabajar con una máscara.

Existen varios sensores que se podrían utilizar para el desarrollo de esta aplicación que permiten obtener información de profundidad. Un sensor de profundidad con luz estructurada (laser) o sonar proporciona la información necesaria pero generalmente tienen un precio muy elevado. El sensor Kinect cuenta con datos obtenidos de una nube de puntos de luz infrarroja que

proporciona información muy precisa de la profundidad. Además, provee datos con información en RGB que puede utilizarse como complemento de la información de profundidad y se encuentra ampliamente en el mercado, a un bajo precio, y con software “*open source*”. Estas características le dan una ventaja sobre otros tipos de sensores, por lo que representa la mejor opción para el desarrollo del proyecto.

Para poder reconocer gestos, se debe poder identificar la zona de interés sobre la cual se va a establecer el enfoque de análisis. En este caso, la zona de interés representa los alrededores de la cabeza, debido a que un acercamiento de un objeto por un tiempo prolongado se podría interpretar como el gesto de hablar por celular. Para poder identificar esta zona de interés, lo mejor es poder identificar el extremo superior de la cabeza y la parte donde inicia el cuello. Para realizar esto existen varias opciones. Esta es una de las partes más importantes de la etapa de simulación por lo que se van a explorar varias de las opciones para poder utilizar la que provee mejores resultados. Una opción que se puede utilizar es la de identificar las articulaciones del cuerpo para determinar la posición de la mano. Para ello se debe poder distinguir la cabeza, los hombros, los codos y las manos. Esta opción es muy utilizada en aplicaciones similares ya que resulta sencillo después poder calcular la distancia absoluta entre la cabeza y la mano y establecer un umbral que determine si la cercanía indica que se está dando el gesto. Además, el Kinect cuenta con varios algoritmos de acceso libre que permiten realizar identificación del esqueleto. Sin embargo, esta aproximación es más orientada a aplicaciones de experiencia de juego y no es completamente confiable, ya que muchas veces falla en el reconocimiento del esqueleto. Asimismo, no se cuenta con esta funcionalidad en el programa de simulación, por lo que tendría que realizarse toda una aplicación para reconocimiento del esqueleto que tomaría más tiempo y puede no ser tan efectiva. Otra alternativa para el reconocimiento del cuello y la cabeza es utilizar la información de color que provee el Kinect. Para ello se deberá reconocer cuando exista un cambio importante de color. Sin embargo, puede haber muchos casos de error ya que en la cara se tiene variedad de colores (vello facial, accesorios, ojos, etc) que causarían una detección errónea. Además, el área del cabello es muy variable entre personas, por lo que sería difícil reconocer el valor de color de la cara. Otro problema que podría surgir es si la persona está utilizando una camisa de color similar a la piel, o con valores RGB muy cercanos.

Debido a estas complicaciones, se eligió trabajar con las últimas dos opciones, las cuales incluyen utilizar la información de profundidad para detectar el inicio del cuello y utilizar histogramas de píxeles para cada fila. Para utilizar la información de profundidad, primero se debe encontrar el punto más alto (que corresponde al extremo superior de la cabeza) y utilizarlo como punto de referencia. Se sabe que la profundidad a lo largo de la cara no va a cambiar mucho, pero se dará un cambio drástico cuando se llega al cuello. Esta es una opción viable si se cuenta con una resolución adecuada en el sensor, ya que el cambio en la profundidad no será notable si se cuenta con un sensor de baja resolución. El sensor utilizado en el entorno de simulación tiene una resolución relativamente baja. No obstante, el sensor que se tiene en físico tiene una resolución mucho mayor, por lo que puede ser utilizado sin mayor problema. La última opción que se exploró es la de utilizar histogramas de píxeles. Para ello, se contabiliza la cantidad de píxeles que pertenecen a la figura del cuerpo (utilizando la máscara) para cada fila horizontal. Se sabe que la parte del torso y los brazos tendrá un ancho casi uniforme hasta llegar a los hombros. Una vez que se llega al cuello, la cantidad de píxeles es mucho menor. Este cambio drástico indica que se tiene el inicio del cuello. Esta es una opción muy sencilla de implementar y muy eficiente, pero se deben tener ciertas restricciones; como por ejemplo utilizar el cabello recogido si se tiene largo, ya que esto podría afectar la medición del ancho del cuello. Además, no todos los tipos de cuerpo son iguales, por lo que deberán realizarse pruebas con una variedad de personas para verificar el funcionamiento adecuado.

Utilizando estas opciones, existen una variedad de posibilidades para hacer la detección del gesto de hablar por celular. Debido a que se descartó la identificación del esqueleto, también se descarta la opción de medir la cercanía entre la articulación de la cabeza y la articulación de la mano. Para la medición por diferencias de profundidades, al detectar el punto máximo y el mínimo de la cabeza (extremo superior y cuello) se pueden analizar las zonas laterales de la cabeza para detectar si se encuentra un objeto cercano, indicando que se acercó la mano a la oreja para hablar por celular. Para la medición por histograma, se puede guardar los valores de cada fila y compararlos con la imagen actual, de modo que si se da un cambio importante en la cantidad de píxeles pertenecientes al cuerpo en la zona de interés, significaría que hay algún objeto que se ha acercado, ya que el ancho de la cabeza no aumenta de manera importante por sí sola. Otra opción que se puede incluir para hacer la detección más precisa es determinar si la mano del usuario se encuentra cerrada, ya que la combinación de cercanía a la oreja y mano

cerrada indicaría el gesto de hablar por celular. El programa de control del sensor Kinect incluye esta funcionalidad que puede ser utilizada una vez que se realicen las pruebas con el sensor en físico. Sin embargo, la figura humana que se encuentra disponible en el programa de simulación no puede cerrar y abrir las manos, por lo que no se puede realizar una simulación de esta funcionalidad.

Para el sistema de alarma cuando se da la detección, la opción más viable es utilizar una alerta sonora, ya que es eficiente para llamar la atención del usuario sin distraerlo de la conducción, como lo haría una alerta escrita en la cual se debe desviar la mirada de la vía para poder leerla. Una señal luminosa también puede ser utilizada para alertar, pero no es tan eficiente como una alerta sonora.

Para la construcción de la estructura, se debe utilizar un material que sea resistente ante temperatura y rayos del sol, ya que se va a posicionar en el interior de un carro que se encuentra constantemente expuesto a estas condiciones. Además, debe ser un material barato y debe poderse mecanizar fácilmente para ser viable. Por lo tanto, la opción más adecuada sería el aluminio, ya que cuenta con estas propiedades y es altamente utilizado en componentes de automóviles.

3.4. Especificaciones

A partir de la definición de los requerimientos y la discusión de las posibles soluciones que se pueden implementar para cada parte, se puede realizar una lista de especificaciones para el proyecto. Para ello se realizaron algunas pruebas en el programa para determinar las características y los alcances de los componentes y así definir más claramente lo que se va a desarrollar.

- 1) El ambiente simulado debe incluir elementos que representen lo mejor posible los objetos reales que se van a dar en la escena, incluyendo sillas, una figura humana, y una inclinación y distancia del sensor que sea similar a la que se va a utilizar dentro de un vehículo.
- 2) El sistema realizará una substracción entre una imagen de fondo previamente capturada y la imagen actual detectada por el Kinect con un valor umbral determinado experimentalmente.

- 3) El programa creará una máscara que identifique los píxeles que pertenecen a la figura humana utilizando un valor de 0 para elementos de fondo y un valor de 1 para elementos de interés.
- 4) El sistema utilizará un sensor Microsoft Kinect para recolectar los datos de profundidad que serán utilizados para la aplicación.
- 5) El programa determinará de manera automática la zona de interés que se irá a analizar utilizando un histograma de píxeles y los valores de profundidad relativos para encontrar dicha zona.
- 6) El programa utilizará los puntos encontrados en el paso anterior para determinar si se encuentra algún objeto cercano a la cabeza y así poder detectar el gesto que se requiere reconocer.
- 7) El sistema activará una alarma sonora cuando se detecte el gesto de hablar por celular por más de 5 segundos consecutivos.
- 8) El dispositivo deberá contar con un soporte hecho de aluminio, el cual deberá tener unas dimensiones máximas de 8 cm de ancho y 10 cm de alto para poder sostener el sensor de manera eficiente y posicionarse de manera segura dentro del vehículo sin obstruir la visión del conductor.
- 9) El sensor estará posicionado a una distancia mínima de 30 cm de la persona para poder obtener un campo de visión adecuado para la lectura de datos.

Capítulo 4: Desarrollo de la simulación

Antes de poder implementar la aplicación que se va a desarrollar se debió realizar una simulación que permite verificar el funcionamiento y realizar pruebas utilizando elementos que se asemejen al entorno real con el que se va a trabajar. Los sensores que se van a utilizar se pueden simular utilizando el programa V-REP para aplicaciones de robótica, el cual permite visualizar el comportamiento y la recolección de información de varios sensores y actuadores reales. Para cada simulación realizada en el programa se utiliza una escena, que contiene todos los elementos involucrados en dicha simulación. Los elementos que forman parte de una escena se conocen como modelos, e incluyen elementos estáticos como mesas, sillas, floreros, ventanas, árboles, etc, y elementos dinámicos como actuadores y robots, tanto móviles como con base fija (brazos industriales). Algunos de los actuadores y robots que se encuentran disponibles en el mercado ya han sido modelados para ser utilizados en el programa, y su comportamiento es muy similar al que se tiene en una aplicación real. Además de estos componentes, también se pueden cargar modelos de algunos sensores como lo son un GPS, sensores de proximidad, sensores de luz, sensores laser, giroscopios, etc. Dentro de estos modelos preestablecidos se puede encontrar el sensor Kinect, que se utilizará para el proyecto.

En el entorno de simulación también se puede regular el espacio de trabajo de los elementos de la escena, así como algunos parámetros como la iluminación, la presencia de humo, y la posición de las “cámaras” o puntos de vista que se tendrá de la escena. Asimismo, se puede configurar para tener distintas vistas de la simulación, que permiten un mejor control de la escena. Para cada elemento que se adiciona a la escena, se le puede modificar el “script” o el código fuente que rige su comportamiento. De esta forma, se determina las acciones que se van a tener dentro de la simulación, dependiendo de las situaciones medibles.

La forma en la que se trabajó la simulación es por elementos. Los componentes principales son un sensor Kinect y una persona, para los cuales se tendrá un código que determine su movimiento o mediciones que realiza. En la escena se tendrá además, una mesa para sostener el Kinect en posición, al menos una silla para simular el asiento del vehículo, algunos elementos aleatorios en el fondo para probar que la substracción se dé de manera apropiada, un “dummy” en la mano de la persona para representar el celular, y un suelo reconfigurable que será la superficie sobre la cual se posicionan todos los elementos de la escena. Inicialmente, se trabajó con una escena como la que se muestra a continuación.

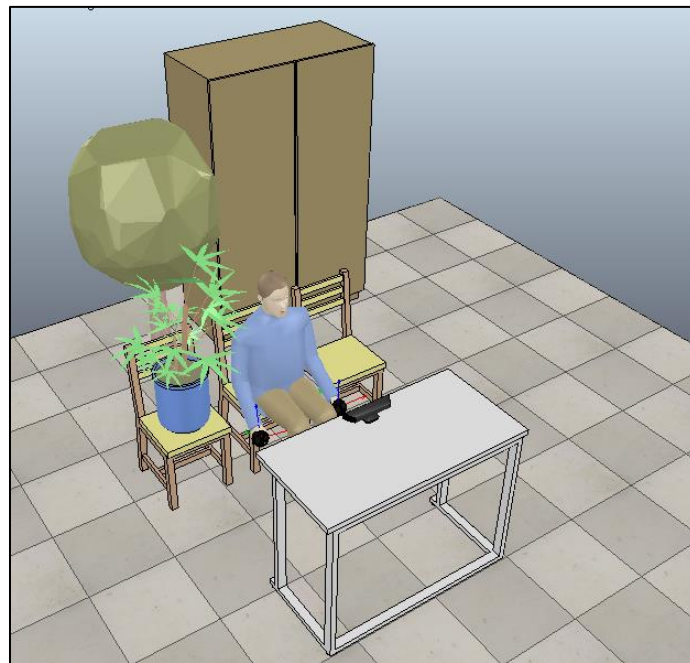


Figura 11: Escena de simulación inicial (Creado por el autor en programa V-REP)

En esta escena se puede ver que se utilizó una mesa alta con un sensor Kinect para hacer mediciones, tres sillas convencionales, una persona sentada sobre una de las sillas y elementos del fondo que incluyen una maceta con una planta, un armario y un árbol. Evidentemente varios de los elementos mostrados en la escena inicial no aparecerán en un ambiente real en donde se utilice la aplicación, pero funcionan como elementos para probar el funcionamiento de la substracción de fondo que se va a realizar. Después de crear la escena simulada, se debió proceder a realizar el código para cada componente que realizará alguna medición o actuación.

4.1. Sensor Kinect

El sensor Kinect tendrá un código propio que determina la información que está recibiendo, los parámetros de medición, y los cálculos que se realizan a partir de la información que se recibe. Además, a partir de los datos que se obtienen por medio del sensor, se pueden establecer variables que se interpretan por otros componentes de la escena y determinan acciones que se van a realizar.

Este sensor cuenta con un campo de visión que abarca 57° en el eje X y 43° en el eje Y (como se muestra en la Figura 12). Sin embargo, al ser un sensor para Windows y no para aplicaciones de experiencia de juego, su distancia mínima de medición es de 0.2 m y la máxima es de 3.3 metros.

En el ambiente de simulación se pueden alterar los parámetros del Kinect dependiendo de las necesidades que se tengan para la aplicación que se esté realizando. Inicialmente, se trabajará con una resolución de 64×48 píxeles, de modo que se reduzca el tiempo de procesamiento de la simulación. Una vez implementado, y utilizando un procesador más rápido, se puede volver a utilizar la resolución original de 640×480 simplemente volviendo a cambiar los parámetros que se modificaron en un principio.

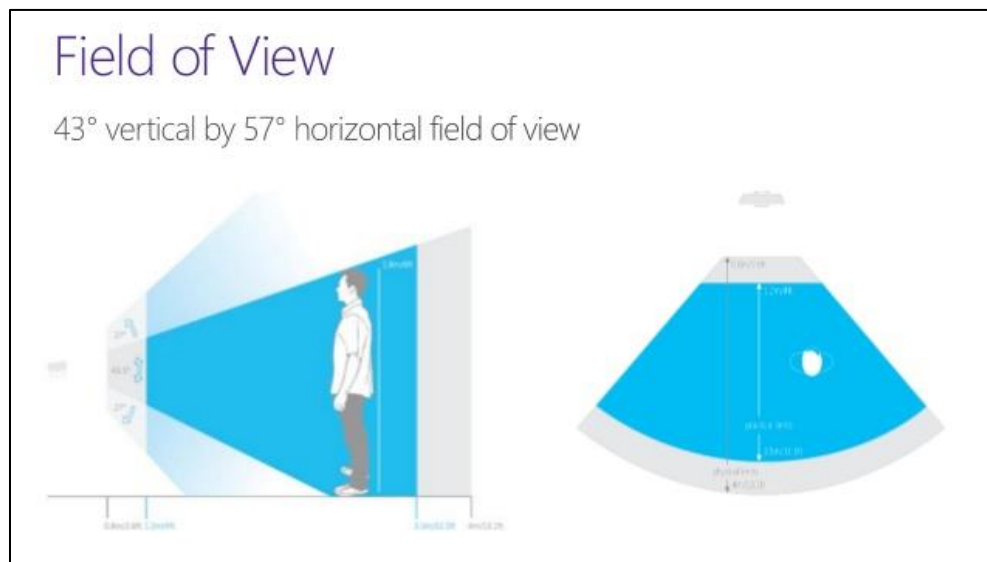


Figura 12: Campo de visión del Kinect (Tomado de <https://msdn.microsoft.com/en-us/library/hh973074.aspx>)

Existen varias funciones que se realizaron utilizando el sensor Kinect, incluyendo la toma de imagen de fondo, la creación de la máscara que identifica a la figura humana (substracción de

fondo), la identificación de la zona de interés (zona de la cabeza) y la identificación del gesto de hablar por celular. Para cada una de estas funciones se tiene un código que determina la forma en la que se va a realizar cada acción.

4.1.1. Toma de imagen de fondo

Como se argumentó anteriormente, la imagen de fondo se obtuvo por medio de una toma de imagen antes de que ingrese la persona al vehículo, de modo que todo lo que se encuentre dentro del automóvil en ese momento será considerado parte del fondo. En la simulación, se utilizó un botón para indicar el momento en el que se debe almacenar los datos de los píxeles que pertenecen al fondo. Para realizar el código para esta parte se utilizó el siguiente diagrama de flujo.

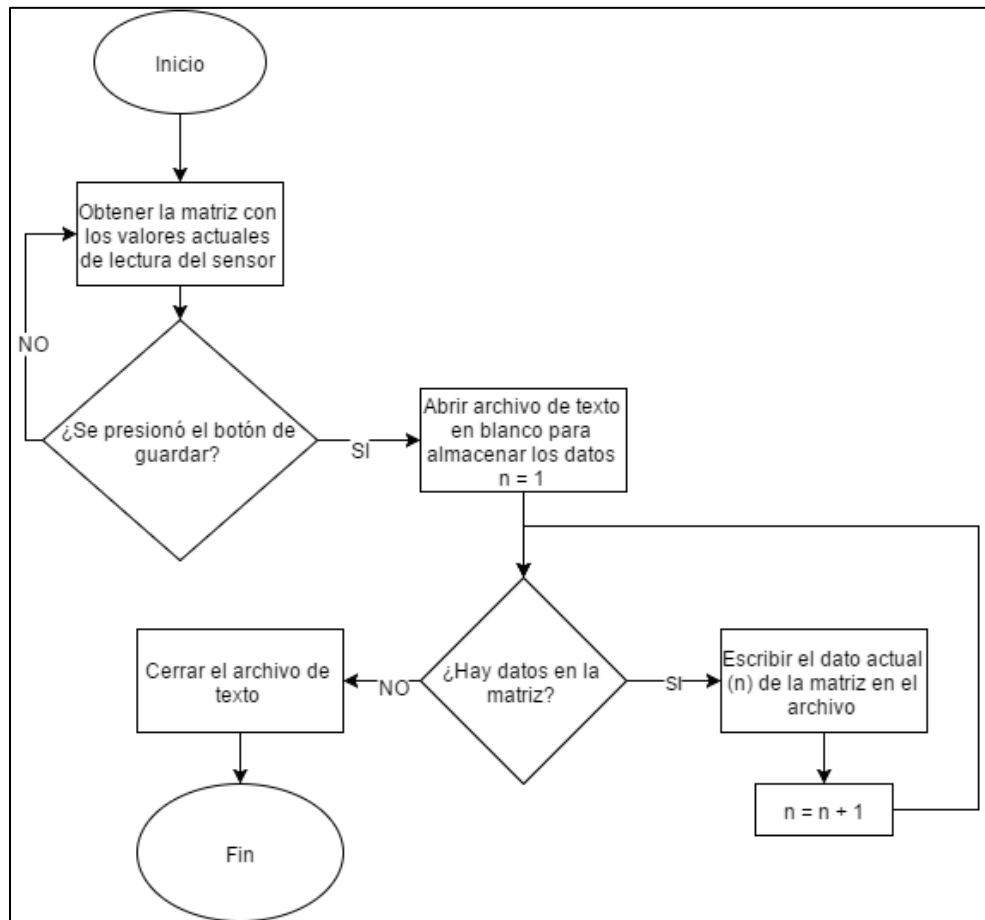


Figura 13: Diagrama de flujo para almacenar los datos del fondo (Creado por el autor en Visio)

Para obtener esta imagen en el programa V-REP, se toma la escena inicial y se elimina temporalmente a la figura humana, para poder obtener únicamente los datos del fondo. Para efectos de la simulación se posicionó un botón llamado *Fondo* que al presionarse abre un archivo de texto en blanco y almacena los valores actuales de profundidad que está recibiendo el Kinect. Este archivo es posteriormente cargado de nuevo para poderse comparar con la imagen actual. En una aplicación real se esperaría que este proceso se realice de manera automática, por lo que se almacenaría la imagen de fondo en el momento en que el usuario presiona el botón para abrir el carro. De esta forma se estaría asegurando que no hay ninguna persona detrás del volante, por lo que todo lo que se encuentra dentro del vehículo pertenece al fondo de la imagen.

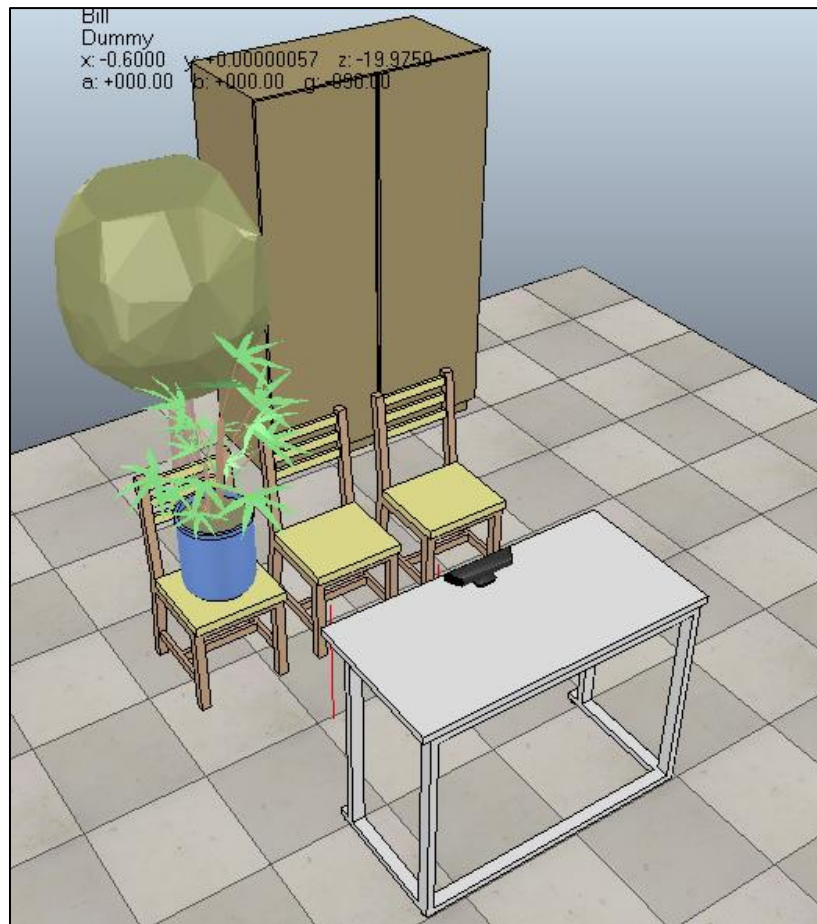


Figura 14: Imagen de fondo en la simulación (Creado por el autor en V-REP)

El archivo resultante que se crea al presionar el botón es un documento de texto con datos numéricos con 4 posiciones decimales que van del 0 al 1 (0 siendo el elemento más cercano a la

cámara y 1 el elemento más lejano). Este archivo tendrá 64 columnas y 48 filas, correspondiendo a la cantidad de píxeles que se tienen por la resolución que se le estableció al Kinect.

4.1.2. Substracción de fondo

Para realizar la substracción de fondo se debe primero obtener la imagen de fondo que se tomó en el paso anterior y compararla con la información actual que está recibiendo el sensor. Para ello se debe cargar el archivo que almacena la información de la imagen de fondo y comparar cada píxel utilizando un valor umbral. Este valor se utiliza para evitar errores importantes que se puedan dar por cambios leves en iluminación o vibraciones del sensor. Debido a que en la simulación no se dan variaciones de iluminación ni movimientos del Kinect, se puede utilizar cualquier valor umbral para esta parte. Sin embargo, al realizar las pruebas con el sensor en físico se deberá ajustar el valor umbral para producir los mejores resultados posibles.

La substracción de fondo se realiza de manera continua en un ciclo infinito mientras se esté corriendo el programa, con el fin de poder trabajar con la posición actual de la persona y poder hacer una detección apropiada. De esta forma, se actualiza con cada ciclo la máscara que se utiliza para identificar la figura humana. Esta máscara se almacena en forma de una matriz y puede ser utilizada después para hacer mediciones que determinen el reconocimiento del gesto que se quiere identificar. La substracción de fondo se realizó siguiendo el diagrama de flujo de la figura 15.

El resultado al aplicar este algoritmo es una matriz con valores de 0 o 1 que permite identificar la posición de los píxeles que pertenecen al objeto de interés (figura humana) y excluye aquellos que pertenecen al fondo.

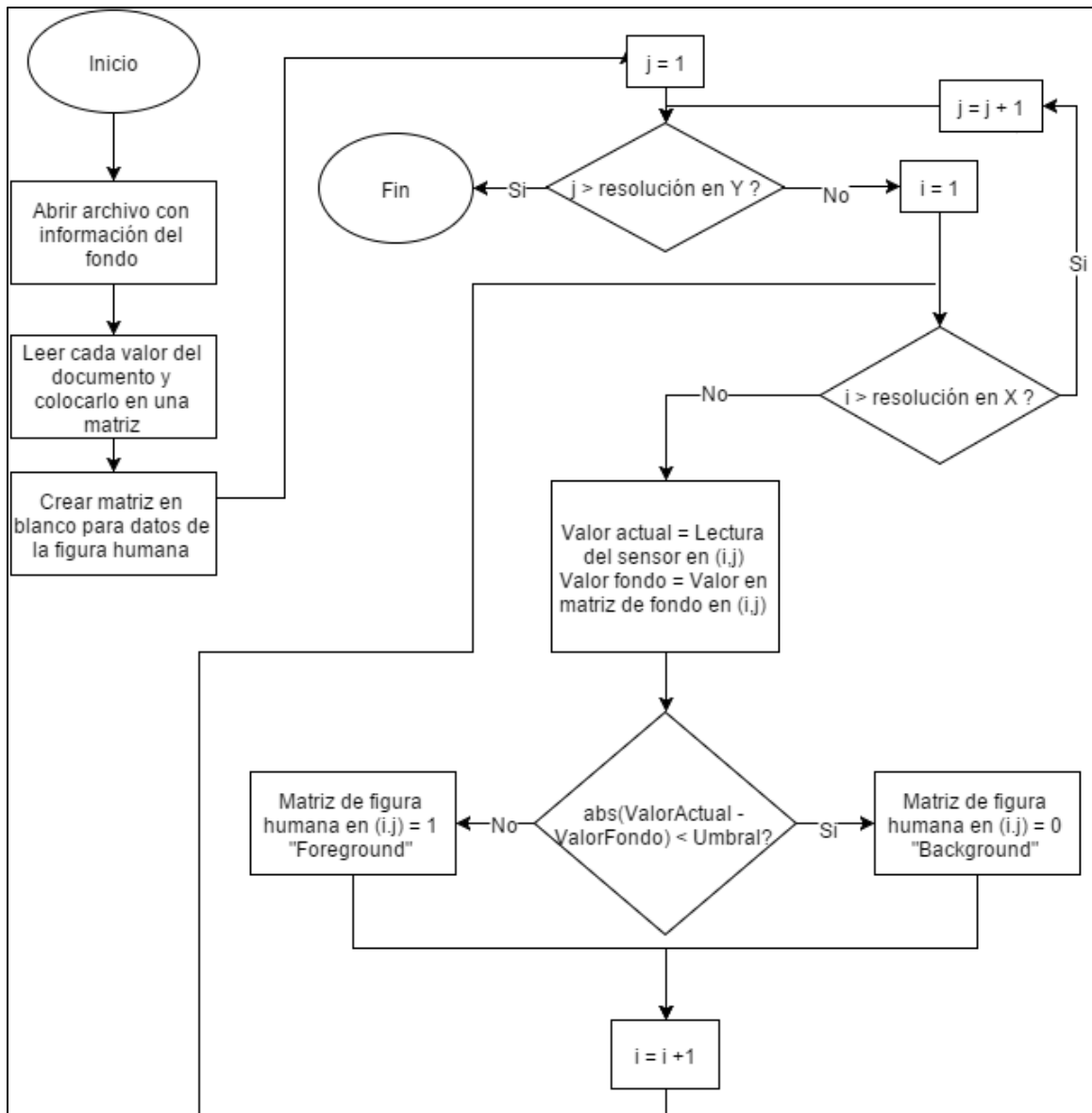


Figura 15: Diagrama de flujo para la substracción de fondo (Creado por el autor en Visio)

4.1.3. Identificación de la zona de interés

La zona de interés con la que se trabajó es el área que va desde el cuello de la persona hasta la punta de la cabeza, ya que al identificar un objeto que se acerca a esta zona durante el tiempo en que la persona está conduciendo, indicaría la presencia de un celular. Antes de poder identificar que existe un objeto en cercanía a la cabeza, se debe determinar el punto en donde

comienza y termina ésta, ya que varía con cada persona y con la posición del asiento que establezca la persona al conducir.

La identificación de la zona de interés deberá hacerse una única vez y asegurando que la persona se encuentre en posición neutral (con los brazos abajo) para evitar mediciones erróneas. Para tomar estos datos, se estableció de nuevo un botón que almacena la información actual del sensor y a partir de ella realiza las mediciones para encontrar los dos puntos de interés. En una aplicación real, dicho botón puede estar ligado al encendido del carro, ya que en este punto se asume que el conductor se encuentra en posición neutral.

Para determinar el punto máximo de la cabeza, se puede utilizar la información de profundidad del sensor Kinect y realizar cálculos de distancias en X, Y y Z, de modo que el punto que tenga coordenada Y máxima y además pertenezca a la máscara de la figura humana, será el punto superior de la cabeza. Para estas mediciones se utiliza los ángulos de medición del Kinect, los cuales son 57° en X y 43° en Y. Luego, de estos ángulos se obtiene el equivalente a la mitad del espacio de medición del sensor y se transforma a radianes.

$$\text{ÁnguloMedioX} = 57^\circ * 0.5 * \frac{\pi}{180} \quad (4.1)$$

$$\text{ÁnguloMedioY} = 43^\circ * 0.5 * \frac{\pi}{180} \quad (4.2)$$

Una vez que se tienen estos valores se recorre la matriz que contiene la información de profundidad para calcular el ángulo al que se encuentra cada punto. El ángulo medido será un porcentaje del ángulo medio, dependiendo de la posición (i,j) en la que se encuentra. Para cada ángulo se puede utilizar las siguientes fórmulas, en donde *ResX* y *ResY* representan la resolución en cada eje del Kinect (64 y 48 respectivamente para el caso del proyecto) y las variables *i* y *j* representan la posición actual en x, y de la matriz. Los factores de 0.5 se utilizan para posicionarse en el punto medio del pixel. El ángulo correspondiente a cada pixel de la matriz es una fracción del ángulo medio.

$$\text{Ángulo X} = \frac{\frac{ResX}{2} - i - 0.5}{\frac{ResX}{2}} * \text{ÁnguloMedioX} \quad (4.3)$$

$$\text{Ángulo } Y = \frac{j - \frac{ResY}{2} + 0.5}{\frac{ResY}{2}} * \text{ÁnguloMedio}Y \quad (4.4)$$

Después de tener el ángulo correspondiente para X y para Y, se obtiene la coordenada Z utilizando la información de profundidad. Con este valor se puede calcular la coordenada X y la coordenada Y medidas desde el sensor. De esta forma, para cada punto de la matriz, se tiene una posición (x,y,z) que se puede utilizar para calcular distancias. Las fórmulas para calcular cada coordenada son las siguientes.

$$\text{Coord}Z = \text{NearClip}P + \text{AmpProf} * \text{Valor}P \quad (4.5)$$

$$\text{Coord}X = \tan(\text{Ángulo}X) * \text{Coord}Z \quad (4.6)$$

$$\text{Coord}Y = \tan(\text{Ángulo}Y) * \text{Coord}Z \quad (4.7)$$

Tabla 3: Descripción de las variables para las ecuaciones 4.5 a 4.7

Nombre	Descripción
NearClipP	El “near clipping plane” representa la distancia mínima que puede ser medida por el sensor. En el caso del Kinect utilizado esta es de 0.2 m.
AmpProf	La amplitud de profundidad representa la distancia máxima que puede ser medida por el sensor. En el caso del Kinect utilizado es de 3.3 m, tomando el “near clipping plane” como el punto 0.
ValorP	El valor de profundidad es un valor de 0 a 1 que indica la profundidad del punto medido. Al ser multiplicado por la amplitud de profundidad se obtiene una distancia real a la que se encuentra el punto.

Con estas coordenadas calculadas, se puede encontrar el punto que tenga la coordenada Y máxima y que al mismo tiempo pertenezca a la máscara con los valores de figura humana. Este será el punto más alto de la cabeza. Para obtener el algoritmo que encuentre el punto máximo de la cabeza se utilizó el siguiente diagrama de flujo.

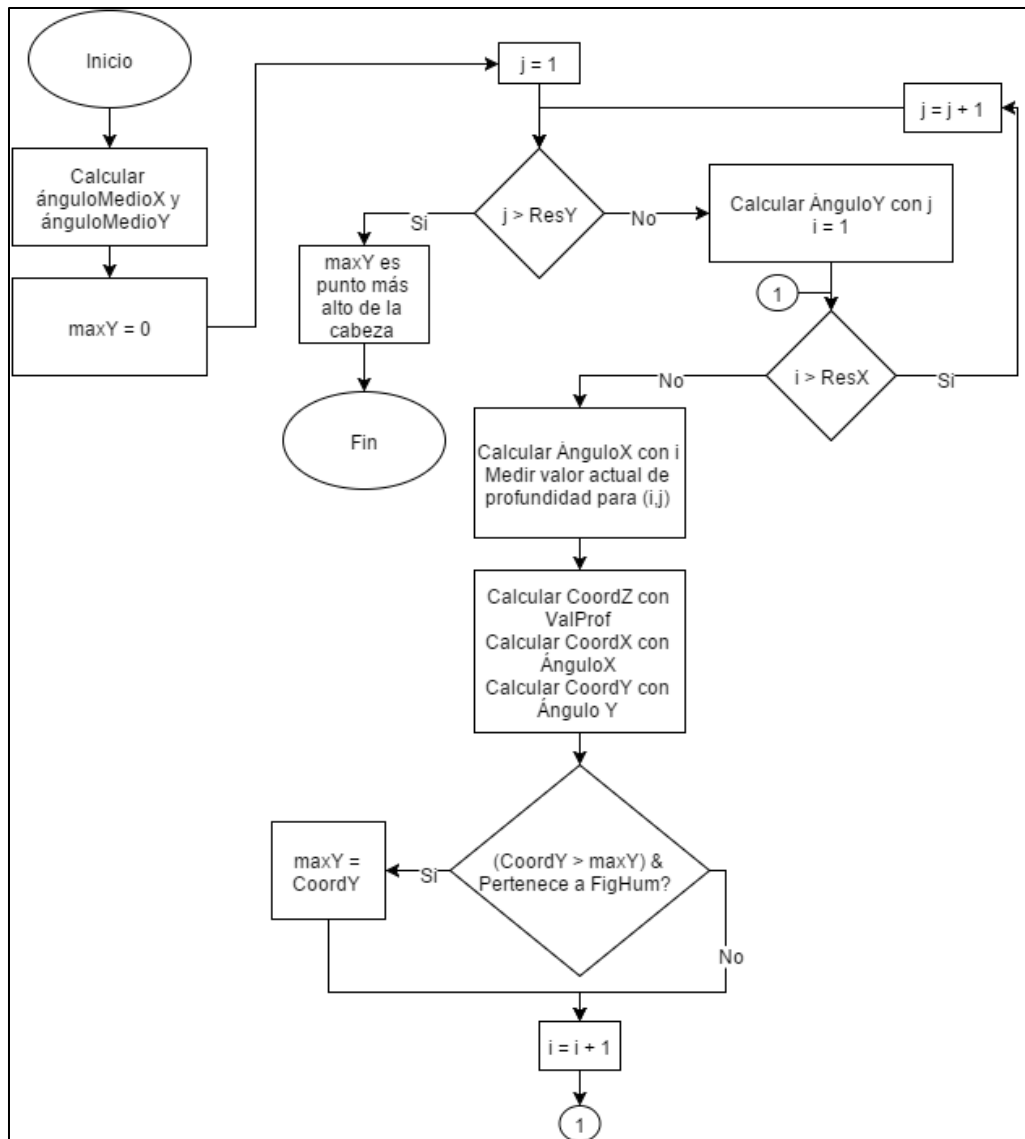


Figura 16: Diagrama de flujo para encontrar el punto máximo de la cabeza. (Creado por el autor en Visio)

El siguiente paso a realizar es encontrar el punto en donde inicia el cuello. Para ello hay dos métodos que se utilizaron. El primero de ellos consiste en realizar un histograma con la cantidad de pixeles que pertenecen a cada fila horizontal de la máscara. Se sabe que una persona en posición neutral tendrá un ancho casi uniforme a lo largo de su cuerpo hasta llegar al cuello, en donde se verá reducido significativamente el ancho medido (cantidad de pixeles). Para poder leer la cantidad de pixeles por fila, se recorre cada fila y se posiciona un contador que se incrementa cada vez que se reconoce un pixel que no pertenece al fondo. Al finalizar el recorrido

de cada fila, se guarda el valor del contador en un vector y se reinicia el contador. Al finalizar el recorrido de todas las filas, se tendrá un vector con todos los valores correspondientes al conteo de cada fila, por lo que se podrá hacer una comparación entre ellos.

Al tener el vector con los valores del contador correspondientes a cada fila, se procedió a determinar una proporción entre el ancho máximo del cuerpo y el ancho del cuello que fuera adecuada para poder identificarlo eficientemente. Utilizando histogramas para distintos casos de distancia e inclinación del sensor (Ver Apéndice 1) se determinó que la relación entre el máximo número de píxeles y el cuello será de al menos 5.1. Este valor se puede utilizar entonces como umbral para determinar el punto de inicio del cuello. Esta parte del algoritmo, sigue el siguiente diagrama de flujo.

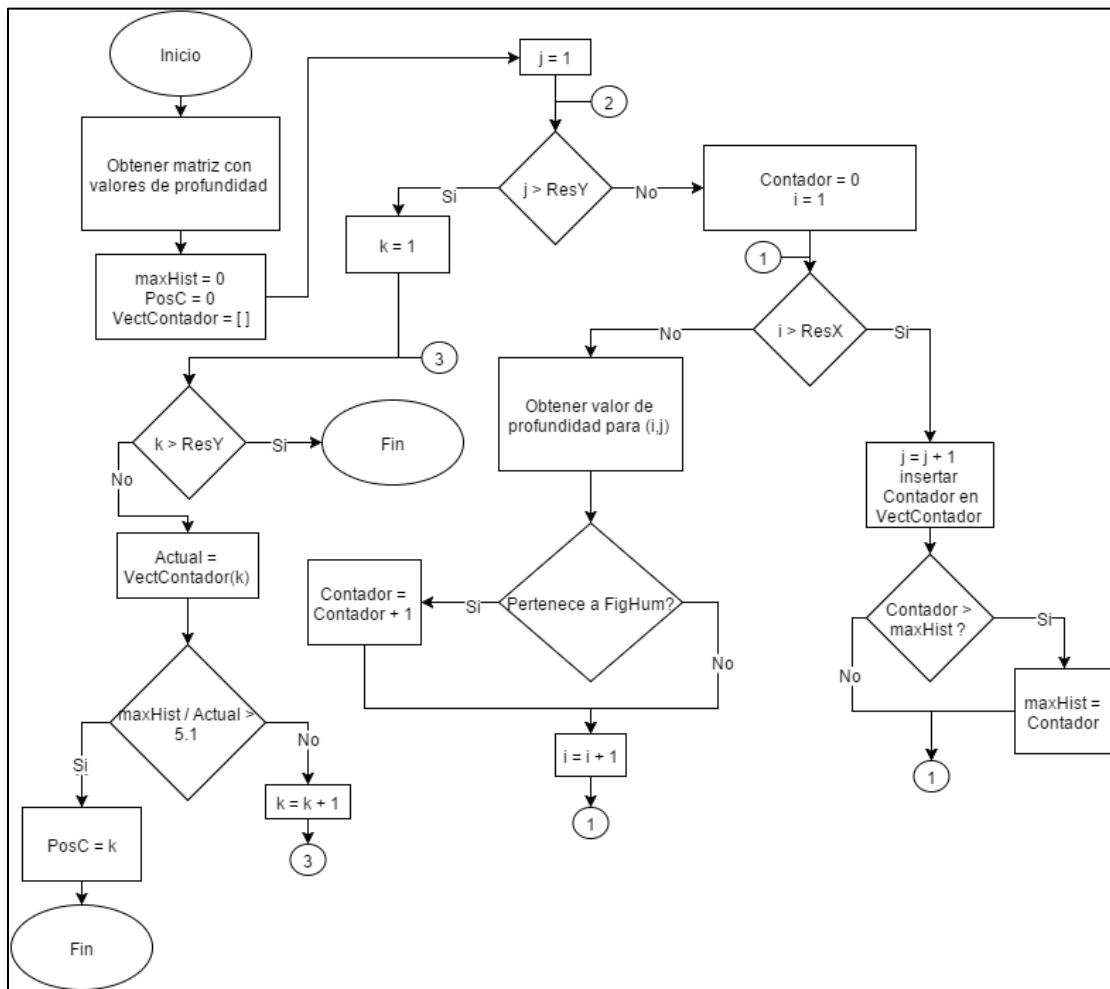


Figura 17: Diagrama de flujo para encontrar el punto de inicio del cuello utilizando el método de histogramas. (Creado por el autor en Visio)

Al realizar un recorrido de las filas y encontrar un punto en donde la razón entre el máximo y la cantidad actual de píxeles sea mayor que el valor umbral (5.1), se puede concluir que dicho punto representa el punto donde inicia el cuello. Este valor se guarda para poderse usar posteriormente.

Otro método para encontrar el punto en donde inicia el cuello consiste en utilizar los valores de profundidad que se pueden medir con el sensor. A lo largo de la cara, los valores de profundidad no cambiarán de manera importante, mientras que al llegar al cuello se verá un cambio más significativo. Teniendo el punto máximo de la cabeza, se puede medir la profundidad en una línea vertical de arriba hacia abajo, hasta encontrar un valor que cambie considerablemente con respecto al anterior.

Sabiendo que el punto máximo de la cabeza tiene una profundidad mayor a cualquier punto sobre la cara pero menor a la profundidad que se puede medir en el cuello, se puede encontrar el punto de inicio del cuello cuando la razón entre la profundidad medida y la profundidad del punto máximo de la cabeza es superior a uno. Esta relación se comprobó realizando medidas con el sensor (ver Apéndice 2). En una aplicación real, el sensor que se utiliza tiene una resolución y una precisión mayor, por lo que la diferencia entre las profundidades será más significativa, produciendo resultados más exactos.

Para poder detectar los objetos posteriormente utilizando este método, también deben poderse encontrar los extremos laterales de la cabeza. Para ello, se utilizan los puntos encontrados de la cara (inferior y superior) para recorrer la matriz que contiene los píxeles pertenecientes a la figura humana solamente entre los dos puntos de la cabeza y determinar el punto con una coordenada X máxima y mínima. Estos puntos serán los extremos laterales de la cabeza, que se utilizan como límites para determinar si el objeto detectado es parte de la cabeza o no. Si no es parte de la cabeza, es interpretado como un dispositivo móvil. El algoritmo para encontrar el punto mínimo de la cabeza y los extremos laterales se realizó utilizando el siguiente diagrama de flujo.

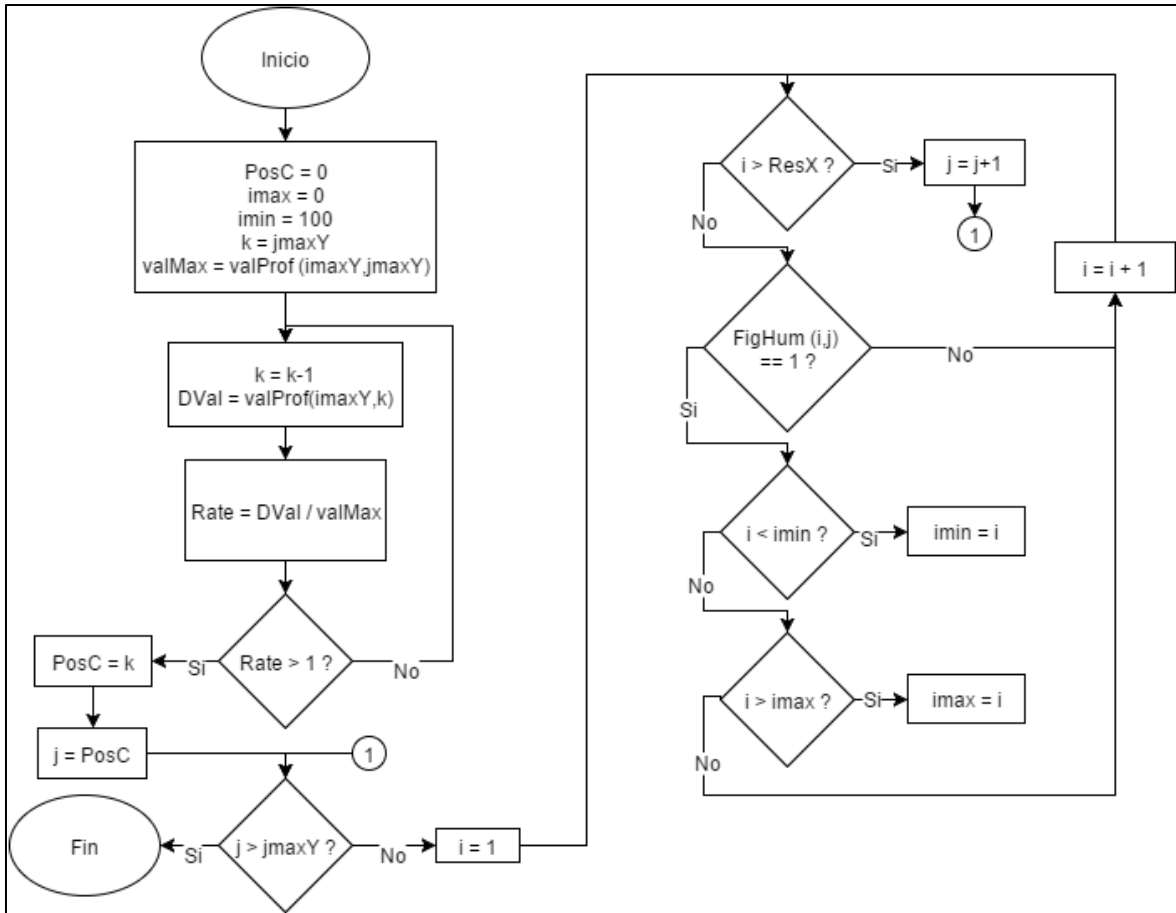


Figura 18: Diagrama de flujo para encontrar el punto de inicio del cuello y los extremos laterales de la cabeza utilizando el método de medición de profundidades (Creado por el autor en Visio)

Finalizada esta parte, se tiene el punto de inicio y final de la cabeza, los cuales se pueden utilizar para determinar si existe un objeto que se aproxima a ella. Este objeto será interpretado como un dispositivo móvil, y al mantenerse un tiempo determinado en cercanía de la oreja, representa el gesto que se quiere identificar. Al aplicar los métodos descritos, se obtuvieron los siguientes resultados.

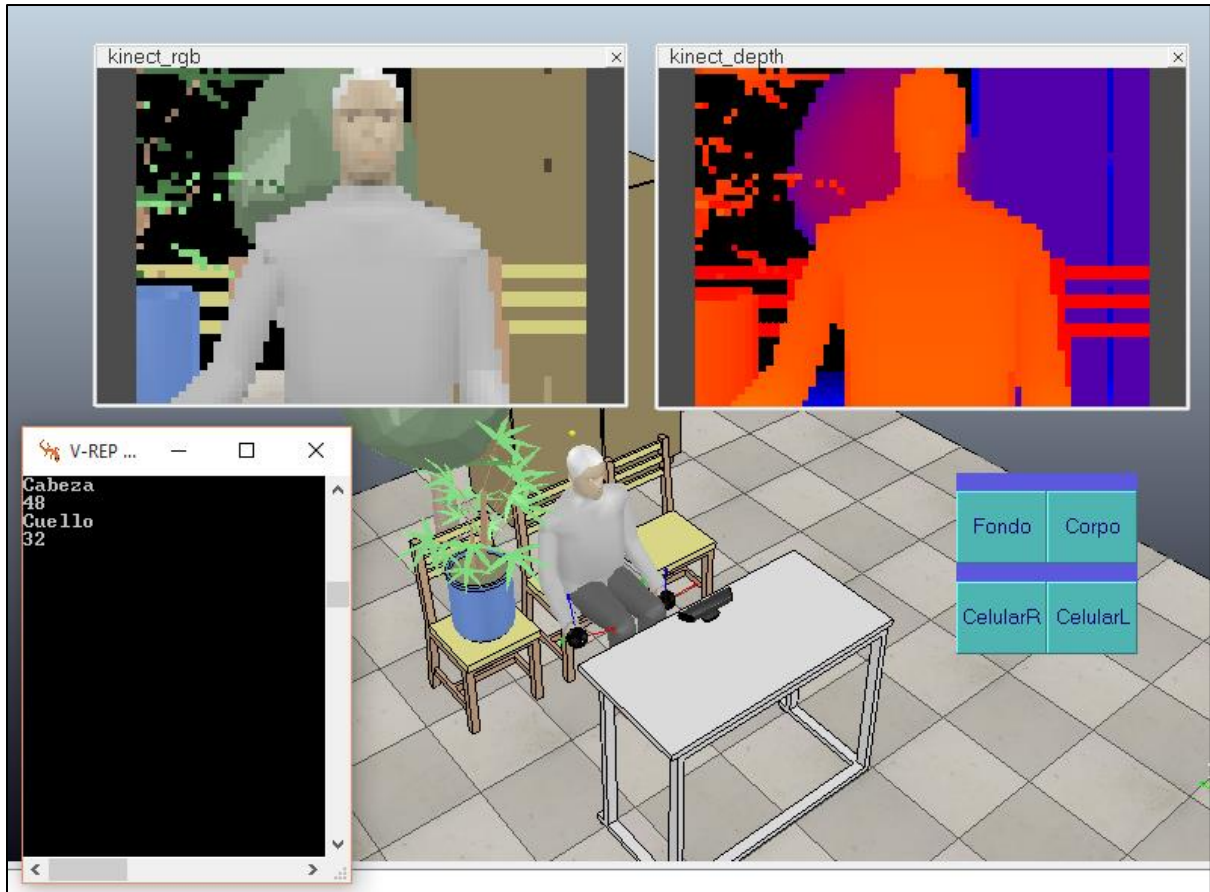


Figura 19: Resultados al calcular los puntos de inicio del cuello y final de la cabeza utilizando el método de número de píxeles por fila (Creado por el autor en programa V-REP)

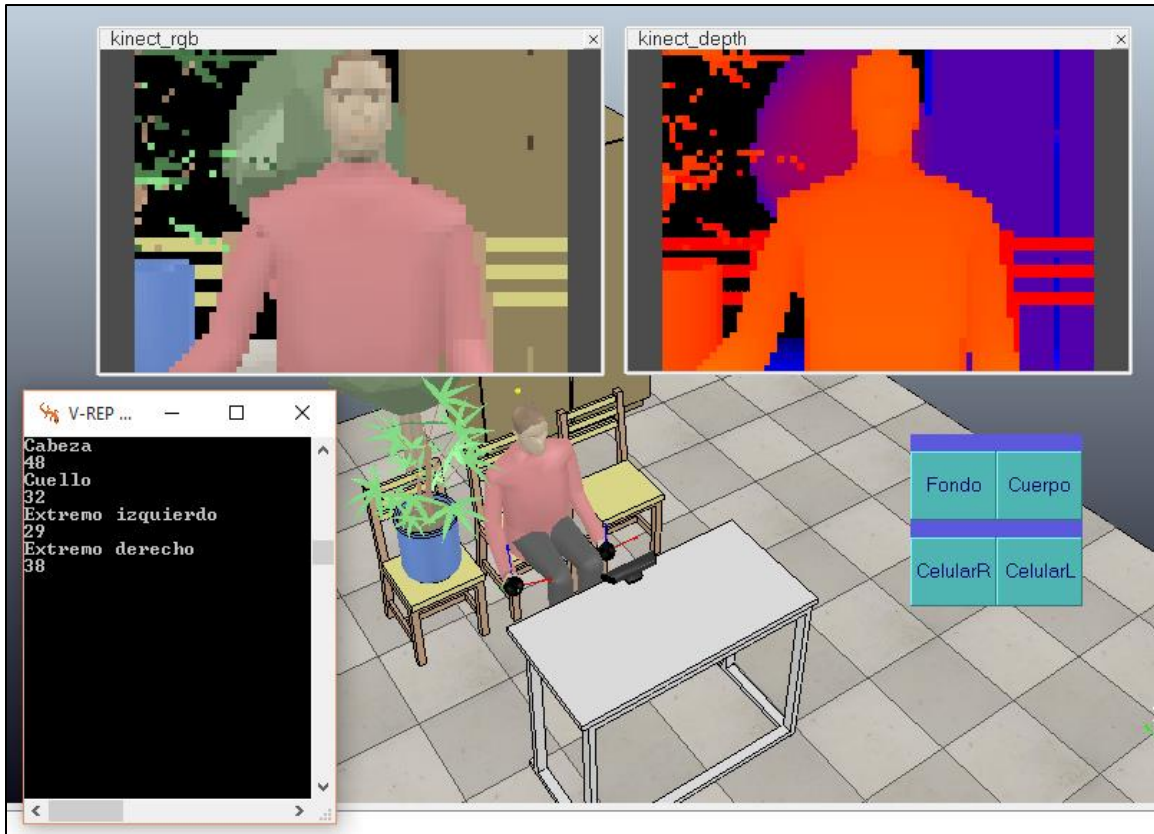


Figura 20: Resultados al calcular el punto de inicio del cuello, el extremo superior de la cabeza, y los extremos laterales de la cabeza utilizando el método de diferencia de profundidades (Creado por el autor en el programa V-REP)

Como se puede observar, en ambos casos, al tener el sensor posicionado a la misma distancia de la persona, se obtienen los mismos resultados para los dos métodos utilizados. Por lo tanto, se deberá comparar su funcionamiento en un caso real, utilizando los sensores en físico, para determinar cuál de los dos métodos produce mejores resultados.

4.1.4. Identificación del gesto de hablar por celular

Una vez que se conocen los límites de la cabeza, se puede analizar los alrededores de esta para determinar si existe algún objeto que no pertenece al fondo y que puede estar causando una distracción para el conductor. Para propósitos de este proyecto, cualquier objeto que se identifique, que tenga una dimensión significativa y que permanezca detectado por intervalo de tiempo, será interpretado como un dispositivo móvil que el conductor está utilizando. Aunque este sea el gesto que se quiere detectar, se considera que cualquier gesto que cause que el conductor separe las manos del volante será peligroso y representa una distracción para el

conductor. Por ello, esta aplicación podrá ser utilizada a futuro para detectar otro tipo de gestos similares que sean una amenaza para la seguridad en las vías.

Para hacer la detección del gesto se utilizaron dos métodos, debido a que también hay dos métodos distintos para encontrar los extremos límites de la cabeza. El primero de ellos, aprovecha el vector que contiene la cantidad de píxeles por fila que no pertenecen al fondo. El segundo método, examina las franjas laterales a la cabeza y detecta si hay un objeto en esa zona.

En ambos métodos se introduce las variables *Stat* y *PStat*, las cuales se utilizan para determinar el estado actual y anterior de detección. Estas variables inician con un valor de cero, y cuando se detecta que hay un objeto en la zona de interés, el valor de *Stat* se hace uno. El valor de *PStat* siempre almacena el estado previo, con el fin de imprimir un mensaje de detección únicamente cuando se da un cambio de estado.

El tiempo de detección se controla con una variable llamada *TD*, la cual determinará el tiempo en el que se ha mantenido un objeto detectado. Esta variable de tiempo se da en número de ciclos, por lo que debe modificarse de acuerdo con la velocidad de procesamiento para ajustarse al tiempo real que se quiere medir. En el caso de la simulación, para alcanzar 5 segundos, se deberá tener un *TD* de 50 ciclos seguidos de detección.

Para el primer método, se crea una nueva variable que almacena el conteo actual de píxeles que no pertenecen al fondo, para cada fila. Esto permitirá realizar una comparación entre los dos vectores para determinar si la cantidad de píxeles se ha incrementado de manera significativa. Dentro del ciclo infinito del programa, una vez que se cuenta con el valor del punto máximo de la cabeza y el punto de inicio del cuello, se establece un contador que se reinicia con el inicio de cada fila. Luego, se recorre cada fila completa, contando los píxeles que no pertenecen al fondo, iniciando desde el cuello hasta llegar al final de la cabeza. El valor del contador, al finalizar el recorrido de la fila, se compara con el valor correspondiente en el vector original que contiene todos los contadores para cada fila. Bajo la premisa de que en un tiempo determinado el tamaño de la cabeza de una persona no aumenta de manera importante, se asume que un incremento en el número de píxeles no pertenecientes al fondo para una fila, significa que existe otro objeto cerca. Por lo tanto, si el valor del contador supera por al menos cuatro píxeles al valor correspondiente del vector, se asigna un valor de uno a la variable *Stat*.

Si se cumple esta condición para cualquiera de las filas que se está analizando, el valor final de *Stat* será de uno. Si no se cumple la condición para ninguna de las filas, el valor de la variable permanece en cero. Una vez finalizado el recorrido de todas las filas, si el valor de *Stat* es uno, se incrementa el valor de la variable de tiempo. De lo contrario, la variable de tiempo se vuelve a hacer cero. Esto es dado a que se debe mantener el objeto al menos 5 segundos seguidos en la zona de detección para accionar el sistema de alarma, con el fin de evitar detecciones erróneas. Sin embargo, para cada objeto que se detecte, se imprimirá un mensaje en la pantalla. Para esto, si al finalizar el ciclo se tiene un valor de uno para variable la *Stat* y un valor de cero para la variable *PStat*, se imprimirá en pantalla el mensaje “Detectado”. Luego, si se tiene un valor de cero para la variable *Stat* y un valor de uno para la variable *PStat*, se imprimirá en pantalla el mensaje “Vuelta a normalidad”. Estas dos situaciones describen un cambio de estado, y los mensajes permiten tener un control de la detección que está realizando el sensor.

Finalmente, si la variable de tiempo *TD* es igual a 50 (se alcanzaron los 5 segundos de detección), se imprime en pantalla el mensaje “El usuario está utilizando un dispositivo móvil” y se activa el sistema de alarma que indica que se está detectado el gesto, alertando al usuario de las acciones peligrosas que está realizando. Al finalizar el ciclo, se iguala la variable *PStat* a la variable *Stat*, y a ésta última se le asigna un valor de cero. Para este método de detección, se utilizó el siguiente diagrama de flujo.

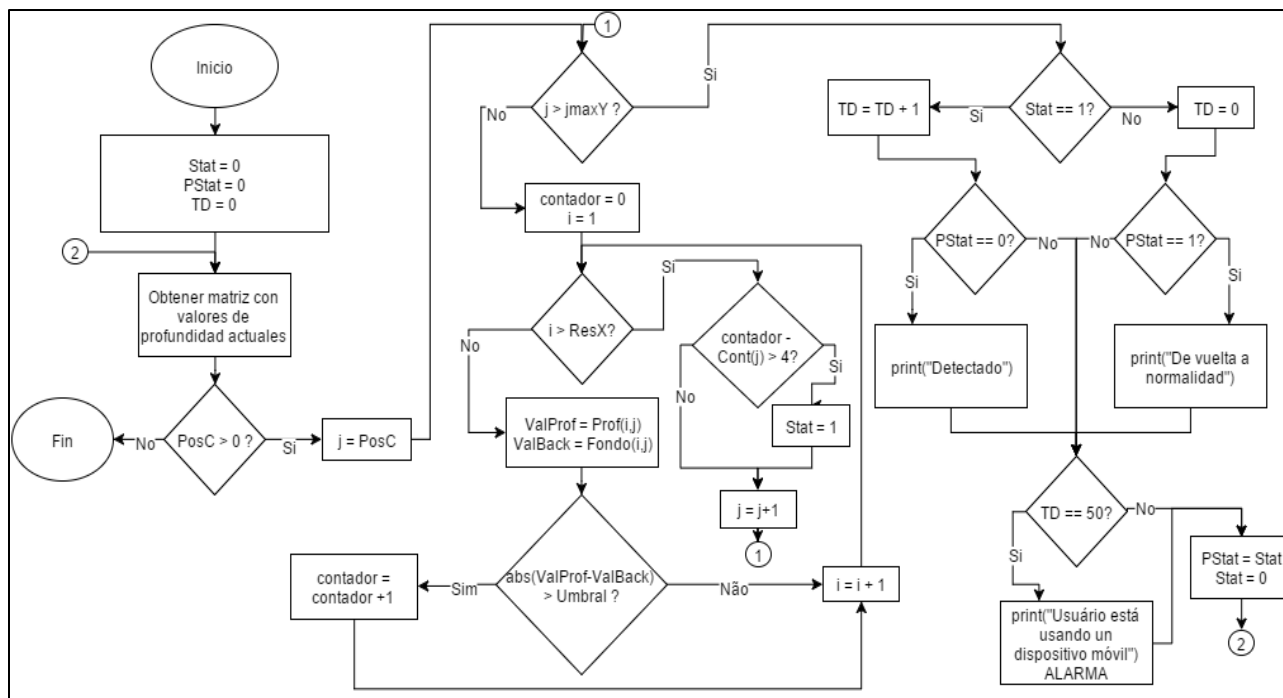


Figura 21: Diagrama de flujo para detectar el gesto de hablar por celular utilizando el método de número de pixeles por fila (Creado por el autor en Visio)

Para el segundo método de detección, también se utilizan las variables de *Stat* y *PStat* para determinar cuándo se da un cambio de estado. En este caso, se analizan las franjas laterales a la cabeza, y se contabiliza el número total de pixeles detectados a cada lado para determinar si se ha encontrado un objeto que no pertenece al fondo, ni es parte de la cabeza. Para ello, se tienen dos contadores, uno para el lado izquierdo y otro para el lado derecho. Esto se hace con el fin de evitar detecciones erróneas al contabilizar un número total de pixeles pero en lados distintos de la cabeza, dado que se asume que el objeto a detectar se encuentra únicamente de un lado a la vez.

Como ya se conoce el punto del cuello y el punto máximo de la cabeza, la lectura solamente se hará entre estos dos puntos en dirección vertical y se dividirá en dos partes en la dirección horizontal. Estas partes están determinadas por los extremos laterales de la cabeza que se calcularon anteriormente. Para ello se hace lectura de los datos del sensor desde el punto inicial hasta el extremo lateral izquierdo de la cabeza y luego desde el extremo lateral derecho de la cabeza hasta el punto final. En el recorrido se compara el valor actual de profundidad con el valor almacenado del fondo, y si se determina que el pixel no pertenece al fondo, se incrementa

el contador correspondiente al lado que se está leyendo (izquierdo o derecho). Una vez finalizados ambos recorridos para todos los valores de Y, se analiza si alguno de los contadores resulta ser mayor que 10 pixeles. Si se cumple esto para cualquiera de los dos casos, se le asigna un valor de 1 a la variable *Stat*.

El resto del programa es idéntico al realizado para el método anterior, ya que el valor de *Stat* y *PStat* determinan el mensaje que se imprime en pantalla, el incremento de la variable de tiempo, y la decisión de si se debe activar una alarma. Finalmente, al igual que en el caso anterior, se iguala la variable *PStat* a la variable *Stat*, y ésta última obtiene un valor de cero. Para el segundo método, el diagrama de flujo utilizado es el siguiente.

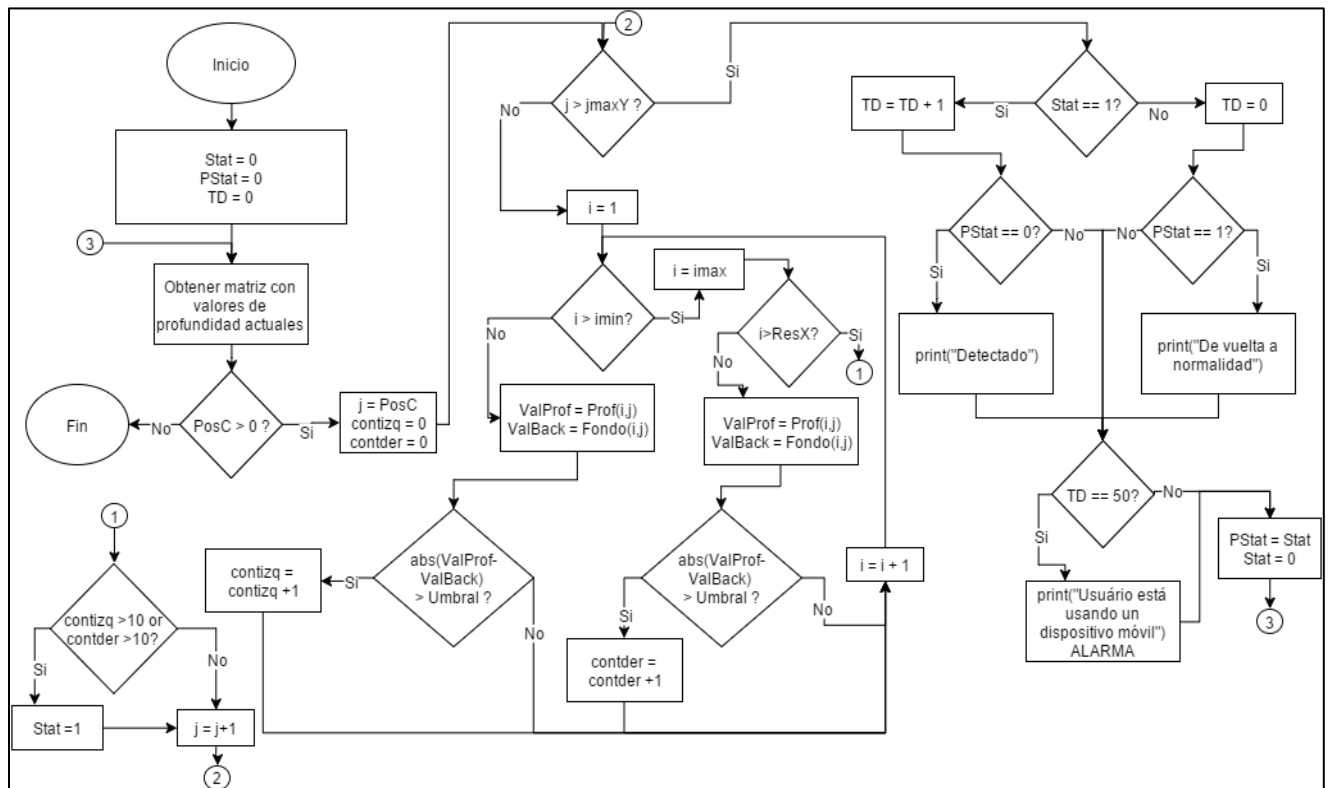


Figura 22: Diagrama de flujo para la detección del gesto de hablar por celular utilizando el método de análisis de franjas laterales a la cabeza (Creado por el autor en Visio)

Una vez realizado el programa para cada uno de los métodos se procedió a evaluar los resultados obtenidos. Para ambos métodos se obtuvieron los mismos resultados debido a que la

información que devuelve el programa es la misma para ambos casos, lo que cambia es la manera de interpretar los datos.

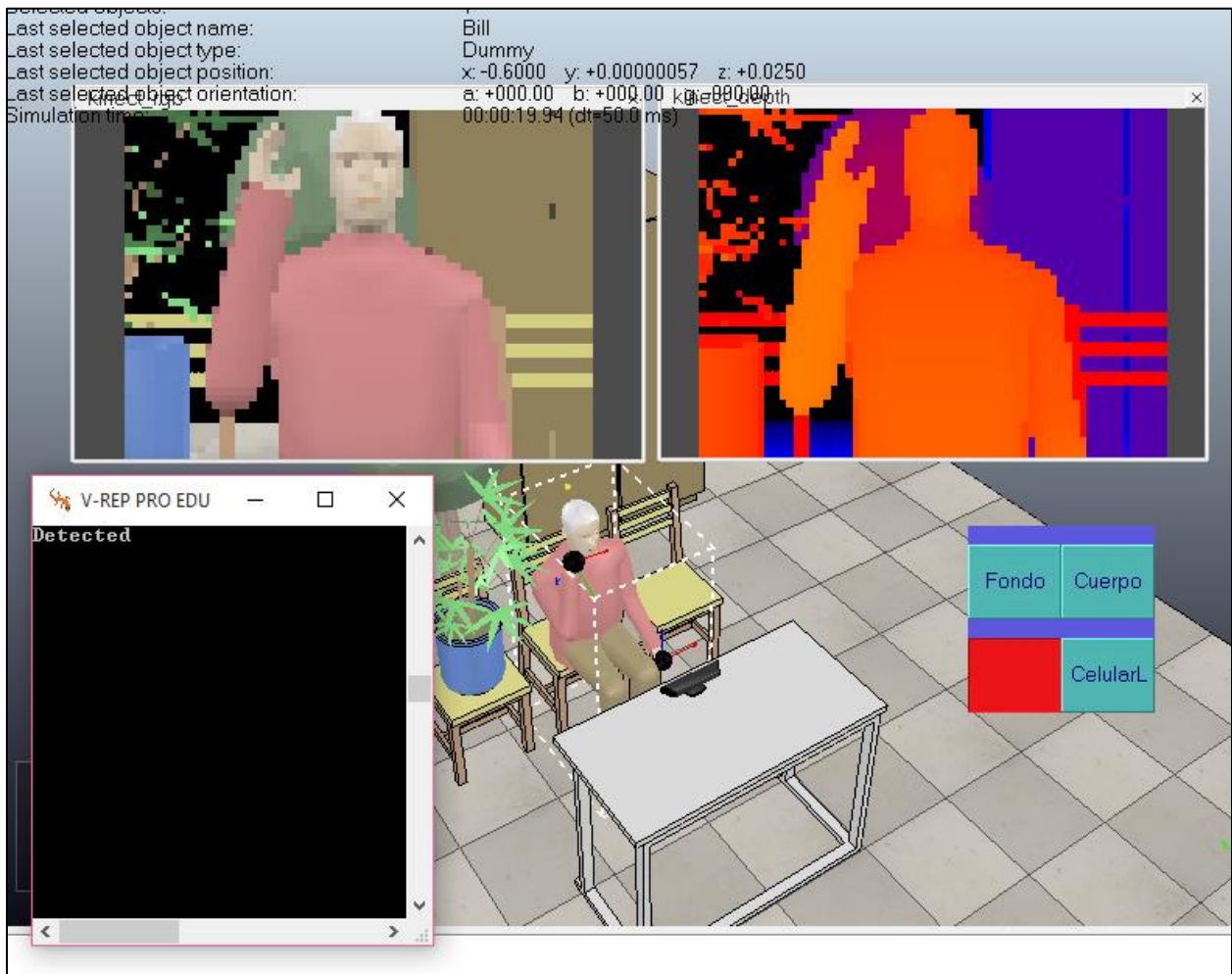


Figura 23: Resultado de detección de gesto por el método de análisis de conteo de pixeles por fila (Creado por el auto en V-REP)

En la figura 23 se puede observar cómo, al tener a la persona hablando por celular, el programa detecta la mano en cercanía a la cabeza e imprime un mensaje en pantalla. Este mensaje se imprime aun cuando la detección se ha dado por un tiempo muy corto, pero no activará una alarma hasta que se alcancen los 5 segundos de detección.

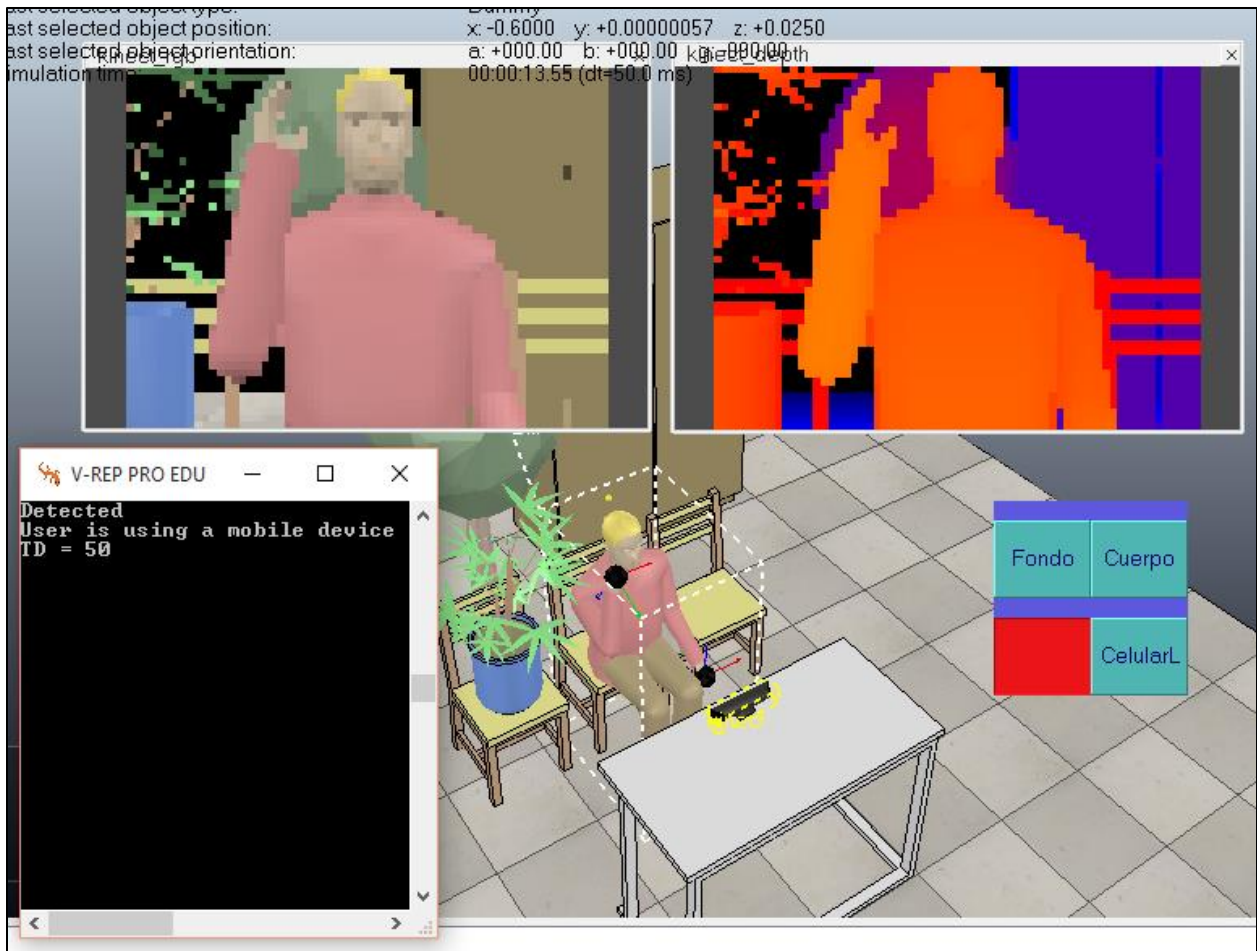


Figura 24: Resultado de activación de alarma al reconocer el gesto por más de 5 segundos seguidos (Creado por el autor en V-REP)

Al detectarse la acción de hablar por celular por más de 5 segundos, se imprime el segundo mensaje en pantalla “El usuario está usando un dispositivo móvil” y se activa una alarma sonora para alertar al conductor. Se puede observar que el tiempo de detección en número de ciclos es de 50, lo cual equivale a 5 segundos.

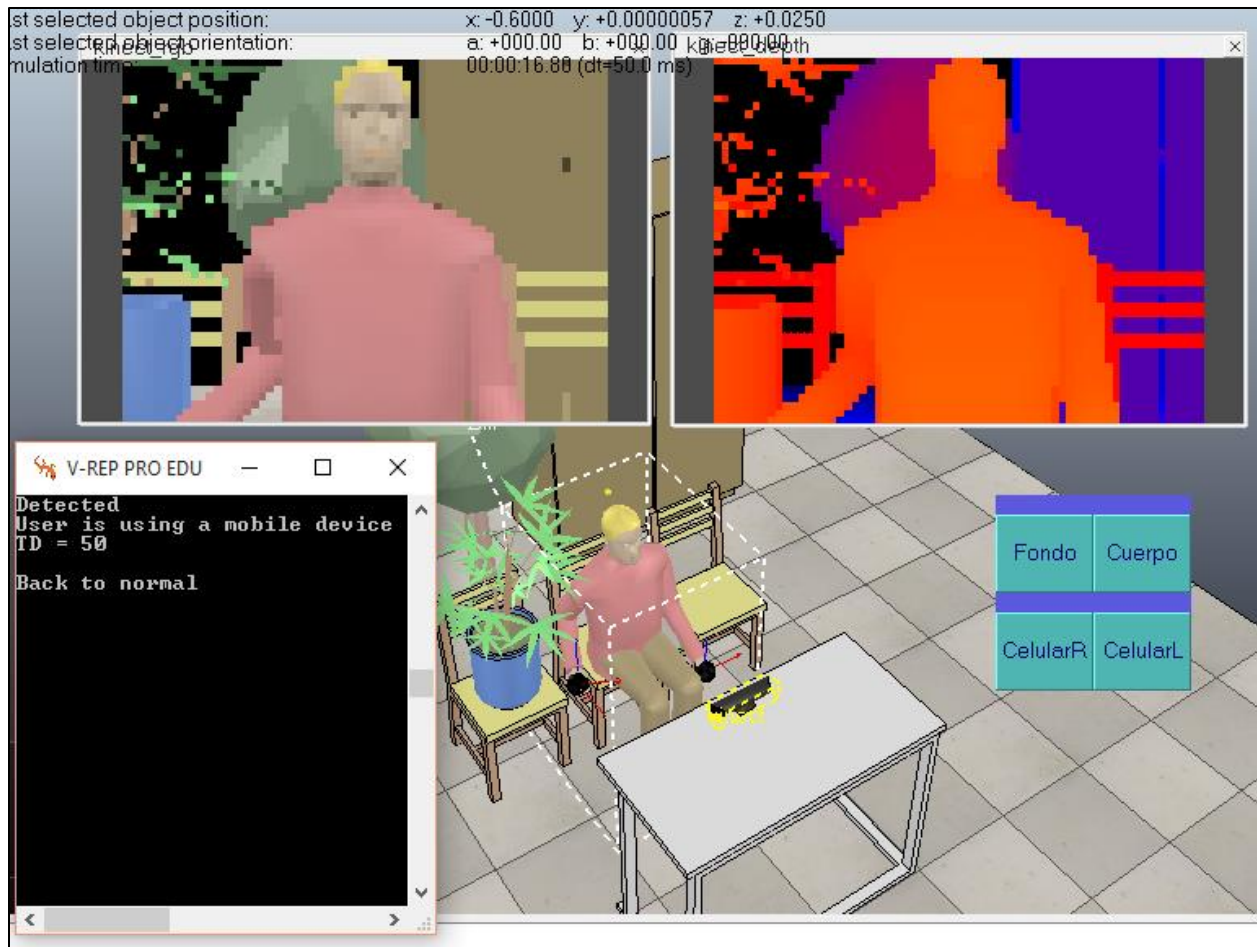


Figura 25: Resultado de detección de regreso a posición neutral después de haber detectado el gesto (Creado por el autor en V-REP)

Finalmente, al regresar a la posición neutral, el programa reconoce que ya no hay ningún objeto cercano a la cabeza que pueda representar un dispositivo móvil. Cuando el programa deja de detectar el gesto, se imprime un mensaje en pantalla “Vuelta a la normalidad”, indicando que el usuario ya no corre peligro y que el contador ha vuelto a cero.

4.2. Figura Humana (Bill)

Para propósitos de la simulación, también debió establecerse un código que controle el movimiento de la figura humana. En una aplicación real este código no será necesario ya que no se hará uso de un robot sino de una persona real para recolectar los datos. Al ser un modelo predefinido e importado, la figura de Bill (nombre que se le ha puesto a la figura humana que se utiliza) ya contiene un código propio. En este código se realizan acciones como determinar el color de la ropa, la piel y el cabello de Bill (que se eligen de manera aleatoria entre varias opciones), definir las variables que definen cada parte del cuerpo (en especial las articulaciones), e indicar en el programa que Bill se encontrará en posición sentada aunque sus manos pueden ser posicionadas sobre una mesa.

Para poder mover los brazos de Bill y ponerlos en una posición que asemeje la acción de hablar por celular, debe poderse manipular las articulaciones del brazo hasta llegar a la posición deseada. El brazo de Bill se compone de tres partes principales, el hombro, el codo, y el antebrazo. El hombro es representado por una articulación esférica, lo cual significa que puede tener rotaciones alrededor de cualquiera de los tres ejes principales. El codo se representa por medio de una articulación rotacional, la cual permite rotación alrededor del eje propio del codo. Esta última solamente tiene un grado de libertad. Finalmente, el antebrazo se utiliza como parte del cuerpo que puede responder ante las circunstancias que se simulan, como interacciones con objetos, colisiones, posicionamiento del brazo sobre una superficie, etc. Estas tres articulaciones están unidas entre sí por medio de juntas, las cuales corresponden a la parte superior del brazo y al antebrazo. Asimismo, se encuentran conectadas al torso, que será la base del cuerpo. En este modelo, las articulaciones y juntas de la mano no fueron simuladas, de modo que no se pueden utilizar para representar gestos ni movimientos específicos.

El movimiento del brazo como un conjunto se debe a la relación que se estableció entre las partes. En el simulador, se puede definir esta relación por medio de jerarquía. Un objeto con menor jerarquía que otro, se rige por el comportamiento del objeto “*parent*” (padre). A este objeto se le conoce como “*child*” (hijo). Al estar unidos, el movimiento del objeto con mayor jerarquía afecta el movimiento del objeto que se encuentra ligado a él. Sin embargo, existe una variedad de tipos de relación que se puede establecer entre los objetos. Para obtener el comportamiento que se requiere, se estableció una ecuación de cinemática inversa. Esta

estrategia permite que, al establecer un punto meta para el “efector final” (en este caso la mano), se pueda encontrar una orientación para el resto de articulaciones ligadas a la mano que permita que se alcance la posición meta. No obstante, los cálculos realizados por medio de cinemática inversa no siempre consiguen encontrar una solución, y algunas veces puede encontrar varias, lo cual puede llevar a que no se tenga un movimiento realista del brazo. Por lo tanto, se deben ajustar los parámetros para asegurar que la solución encontrada sea la más adecuada para la simulación del movimiento deseado.

Para establecer las ecuaciones que definen el movimiento se posicionó un objeto esférico en la mano del modelo humano (en la parte final del antebrazo). A este objeto se le dio el nombre de “*tip*” y será la representación del dispositivo móvil. Este punto se mantendrá fijo en el brazo. Luego, se posicionó otro objeto invisible con el nombre de “*target*”, el cual será capaz de moverse a la posición final deseada para representar el objetivo que se quiere alcanzar. Luego, utilizando las funciones del programa, se define una relación de cinemática inversa entre el “*tip*” y el “*target*”, de modo que, al moverse el “*target*” de posición, el “*tip*” seguirá su trayectoria para alcanzar en la medida de lo posible la posición final de éste, moviendo a su vez el resto de articulaciones a las que está ligado. No todas las posiciones alcanzadas por el “*target*” serán alcanzables por el “*tip*” debido a los límites de las articulaciones y a la combinación de ángulos necesaria para alcanzar la posición, pero se tendrá un resultado que se aproxime lo mejor posible. Para establecer esta relación, se define el torso de Bill como la base de la cadena cinemática, el “*tip*” como el efector final, y el “*target*” como la posición que se requiere alcanzar. Al realizar esto, cualquier movimiento del objeto que define el objetivo provocará que el programa calcule la combinación de ángulos necesaria para posicionar el efector final del brazo en dicha posición. Si la posición definida no es alcanzable por el brazo, se tendrá un movimiento aleatorio y continuo de las articulaciones tratando de encontrar una solución. Si existen varias soluciones posibles, se utilizará la más cercana, lo cual puede causar que se tenga una posición del brazo que no es realmente alcanzable por un brazo humano.

Con el fin de evitar que se obtengan posiciones irreales del brazo, se limitaron los rangos alcanzables por las articulaciones del hombro y del codo. De esta forma, el movimiento hacia la posición deseada se da de manera más precisa y controlada. Posteriormente, se utilizó la simulación para mover el “*target*” hasta una posición en donde se produzca una combinación de

ángulos de las articulaciones para que Bill se encuentre en posición de hablar por celular, como se muestra en la figura 26. Las coordenadas de esta posición final y la orientación necesaria para alcanzarla, se guardaron para ser utilizadas como posición y orientación meta. Finalmente, se utilizó un botón que, al ser presionado, mueve el “target” a la posición meta definida anteriormente, causando que el brazo adopte la posición final. Al ser presionado nuevamente, el brazo regresa a su posición inicial. Esto se realizó para ambos brazos, de forma que se pueda simular el gesto para cualquiera de ellos.

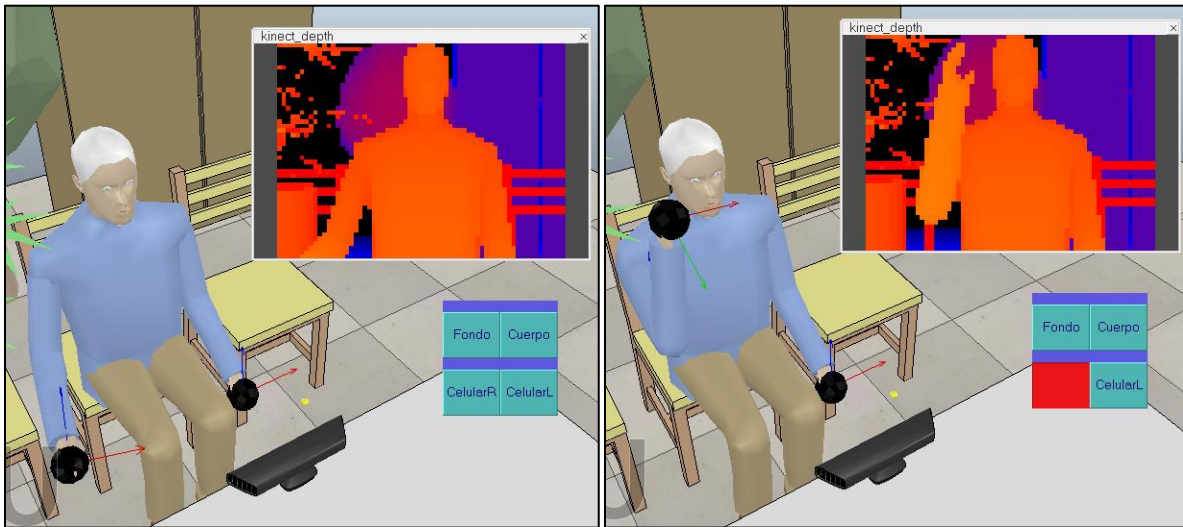


Figura 26: Posición inicial y final de Bill al presionar el botón que define la posición del brazo al hablar por celular (Creado por el autor en V-REP)

Para programar el movimiento, simplemente se almacenaron las coordenadas de la posición final y la posición inicial del brazo, y se definió un código para que, cuando se presione el botón, se traslade el objeto “target” a la posición almacenada, y cuando se presione de nuevo, se traslade a la posición inicial. Para poder tener el movimiento para ambos brazos, se definió una segunda posición meta, igual a la primera pero desplazada en el eje X, ligada al segundo botón. De esta forma se tiene un movimiento independiente para cada brazo.

Capítulo 5: Verificación de funcionamiento

Para poder verificar el funcionamiento adecuado de la aplicación que se desarrolló, se utilizó un sensor Kinect real para obtener datos que permitan determinar si la aplicación desarrollada realmente es efectiva. Debido a que se desarrollan una gran variedad de proyectos en el laboratorio que hacen uso de los vehículos, no se pudo hacer la verificación de datos en uno de ellos. En su lugar, se utilizó un ambiente similar al de la simulación, en donde se tienen algunos elementos que forman parte del fondo y una persona en posición sentada. Para ello, se utilizó el software de Microsoft SDK con un sensor Kinect II (Xbox One), y se tomó una serie de 80 imágenes (en formato .bmp) que incluyen el fondo sin personas en la escena e imágenes de la persona sentada con los brazos en posición neutral y utilizando el celular con cada una de las manos.¹

El Kinect II trabaja con información de profundidad en 13 bits en lugar de 8 bits (Kinect I). Esta información se organiza en los canales R y G de una imagen RGB. El canal R contiene los bits más significativos, mientras que el canal G contiene los bits menos significativos. El canal B tendrá siempre un valor de cero. Por lo tanto, la información del canal G estará en 8 bits y la información del canal R estará en 5 bits. La resolución de este sensor también es mucho mayor que la del Kinect I y que la resolución con la que se trabajó la simulación. Las imágenes adquiridas del Kinect II tienen una resolución de 512x424 píxeles.

Para trabajar las imágenes y observar el comportamiento del programa desarrollado se utilizó el software de Octave para procesamiento de imágenes. Este programa permite la manipulación de archivos de imagen y observar los resultados de manera sencilla. En este software se desarrollaron 5 programas (uno para cada etapa de la simulación y para cada uno de los métodos utilizados) en donde se comprobó la efectividad de la detección de gestos que se desarrolló anteriormente.

¹ La colección completa de imágenes se puede ver en <http://1drv.ms/1NtqUbE>

5.1. Substracción de fondo

Este es el primer programa que se utiliza para la validación de los datos. El programa de substracción de fondo se encarga de tomar las imágenes y realizar un pre-procesamiento para después poder obtener la máscara que define los pixeles que no pertenecen al fondo de la imagen. Como se mencionó, se tiene una serie de 80 imágenes en una secuencia de acción, por lo que se tienen varias de ellas que únicamente contienen los objetos del fondo, varias con la persona en posición neutral, y varias realizando el gesto de hablar por celular. Al tener información de profundidad utilizando “*point clouds*”, se tienen imágenes con bastante ruido, ya que la recepción de los datos por parte del sensor puede ser muy inestable. Por esta razón, se debe realizar un pre-procesamiento adecuado que permita analizar las imágenes efectivamente.

5.1.1. Pre-procesamiento de las imágenes

El objetivo del pre-procesamiento de la imagen es poder analizar los datos que ella contiene más fácilmente, ya que se elimina el ruido que pueda afectar este proceso. Existe una variedad de operaciones que se le puede realizar a una imagen en esta etapa. Las que se realizaron surgen de las necesidades específicas para este proyecto y de los resultados satisfactorios que se obtuvieron.

Lo primero que se debió hacer es identificar aquellas imágenes que forman parte del fondo. Estas se pueden encontrar al inicio y al final de la secuencia de imágenes, ya que este es el momento en el cual no hay persona en la escena. Para este caso, las primeras 20 imágenes corresponden al fondo. Sin embargo, debido al ruido en la lectura del Kinect, no todas son exactamente iguales. Si por alguna razón en particular se obtiene una lectura de cero, ya sea porque se tiene una superficie especular o demasiado fina para reflejar la luz, porque el objeto se encuentra muy lejos o porque se encuentra demasiado cerca para poderse detectar por el sensor; el valor en la imagen se verá como un punto negro. Estos puntos varían entre imágenes, por lo que se tendrán variaciones como se ve en el siguiente ejemplo.

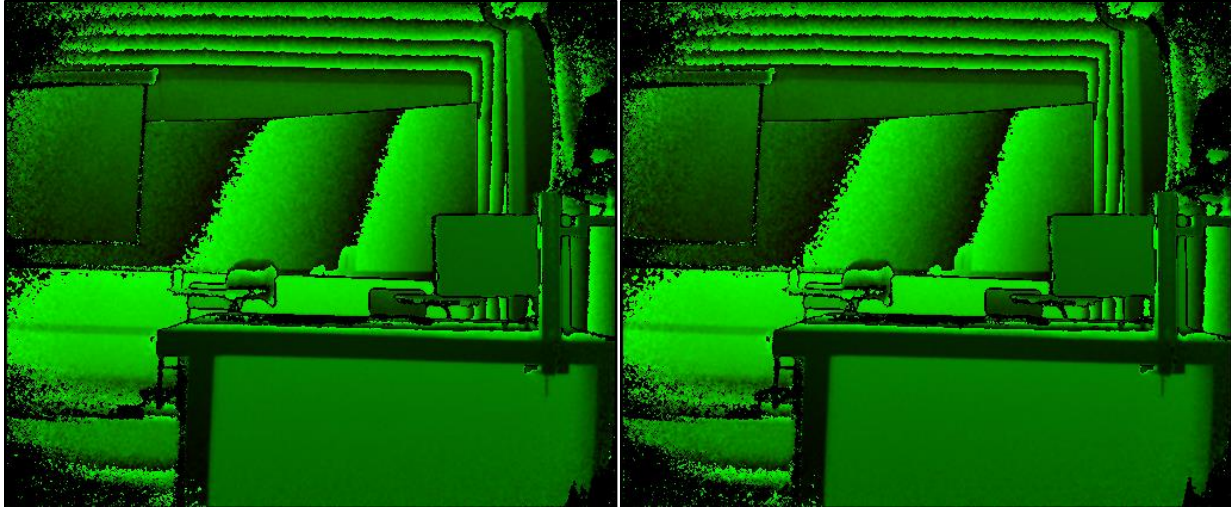


Figura 27: Dos imágenes distintas con información del fondo (Osório&Berri, 2015)

Para reducir los problemas de lectura por variaciones como ésta, se decidió hacer un promedio entre varias imágenes del fondo para obtener una que contenga los elementos que tienen todas ellas en común. Este es el primer paso del pre-procesamiento. Para ello se tomaron las primeras 20 imágenes (todas las que contienen información del fondo antes de que entre la persona en la escena) y se realizó un promedio entre ellas. La imagen que se obtuvo fue la siguiente.

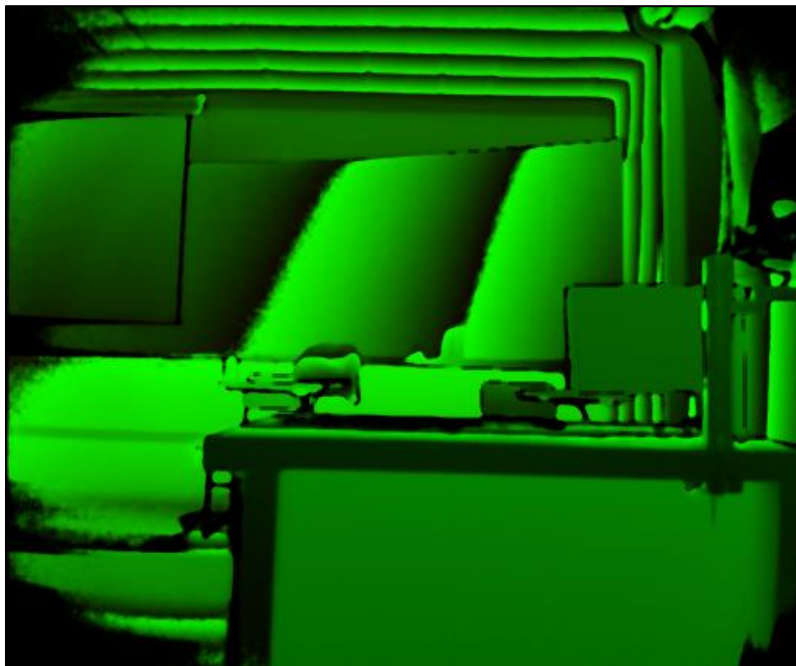


Figura 28: Promedio de 20 imágenes con información del fondo (Realizado por el autor en Octave)

Como se puede observar en la imagen resultado, al calcular el promedio se elimina mucho del ruido que se tenía originalmente y se tiene una imagen suavizada. Esto permite lecturas más precisas de los píxeles y una sustracción de fondo más limpia. Esta misma operación también se puede realizar para las imágenes con la persona en la escena. Debido a que se tiene una secuencia, existen también varias imágenes en las cuales la persona se encuentra en cada una de las posiciones de interés (neutral o hablando por celular). Para poder encontrar la máscara con los píxeles que no pertenecen al fondo, se utiliza la serie de imágenes que tiene a la persona en posición neutral. Para ello, se realizó nuevamente un promedio, esta vez de 12 imágenes que contienen a la persona en posición neutral. El resultado del promedio de imágenes fue el siguiente.

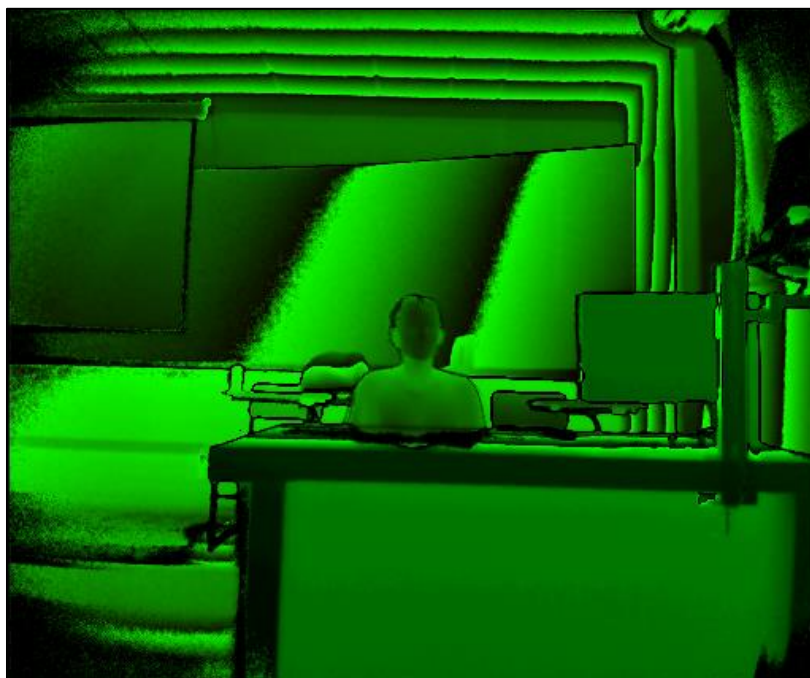


Figura 29: Promedio de las imágenes con la persona en posición neutral (Creado por el autor en Octave)

Al igual que con la imagen del fondo, al realizar el promedio de las imágenes en posición neutral se obtiene una imagen suavizada en la cual se pueden discernir claramente los elementos

que la componen. En ella también se puede ver claramente la forma de la persona que se encuentra en la escena.

En una implementación real de la aplicación, el sensor se encontraría mucho más cerca a la persona de lo que está para esta serie de fotos. Por lo tanto, se realizó un recorte a la imagen para poder tener la figura de la persona con mayor tamaño y eliminar elementos que puedan afectar los resultados que se obtienen. Para esto se utilizó una sección de 150x150 píxeles, en la parte central de la imagen. Se realizó este recorte tanto para el fondo como para la imagen con la persona en la escena.

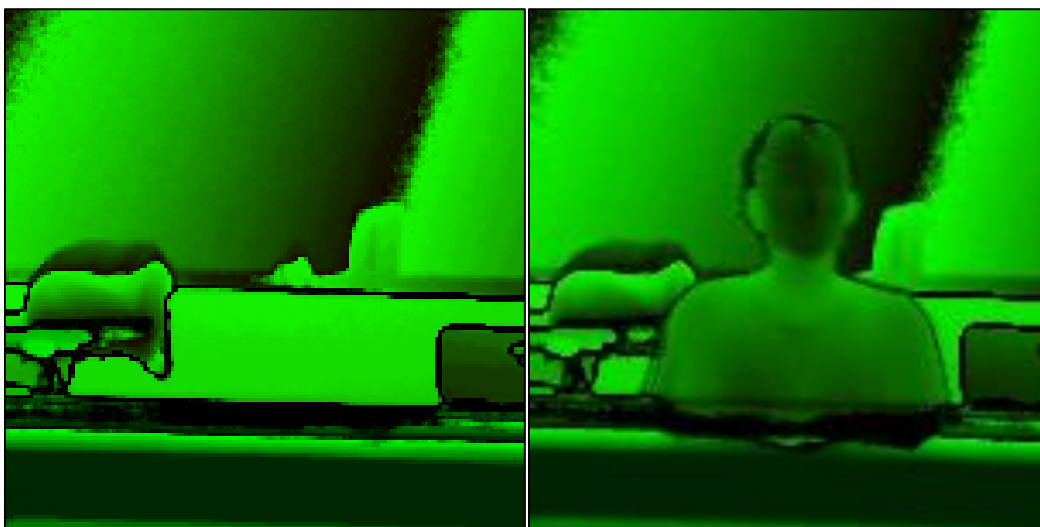


Figura 30: Recorte de la imagen de fondo y la imagen con la persona en posición neutral (Creado por el autor en Octave)

El último paso de pre-procesamiento es separar los canales R y G de las imágenes, ya que los datos de profundidad están repartidos en 13 bits de estos dos canales. El canal B no se utiliza ya que siempre tiene un valor de cero.

5.1.2. Obtención de la máscara

Una vez que se tienen las imágenes pre-procesadas, se puede realizar la comparación entre ellas que lleva a la substracción de fondo. Conociendo las dimensiones de la imagen, se recorre cada píxel en la imagen con la persona para compararse con la imagen de fondo. Teniendo los dos canales separados, se encontró el valor de profundidad con la siguiente fórmula.

$$\textit{Profundidad} = 256 * \textit{ValorCanalR} + \textit{ValorCanalG} \quad (5.1)$$

Esta fórmula se debe a que el canal R contiene los 5 bits más significativos de la profundidad, mientras que el canal G contiene los 8 bits menos significativos. Por lo tanto, los valores que se tienen en el canal rojo deben ser escalados para tener un valor equivalente a tener los 13 bits como un solo número.

Utilizando la fórmula 5.1 se obtiene el valor de profundidad para la imagen con la persona, y el valor de profundidad para la imagen de fondo. Luego, si alguno de estos dos valores es igual a cero (no se obtuvo datos de profundidad) para cualquier punto, el valor de la máscara en esa posición es igual a cero, ya que realmente no se puede obtener información acerca de si ese pixel pertenece o no al fondo.

Para los pixeles que tienen un valor diferente a cero, se resta el pixel de la imagen con la persona del pixel de la imagen de fondo. Luego, se utiliza un valor umbral para determinar qué tan similares son entre ellos. Esto se hace de esta forma, debido a los cambios entre imágenes, causados por el ruido y por los cambios leves en iluminación. El valor del umbral se determinó experimentalmente para la situación en específico con la que se está trabajando. Para esta colección de imágenes, se utilizó un valor umbral de 2050. Al realizar la resta entre los dos pixeles, si se obtiene un valor menor a 2050, quiere decir que el pixel de la imagen con la persona es muy similar al pixel de la imagen de fondo, y se considera que por lo tanto pertenece al fondo. Para un pixel que se determina es parte del fondo, se asigna un valor de cero en la posición correspondiente de la matriz máscara. Si el valor de la resta es un número mayor que 2050, significa que son datos muy distintos, y por lo tanto el pixel en análisis representa un objeto que no forma parte del fondo. A este pixel se le asigna un valor de 1 en la máscara. La máscara resultante que se obtuvo al finalizar el programa es una imagen binaria como la siguiente.



Figura 31: Máscara resultado de realizar substracción de fondo (Creado por el autor en Octave)

En esta imagen se puede ver claramente la forma de la persona en la escena y no se observan otros elementos que forman parte del fondo. Aunque hay algunos espacios en negro dentro de la imagen, se puede utilizar fácilmente para encontrar los puntos superior e inferior de la cabeza y así poder hacer la detección de gesto posteriormente, comparando esta imagen con una en la que la persona se encuentra hablando por celular. Para poderse utilizar en los siguientes programas, se guardó un archivo con esta imagen.

5.2. Obtención de la zona de interés

Una vez que se tiene la imagen que contiene únicamente la figura humana, se puede utilizar para obtener el punto superior de la cabeza y el punto de inicio del cuello. Para el segundo método, adicionalmente se obtienen los extremos laterales de la cabeza, utilizando esta imagen. Para encontrar estos puntos, se realiza el mismo análisis que se utilizó para la parte de simulación, de modo que se pueda comprobar el funcionamiento del programa que se realizó. Al igual que para la parte de simulación, se utilizaron dos métodos distintos para encontrar la zona de interés. El propósito de esta parte de validación es poder determinar cuál de los dos métodos produce mejores resultados, o si se deben utilizar ambos para tener una redundancia que reduzca la posibilidad de errores.

5.2.1. Método de análisis de histogramas

El primer método que se utilizó fue el de análisis de histogramas. Con este método se recorre cada fila de la imagen contando los píxeles que no pertenecen al fondo para luego utilizar esta información para determinar la posición del cuello. El primer paso para este programa es cargar la imagen de la máscara para poder utilizarla en el análisis. Luego, se tienen cuatro variables que se requieren para obtener los puntos de interés. Estas variables son: el valor en X del punto máximo de la cabeza, el valor en Y del punto máximo de la cabeza, la posición en Y de inicio del cuello, y valor máximo de píxeles en una fila que no pertenecen al fondo. La diferencia entre el programa para la simulación y el programa para Octave, es que en éste último, la lectura de la imagen se hace de arriba hacia abajo, por lo que un valor menor en Y significa un valor más arriba en la imagen.

Para encontrar el punto superior de la cabeza, se recorre la imagen binaria de la máscara y, si el valor leído es igual que 1 (lo que significa que es un elemento que no pertenece al fondo), compara el valor en i (número de fila) para ver si es menor que la variable $imaxCab$ (punto máximo en Y de la cabeza). Si es así, asigna el número de fila a esta variable, y asigna el número de columna actual a la variable $jmaxCab$ (posición en X correspondiente del punto máximo). Además, cada vez que encuentre en una fila un 1 en la matriz máscara, se incrementa el valor de un contador. Al finalizar la lectura de cada fila, el valor del contador se almacena en un vector llamado *Conteo*, el cual contiene todos los contadores para cada fila de la imagen. Además, al final de cada fila, se compara el valor del contador con la variable $MaxH$ (valor máximo de píxeles en una fila que no pertenecen al fondo), y si es mayor, se le asigna el valor del contador actual a la variable. Al finalizar toda la lectura, se tendrán las variables $imaxCab$, $jmaxCab$, y $MaxH$,

Teniendo el vector con todos los contadores de la imagen, se lee cada valor y se asigna a una variable Act que almacena el valor actual en análisis. Luego, se utiliza la siguiente fórmula para encontrar la relación entre el valor máximo y el valor actual.

$$Razón = \frac{MaxH}{Act} \quad (5.2)$$

Si el resultado de la ecuación 5.2 da un valor mayor que 8, se le asigna el número de la posición en el vector *Conteo* a la variable $PosC$ (posición de inicio del cuello). El número de la

posición en el vector será el mismo número de fila en el que se encuentra el inicio del cuello. Los valores de $imaxCab$, $jmaxCab$, y $PosC$ se almacenan para poderse utilizar en el siguiente programa, y además se utilizan para graficar los puntos sobre la imagen de la máscara y poder observar que se haya encontrado los puntos adecuados.

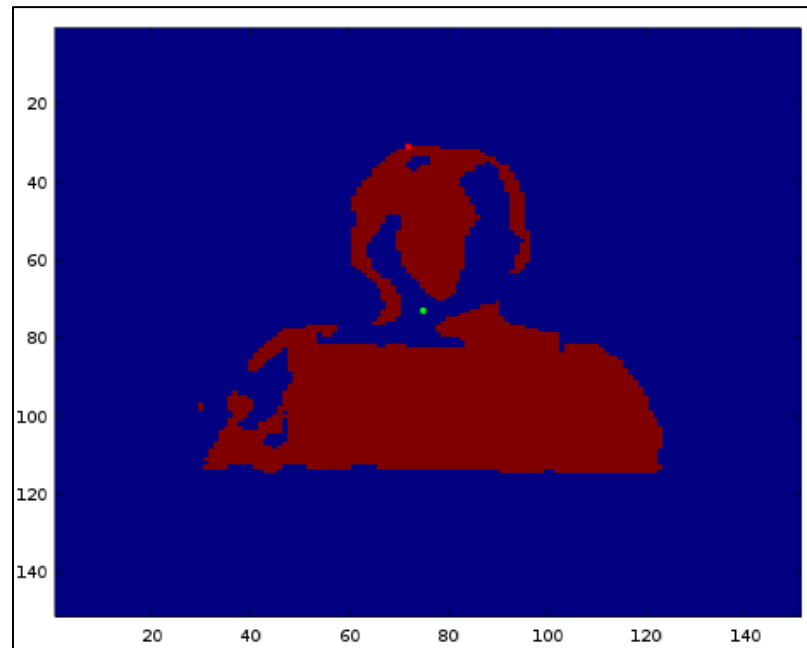


Figura 32: Punto máximo de la cabeza y punto de inicio del cuello graficados sobre la máscara de la figura humana (Creado por el autor en Octave)

De esta imagen se puede observar que se encontraron los puntos correspondientes al extremo superior e inferior de la cabeza de manera adecuada. La imagen resultante se observa en pseudo-color. Estos puntos se almacenan para ser utilizados en la parte de detección del gesto, ya que delimitan la zona de interés con la que se quiere trabajar.

5.2.2. Método de análisis de profundidad

Este es el segundo método utilizado para encontrar la zona de interés con la que se va a trabajar. Para este método, también se debe cargar la imagen de la máscara que se obtuvo en la etapa de substracción de fondo, pero además se debe obtener la imagen original que tiene a la persona para poder hacer un análisis de los valores de profundidad. Las variables que se requieren son nuevamente: posición en Y máxima de la cabeza, posición en X correspondiente al punto máximo de la cabeza, y posición de inicio del cuello. Además de estas se requiere: la

posición en X del extremo izquierdo de la cabeza ($jmin$), posición en Y correspondiente al extremo izquierdo ($ijmin$), posición en X del extremo derecho de la cabeza ($jmax$), y la posición en Y correspondiente al extremo derecho ($ijmax$).

Para encontrar el punto máximo de la cabeza se utiliza el mismo procedimiento que se utilizó para el método de los histogramas, en donde se recorre la máscara para encontrar el punto que tenga el valor mínimo de i (posición más alta en Y) y que además tenga un valor de 1 (no pertenece al fondo).

Conociendo el punto más alto de la cabeza, se extrae el valor de profundidad que corresponde de la imagen original con la persona, utilizando la fórmula 5.1 y se asigna a la variable $ProfPMax$. Este valor es la referencia que se utiliza para calcular una razón que determina cuándo se llega a la parte del cuello. La posición en i del punto máximo se asigna a una variable k . Luego, esta variable se incrementa (se desciende en dirección vertical) y se calcula el valor de profundidad nuevamente, utilizando la fórmula 5.1, asignándolo a una variable $Actual$. Con este valor se calcula una nueva razón con la siguiente fórmula.

$$Razón = \frac{Actual}{ProfPMax} \quad (5.3)$$

Se sabe que el punto máximo de la cabeza tendrá una profundidad mayor que cualquier punto sobre la cara, pero una profundidad menor a cualquier punto del cuello. Por lo tanto, se determina que la variable $Actual$ pertenece al cuello cuando la razón calculada con la fórmula 5.3 es mayor que 1. El valor de k se continúa incrementando hasta que la razón produzca un valor mayor que 1. Cuando esto sucede, se le asigna el valor de k (correspondiente a la fila en la que comienza el cuello) a la variable $PosC$.

Al finalizar, se tienen los valores de las variables que definen completamente la zona de interés, los cuales serán utilizados posteriormente para realizar la detección del gesto. Estos puntos también se utilizan para graficarse sobre la imagen de la máscara y así observar si se obtuvieron correctamente.

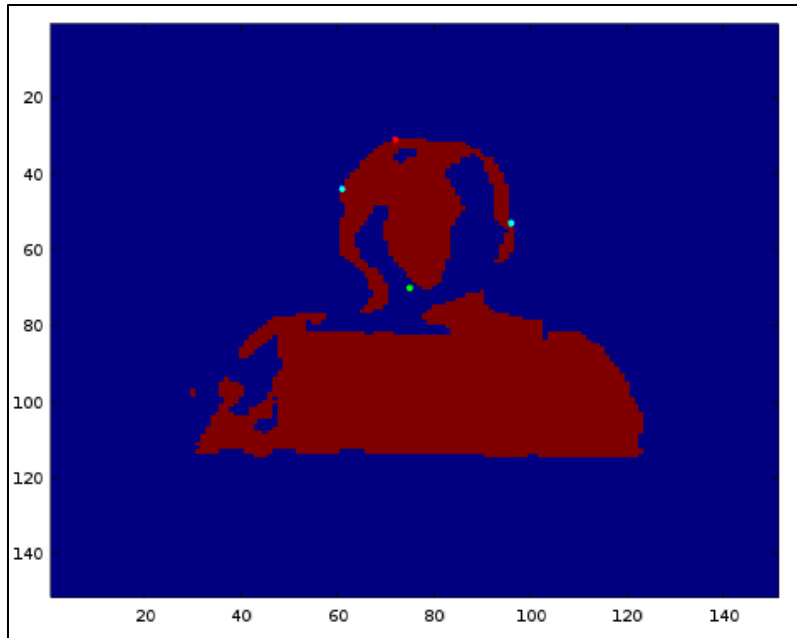


Figura 33: Punto máximo de la cabeza, extremos izquierdo y derecho de la cabeza y punto de inicio del cuello graficados sobre la máscara de la figura humana (Creado por el autor en Octave)

De esta imagen se puede observar que se obtuvieron satisfactoriamente los cuatro puntos de interés. A diferencia de la imagen obtenida con el primer método, esta imagen tiene el punto del cuello en la parte superior de éste. Esto se debe a que el análisis en este caso se realiza de arriba hacia abajo y el anterior se hacía en dirección contraria. Sin embargo, esto no presenta mucho problema a la hora de analizar la zona de interés para motivos de reconocimiento del gesto de hablar por celular, ya que el dispositivo móvil se encontrará cerca de la oreja, la cual está en la parte central de la cabeza y por lo tanto dentro de la zona de interés para los dos casos.

5.3. Detección del gesto de hablar por celular

Los últimos dos programas analizan los datos de varias imágenes para determinar si en ellas está presente el gesto que se quiere reconocer. En una implementación real de la aplicación, esta parte se realiza continuamente para poder detectar cuando se realiza esta acción en cualquier momento durante el periodo en el que se está conduciendo el vehículo. Para comprobación de datos, se analizan cuatro imágenes, una en posición neutral, una con las manos hacia el frente, una hablando por celular con la mano derecha y otra hablando por celular con la mano izquierda.

Esto se realiza con cada uno de los dos métodos utilizados para determinar cuál de ellos produce mejores resultados.

5.3.1. Método de análisis de histogramas

El primer método utilizado corresponde al primer método que se utilizó para encontrar los puntos límite de la cabeza. De esta forma, se puede aprovechar la información adquirida en el programa anterior para la parte de detección. Por lo tanto, el primer paso en este programa es llamar al programa que calcula la posición de la cabeza. Luego se carga la imagen de fondo que se tiene almacenada, para poderse utilizar en la substracción de fondo. Al igual que en la simulación, se utiliza una variable llamada *Stat* que determina el estado de detección. Esta variable debe ser inicializada en cero. Un valor de uno en la variable de estado, significa que se detectó el gesto, mientras que si se mantiene el valor en cero, no hubo detección.

El siguiente paso es cargar la serie de imágenes que se va a utilizar para realizar el análisis. Para esta parte se elige una serie de imágenes para cualquiera de los casos mencionados anteriormente. Al igual que para la primera substracción de fondo, se realiza un promedio de varias imágenes continuas para suavizar la imagen y poder analizarla más fácilmente. Luego, la imagen se recorta utilizando el mismo criterio con el cual se recortó el fondo, para obtener únicamente la parte que contiene a la persona. Posteriormente, se extraen los canales de color para poder hacer una lectura adecuada de la información de profundidad.

La substracción de fondo se realiza de la misma forma que se realizó para la primera parte. Se recorre toda la imagen, comparando los valores de profundidad calculados con la fórmula 5.1. Si cualquiera de los valores (de la imagen en análisis o de la imagen de fondo) es igual a cero, la nueva máscara tendrá un valor de cero en esa posición. Luego, se restan los valores y, si la diferencia es menor que el valor umbral de 2050, se asigna un valor de cero en la máscara. De lo contrario, se asigna un valor de 1. Al finalizar el recorrido se tiene una imagen binaria como las que se presentan a continuación.

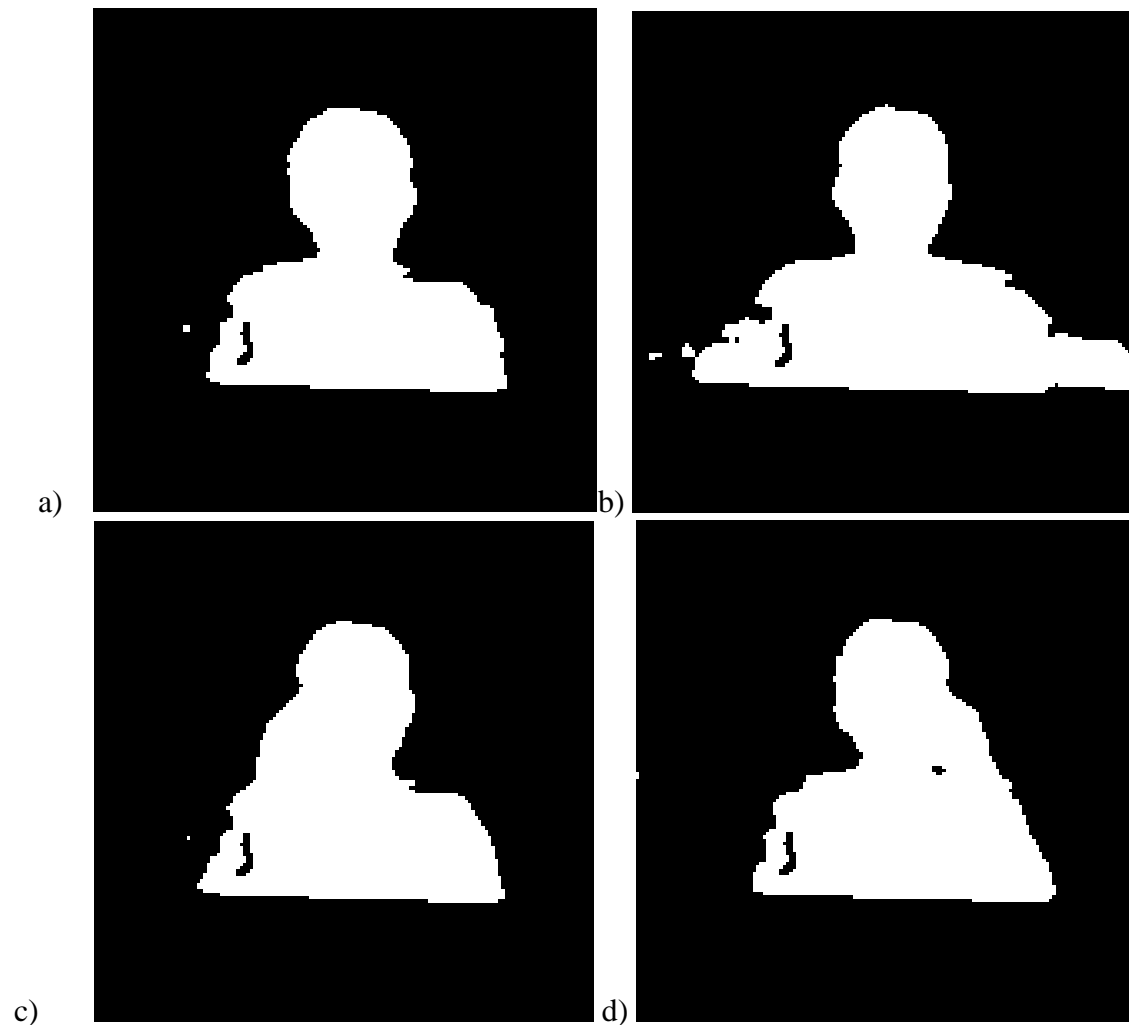


Figura 34: Substracción de fondo para diferentes casos (a) persona en posición neutral, b) con las manos al frente, c) utilizando el celular con la mano derecha, d) utilizando el celular con la mano izquierda) (Creado por el autor en Octave)

En estas imágenes se puede observar claramente si existe algún objeto cerca de la oreja. Estos datos se utilizan para poder realizar la detección del gesto, comparando la imagen actual que se está analizando, con la imagen anterior que se utilizó para calcular los límites de la cabeza.

Para este método en particular, se utiliza el vector *Conteo* que se tenía anteriormente. Este contiene la cantidad de píxeles en blanco (no pertenecientes al fondo) para cada una de las filas de la imagen. La zona de interés con la que se trabaja es únicamente la parte de la cabeza, por lo que se va a recorrer el vector *Conteo* solamente entre los puntos inferior y superior de la cabeza. Luego, se analiza la imagen actual entre estos dos puntos *imaxCab* y *PosC* que definen la

coordenada Y de la parte superior de la cabeza y del inicio del cuello. Se utiliza un nuevo contador que se reinicializa al comienzo de cada fila, para contar la cantidad de píxeles en la fila que tienen valor de 1. Al finalizar la lectura de cada fila, se compara el valor del contador con el valor en el vector *Conteo* en la posición correspondiente al número de fila que se está leyendo. Si el contador supera al valor del vector en al menos 30 píxeles (valor determinado experimentalmente), significa que se tiene un objeto cerca de la cabeza que no pertenece al fondo. Si esto se cumple para cualquiera de las filas, el valor de *Stat* se hace 1 (se dio la detección).

Finalmente, si el valor de *Stat* es igual a 1 cuando se finaliza el recorrido de todas las filas, se imprime en pantalla el mensaje “PELIGRO: El usuario está utilizando un dispositivo móvil”. De lo contrario, se imprime un mensaje que dice, “Condiciones seguras”. Para este caso, no se utilizó la variable *PStat* que almacena el estado anterior de detección, ya que se realiza una única lectura de imagen. Sin embargo, en una implementación real de la aplicación, se deberá utilizar esta variable, ya que, al igual que en la simulación, se estará haciendo una lectura continua de los datos del sensor.

5.3.2. Método de análisis de zona alrededor de la cabeza

Este método analiza la zona de alrededor de la cabeza según los límites encontrados con el método de análisis de profundidades. Con este método, se determina si existe un objeto en las franjas laterales a la cabeza. El problema que se puede presentar con este método de detección es que la persona puede moverse en cierta medida hacia los lados o inclinar la cabeza durante el tiempo en el que está conduciendo. Esto afectaría la detección, ya que se asume una posición fija de la cabeza, determinada por la posición en la que se encuentra cuando se hace la medición de los puntos límites de ésta. Ya que se trabaja con un valor umbral, un cambio leve en la posición no causa una detección errónea. Sin embargo, un corrimiento considerable si se interpreta como la presencia de otro objeto en la escena.

La primera parte de este programa es idéntico al del método de análisis de histogramas. Se carga la imagen que contiene los datos del fondo. Posteriormente, se recibe la serie de imágenes que representa la situación actual y se obtiene el promedio. Luego, se recorta para tener únicamente la parte de la persona y se extraen los canales de color R y G. Con la

información de profundidad se hace la substracción de fondo para obtener imágenes como las que se presentan en la figura 34.

La diferencia entre los dos programas se da en el momento de detección. Para este método, se tienen dos contadores, uno para el lado izquierdo y otro para el lado derecho. La imagen máscara se recorre desde *PosC* hasta *imaxCab*, los cuales son los límites superior e inferior de la cabeza. Luego, se hace un recorrido en dirección horizontal desde el inicio de la imagen hasta el punto *jmin*, el cual representa el extremo izquierdo de la cabeza. Si se detectan pixeles con valor de 1 (que no pertenecen al fondo) en este rango, se incrementa el contador del lado izquierdo. Seguidamente, se analiza el lado derecho, desde el punto *jmax* (extremo derecho de la cabeza) hasta el final de la imagen. Si se encuentran pixeles con valor de 1 en este rango, se incrementa el contador del lado derecho.

Al finalizar el recorrido, se analiza el valor final de los contadores. Si cualquiera de ellos tiene un valor mayor a 100 pixeles (valor determinado experimentalmente), entonces quiere decir que hay un objeto cerca de la cabeza, y se asigna un valor de 1 a la variable *Stat*. Este objeto se interpreta como un dispositivo móvil. Finalmente, si la variable *Stat* tiene un valor de 1, se imprime el mensaje “PELIGRO: El usuario está utilizando un dispositivo móvil”. De lo contrario, se imprime el mensaje “Condiciones Seguras”, al igual que con el método anterior.

5.3.3. Resultados obtenidos

Al utilizar los programas de detección del gesto para diferentes situaciones posibles, se obtuvieron buenos resultados. A continuación se van a presentar las pruebas que se realizaron utilizando los dos métodos para una serie de imágenes de cada tipo.

Inicialmente, se probó el programa utilizando algunas imágenes de la persona en posición neutral. Estas imágenes son distintas a las utilizadas en la primera parte (para hacer la primera substracción de fondo). Algunas de las imágenes utilizadas para esta parte se presentan a continuación.

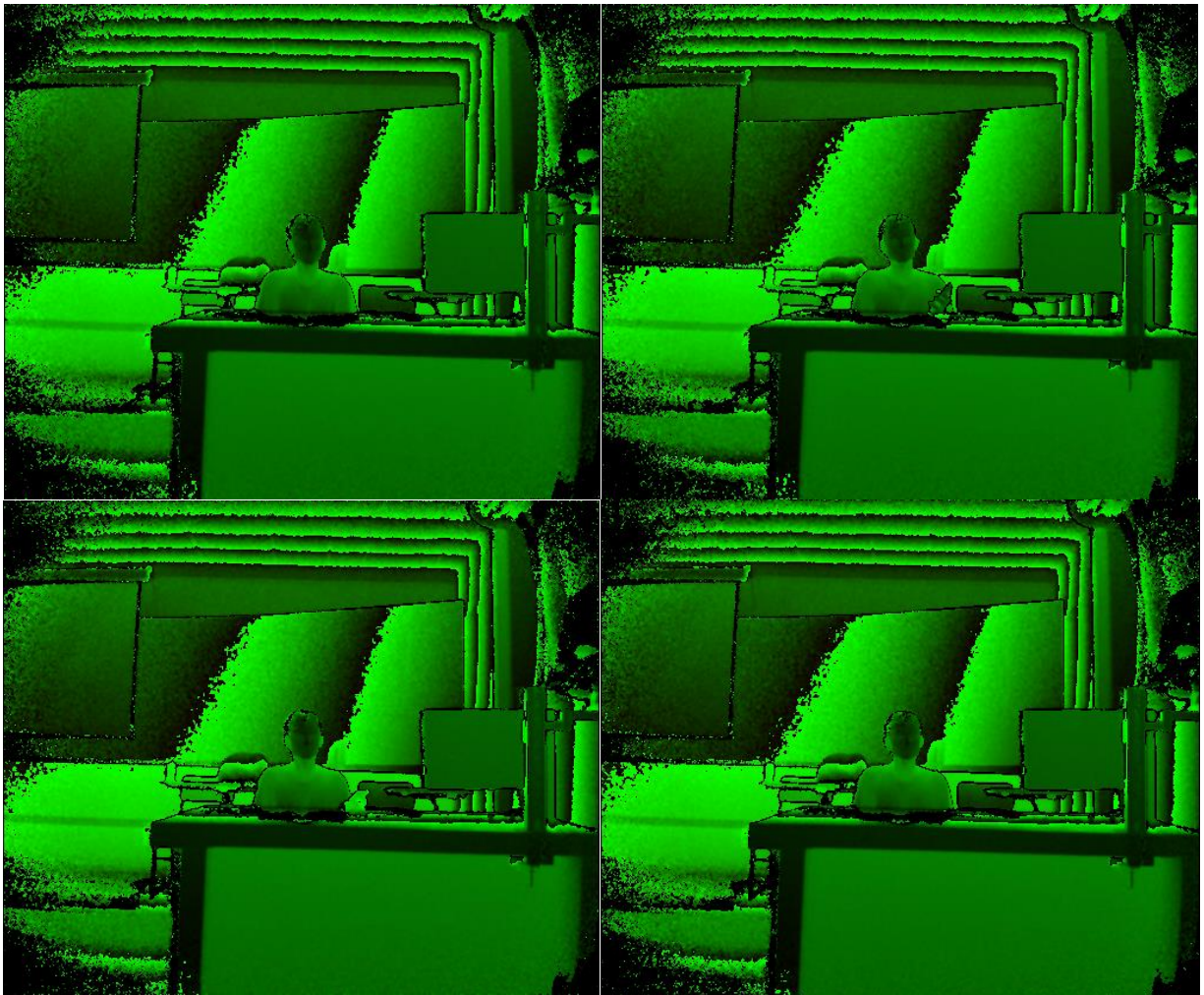


Figura 35: Imágenes pertenecientes a la serie con la persona en posición neutral utilizada para probar el algoritmo (Osório & Berri, 2015)

De las imágenes se puede observar que, aunque en algunas de ellas se tiene el celular en la mano del usuario, éste no se encuentra en los alrededores de la cabeza, por lo que no va a ser un problema a la hora de la detección. Al realizar un promedio de las imágenes, este elemento desaparecerá y solo se tendrá la figura del cuerpo de la persona con los brazos abajo. El resultado de correr el programa para esta serie de imágenes es el siguiente.

Método de análisis de histogramas (DetecciónHist)

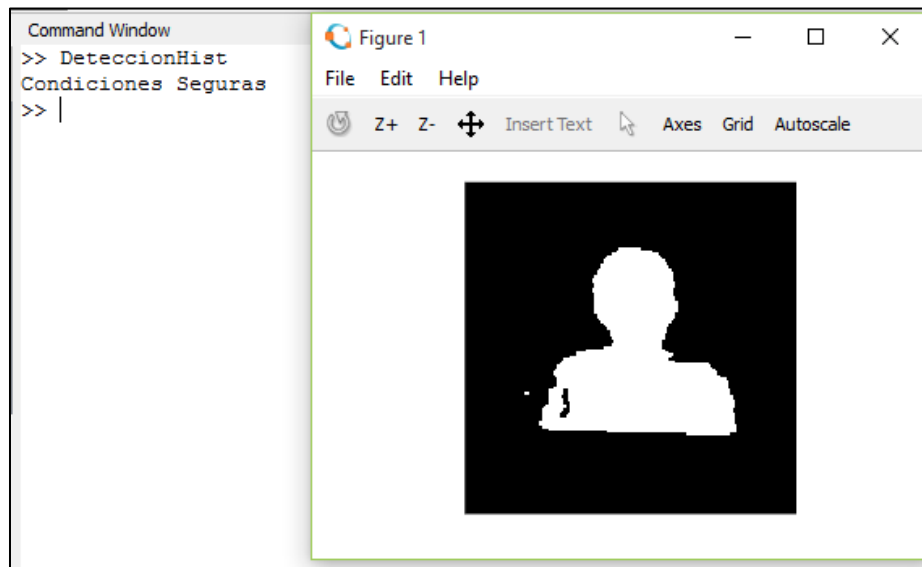


Figura 36: Resultado de correr el programa DetecciónHist (método 1) con una serie de imágenes en posición neutral (Creado por el autor en Octave)

Método de análisis de zona alrededor de la cabeza (DetecciónProf)

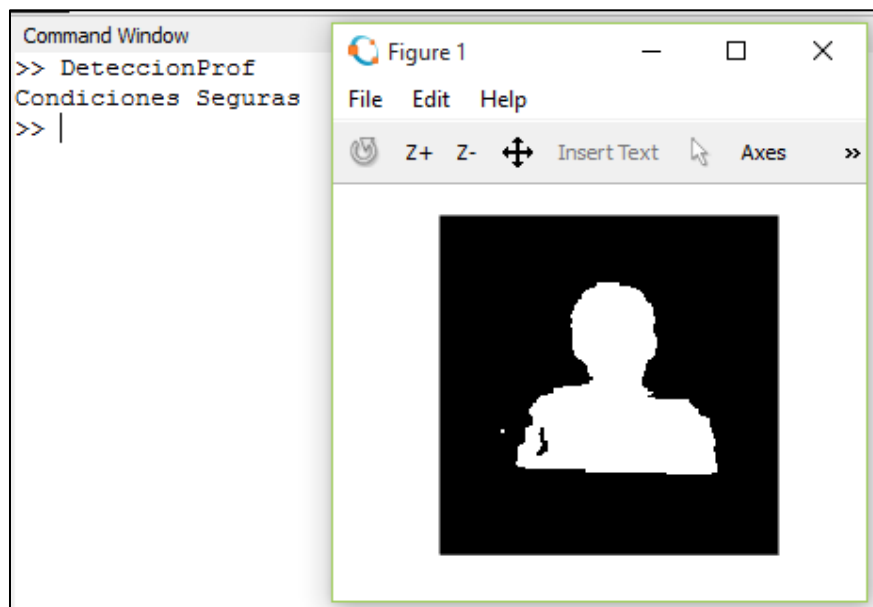


Figura 37: Resultado de correr el programa DetecciónProf (método 2) con una serie de imágenes en posición neutral (Creado por el autor en Octave)

La segunda prueba se realizó con imágenes en las que la persona tiene los brazos hacia el frente pero no cerca de la cabeza. Algunas de estas imágenes se muestran a continuación.



Figura 38: Imágenes pertenecientes a la serie con la persona en posición neutral y sus brazos hacia el frente (Osório & Berri, 2015)

Como se observa en estas imágenes, la posición de las manos cambia bastante, pero nunca se acerca a la zona de la cabeza. Por esto, al realizar el promedio de imágenes se va a obtener una en donde no se aprecia bien la posición de las manos, pero se puede distinguir claramente la figura de la cabeza y el cuerpo, la cual se requiere para el análisis y detección del gesto. Al utilizar el programa con cada uno de los métodos sobre estas fotos se obtuvieron los siguientes resultados.

Método de análisis de histogramas: (DetecciónHist)

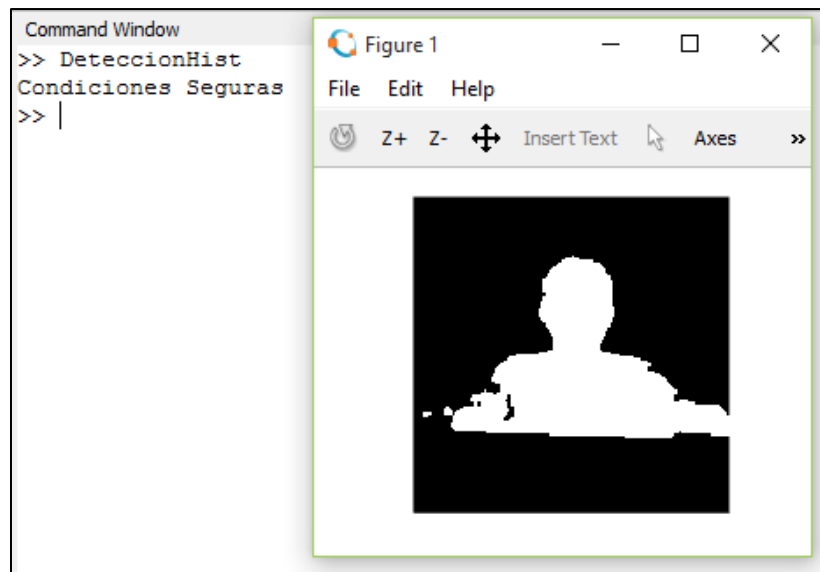


Figura 39: Resultado de correr el programa DetecciónHist (método 1) con una serie de imágenes con la persona en posición neutral y con los brazos hacia el frente (Creado por el autor en Octave)

Método de análisis de zona alrededor de la cabeza (DetecciónProf)

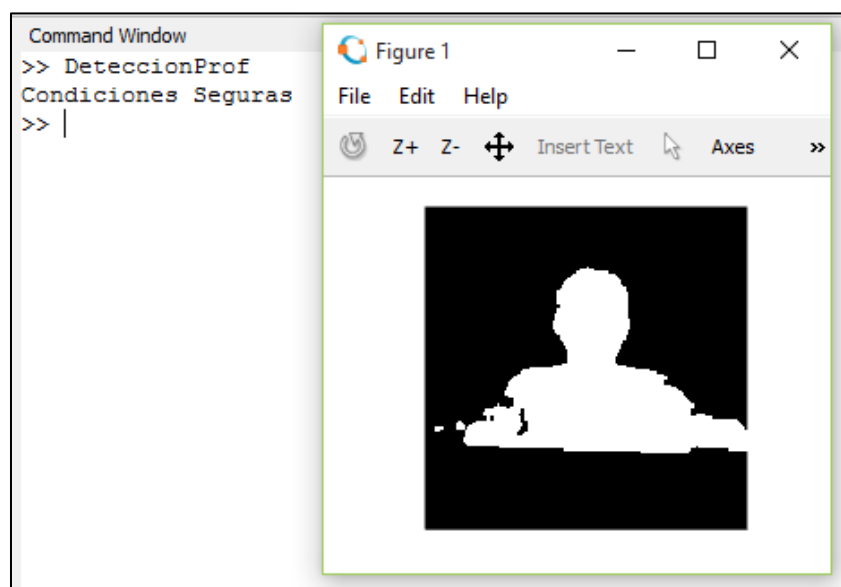


Figura 40: Resultado de correr el programa DetecciónProf (método 2) con una serie de imágenes con la persona en posición neutral y con los brazos hacia el frente (Creado por el autor en Octave)

Para la tercera prueba se utilizó una serie de imágenes en las cuales la persona se encuentra utilizando el celular con la mano derecha. Algunas de estas imágenes se muestran a continuación.



Figura 41: Imágenes pertenecientes a la serie con la persona hablando por celular con la mano derecha (Osório & Berri, 2015)

En estas imágenes se pueden observar movimientos leves e inclinaciones en la cabeza y la mano, pero en todas se mantiene que la posición del celular está cerca de la oreja (dentro de la zona de interés en los alrededores de la cabeza). Al realizar un promedio entre estas imágenes, se

obtiene una forma clara de la cabeza y el cuerpo, así como del brazo con el celular cerca de la oreja. Esto permite que se haga una substracción de fondo más clara y consistente. Los resultados para cada uno de los métodos se presentan a continuación.

Método de análisis de histogramas: (DetecciónHist)



Figura 42: Resultado de correr el programa DetecciónHist (método 1) con una serie de imágenes con la persona hablando por celular con la mano derecha (Creado por el autor en Octave)

Método de análisis de zona alrededor de la cabeza (DetecciónProf)

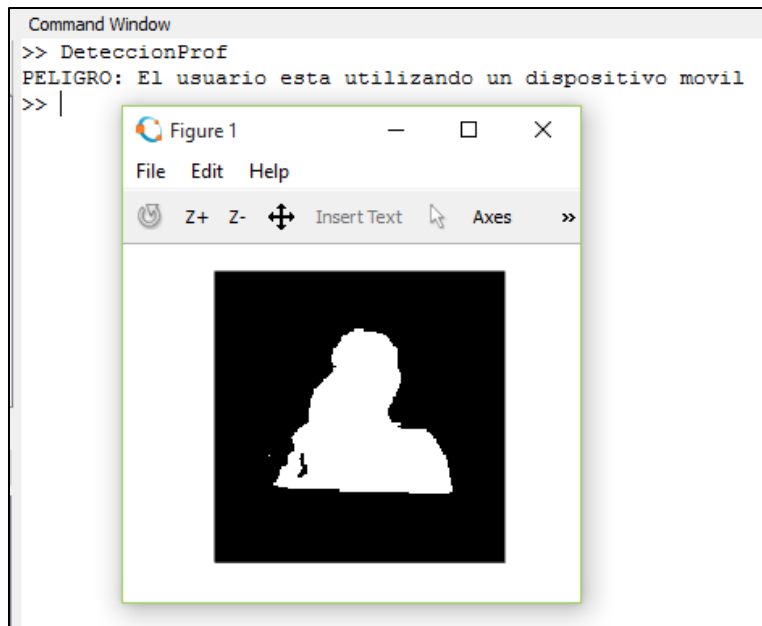
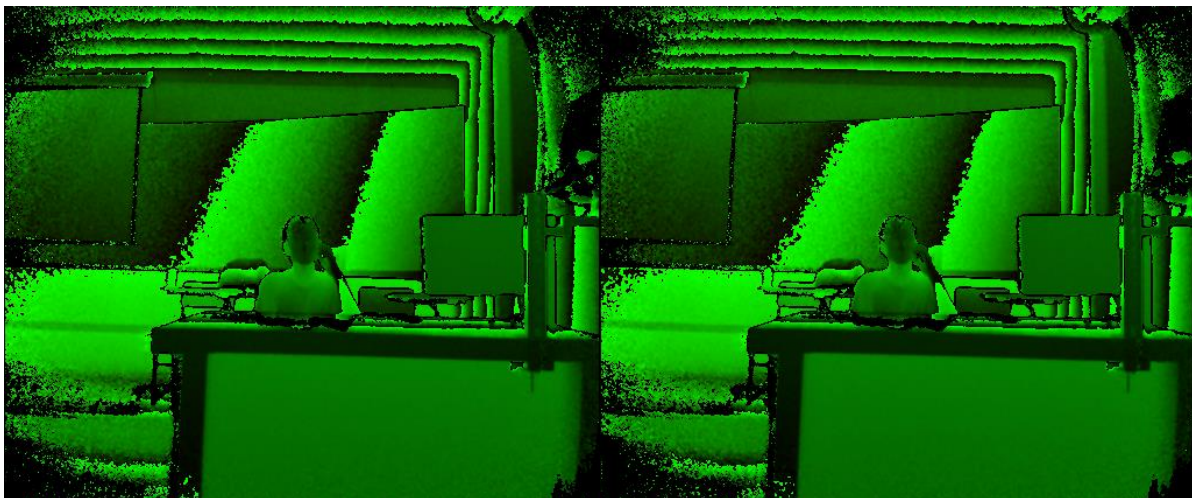


Figura 43: Resultado de correr el programa DetecciónProf (método 2) con una serie de imágenes con la persona hablando por celular con la mano derecha (Creado por el autor en Octave)

Finalmente, se realizó una cuarta prueba utilizando una serie de imágenes en las cuales la persona está hablando por celular con la mano izquierda. Algunas de las imágenes se muestran a continuación.



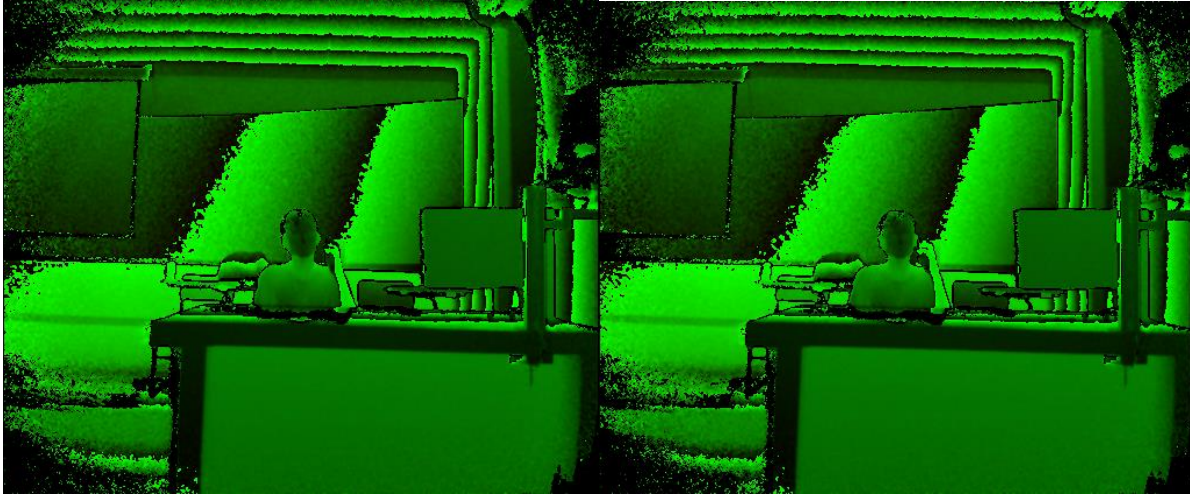


Figura 44: Imágenes pertenecientes a la serie con la persona hablando por celular con la mano izquierda (Osório & Berri, 2015)

Para estas imágenes se tiene una situación similar a la anterior, en donde se tienen movimientos leves e inclinaciones de la cabeza y la mano. De la misma forma, se resuelve al realizar el promedio de las imágenes. El resultado de utilizar cada uno de los métodos para esta serie de imágenes es el siguiente.

Método de análisis de histogramas: (DetecciónHist)

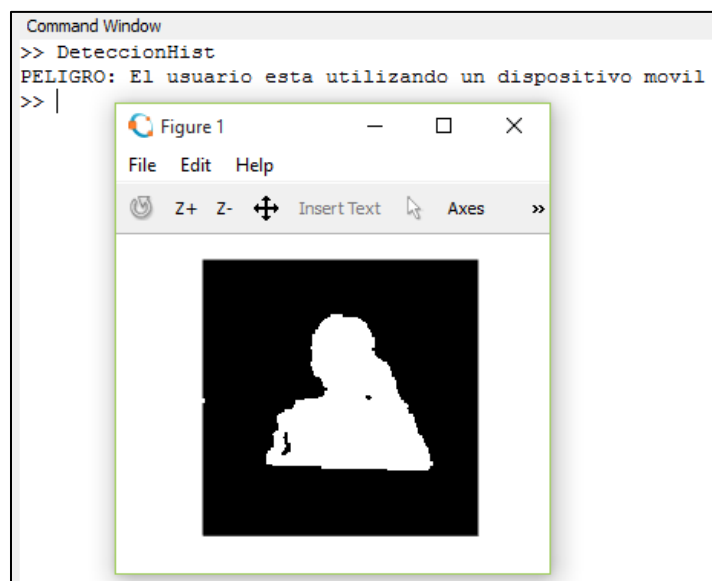


Figura 45: Resultado de correr el programa DetecciónHist (método 1) con una serie de imágenes con la persona hablando por celular con la mano izquierda (Creado por el autor en Octave)

Método de análisis de zona alrededor de la cabeza (DetecciónProf)

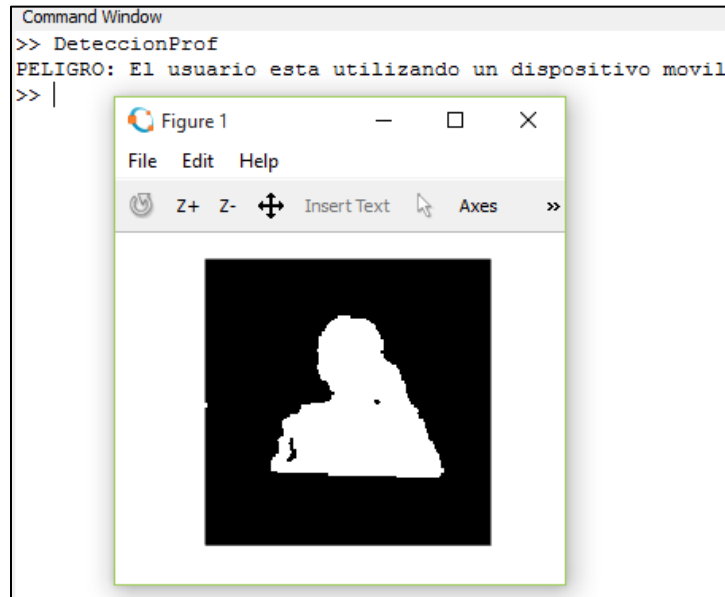


Figura 46: Resultado de correr el programa DetecciónProf (método 2) con una serie de imágenes con la persona hablando por celular con la mano izquierda (Creado por el autor en Octave)

Capítulo 6: Discusión de resultados y limitaciones

Los resultados obtenidos tanto para la simulación como para la parte de verificación de datos fueron los esperados ya que se consiguió realizar un reconocimiento efectivo del gesto de hablar por celular. Sin embargo, aunque se utilizaron datos reales, aún para la verificación de datos se tiene un ambiente idealizado con el que se querría trabajar. En este ambiente se tiene una iluminación controlada y artificial (se realizó en un ambiente interno). Esto significa que no se tiene influencia directa de luz solar, la cual afecta de manera importante la lectura de datos en el sensor Kinect. Esta es una limitación importante del proyecto, ya que no se llega a utilizar un ambiente que incluya todas las condiciones que se pueden presentar a la hora de implementar el

sistema. En situaciones reales, puede existir una variedad de inconvenientes que pueden afectar los resultados del programa.

Primeramente, los valores umbrales utilizados para el programa fueron determinados experimentalmente. Esto quiere decir que solo funcionan para la situación en específico en la cual se obtuvieron los datos. Si se quiere tener una aplicación más robusta, se debe realizar una serie de pruebas en diferentes ambientes y con diferentes personas, de modo que se pueda obtener valores umbrales que resulten en detecciones correctas para una variedad de circunstancias. Para esto debe hacerse pruebas en el ambiente real en el que se aplicaría el sistema, un vehículo. De cualquier forma, diferentes vehículos requerirían diferentes valores umbrales debido a que su geometría cambia, los elementos que se encuentran en el fondo cambian (asientos, el techo, accesorios del vehículo) y la posición en la que se ubica el sensor (distancia e inclinación) también cambia. Asimismo, las variaciones en iluminación y las vibraciones del sensor debido al movimiento del vehículo pueden afectar de manera importante las mediciones que este realiza y por lo tanto la detección del gesto. Para poder solucionar estos problemas se debe trabajar más a fondo la aplicación, realizando una serie de pruebas que permitan encontrar los valores que mejor se ajusten a la solución del problema y que toleren los pequeños cambios que puedan afectar la lectura de datos.

Otro problema que se presenta es que, aunque el programa detecta efectivamente el gesto de hablar por celular, existe otra variedad de gestos diferentes que también serán detectados por el programa. Por ejemplo, el hecho de acomodarse el cabello o rascarse la cabeza por más de 5 segundos será detectado como peligroso ya que se interpreta como si fuera el gesto objetivo de este proyecto. Esto se debe a que el único criterio utilizado para la detección fue la cercanía del objeto con la cabeza. Para evitar detectar otro tipo de gestos se puede ampliar este proyecto, agregando una función que detecte cuando la mano está cerrada, lo cual sucede cuando la persona sostiene el celular. Además, podría utilizarse información del dispositivo como tal, como su color por ejemplo, para determinar que existe un objeto además de la mano que se está acercando a la oreja. Para conseguir hacer esto se debe utilizar otro tipo de simulador que permita realizar lecturas más precisas de la mano y se debe desarrollar un algoritmo más avanzado para conseguir detectar estos detalles. No obstante, se considera que cualquier gesto que cause que el conductor separe las manos del volante por un tiempo determinado puede

considerarse peligroso. Por ello, detectar otro tipo de gestos que involucren las manos cerca de la cabeza no representa realmente un problema.

Al observar los resultados que se obtienen con cada uno de los métodos se puede ver a simple vista que ambos funcionan de la misma forma y producen los mismos resultados, por lo que no se puede determinar únicamente con esta información cuál de ellos es más eficiente o funciona mejor. Para compararlos, se debe considerar los problemas que puedan surgir por causa de variaciones en el ambiente y en la persona, movimientos o presencia de otros objetos. Por ejemplo, si una persona tiene el cabello largo que sobrepasa el cuello, la obtención del punto de inicio del cuello se verá afectada para el método de análisis de histogramas, ya que el cabello hace parecer que el cuello es más ancho de lo que es en realidad. Esto causa que no se pueda determinar correctamente la posición de inicio del cuello, ya que la razón que se utiliza para esto nunca tendrá el valor que se requiere. Por otro lado, esto no será un problema para el método de análisis de profundidad, ya que solo depende de los datos de profundidad de la cara y el cuello. El método de análisis de histogramas solo funcionaría correctamente con el cabello recogido, o utilizando un algoritmo de substracción de cabello que, aunque ya existe, es bastante complejo de implementar.

Las variaciones físicas entre personas también puede afectar el cálculo de los puntos límites de la cabeza para este método, ya que se utilizan valores estándar de fisionomía promedio para determinar la razón que se utiliza. Según estas dificultades se podría argumentar que el método de análisis de profundidades es más efectivo. Sin embargo, si se utilizan accesorios en el cuello como bufandas, se afectaría el resultado con este método, ya que la profundidad del cuello puede llegar a ser menor que la profundidad del punto máximo de la cabeza, por lo que la razón entre estos valores nunca será mayor que 1. Por lo tanto, se puede ver que existen limitaciones para ambos métodos en cuestión de variaciones físicas de las personas y uso de accesorios.

Los movimientos de la persona durante el tiempo de manejo o la inclinación de la cabeza pueden afectar las mediciones para ambos métodos utilizados. Sin embargo, debido a que el método de análisis de profundidades utiliza puntos fijos tanto en X como en Y, cualquier movimiento de la cabeza va a causar una lectura inadecuada. En el caso del método de análisis de histogramas, un movimiento en dirección horizontal no afecta la lectura y detección, ya que el número de píxeles de la cabeza se mantiene igual (ésta no crece ni se encoge), pero una

inclinación podría causar una detección errónea ya que el ancho de cada fila va a cambiar. Para este aspecto se puede decir que el método de análisis de histogramas es mucho más efectivo y presenta menos problemas, ya que inevitablemente la persona va a mover la cabeza durante el tiempo en el que está conduciendo. Para solucionar esto para el método de análisis de profundidad, se podría utilizar un algoritmo de detección continua de la cara, como los que se utilizan para ambientes de juego con Kinect, utilizando la información de geometría promedio de una cara.

Finalmente, existe un problema que puede surgir para cualquiera de los dos métodos: saber cuándo se tiene cuál situación. Esto se refiere a poder saber cuándo se debe tomar la imagen que se utiliza para medir los extremos de la cabeza y cuándo se debe tomar la imagen que se utiliza para detección. La imagen utilizada para el cálculo de los puntos debe ser una en la que la persona se encuentre en posición neutral. Para esto se argumentaba que se puede ligar la toma de la imagen al encendido del carro. Sin embargo, esto no asegura que la persona no esté utilizando el celular o tenga las manos cerca de la cabeza en el momento en que enciende el carro. Para solucionarlo, se puede utilizar fusión de sensores, al posicionar sensores de tacto sobre el volante. Al tener ambas manos sobre los sensores se sabe que la persona está en posición neutral y se puede tomar la imagen que se utiliza para los cálculos. Luego, cuando alguna de las manos se separa del sensor, el programa empieza con la etapa de detección, tomando imágenes continuas para analizar. Ya que el programa corre mucho más rápido de lo que se mueve una persona, varias imágenes consecutivas tendrán posiciones similares, por lo que se puede realizar el promedio de varias imágenes sin problemas.

Para cualquiera de las situaciones descritas anteriormente, se requiere un extenso trabajo posterior. Estos temas serían buenas opciones para trabajos futuros basados en esta aplicación como base.

Capítulo 7: Diseño de la estructura de soporte

Para poder posicionar el sensor Kinect correctamente dentro del vehículo y sin que éste se desplace con el movimiento del carro o se vea afectado por las vibraciones, se diseñó una estructura para ser posicionada en el interior de un vehículo. Las dimensiones y posicionamiento de la estructura varía de manera importante entre diferentes vehículos, ya que cada uno tiene un

espacio y distribución de las partes único para su modelo. Debido a que este proyecto fue asignado por el laboratorio para ser implementado en su proyecto más reciente, el cual consiste en un camión autónomo desarrollado para la empresa SCANIA, el diseño del soporte se ajustó a este vehículo.



Figura 47: Camión autónomo desarrollado por el LRM (Laboratório de Robótica Móvel, 2015)

Al ser un camión, este vehículo cuenta con mucho espacio detrás del espejo retrovisor, sin afectar la visión del conductor. Esto permite que se realice una estructura con mayor altura, para posicionar el sensor de forma que tenga un adecuado campo de visión centrado en el conductor. La estructura que se desarrolló es la siguiente.

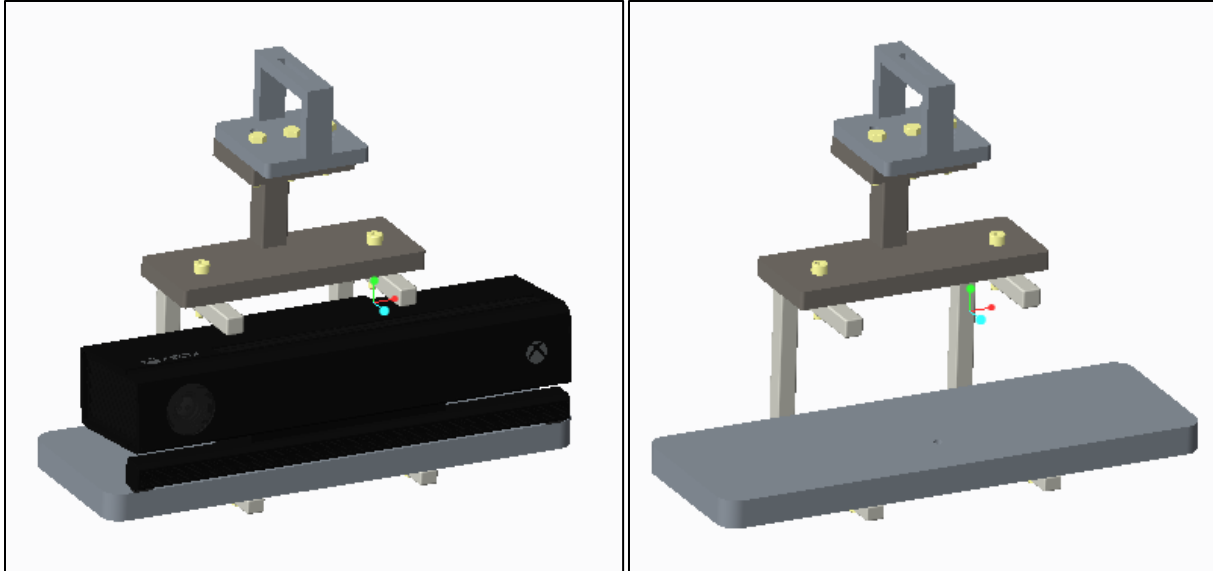


Figura 48: Estructura de soporte para sostener el sensor (con el Kinect en su posición y sin el Kinect) (Creado por el autor en CREO Parametric) Kinect tomado de (Sintonen, 2014)

Esta estructura fue diseñada para colocarse detrás del espejo retrovisor, ajustándose en un soporte que fue posicionado en el techo del camión. Todas las piezas están diseñadas para ser construidas en aluminio, de modo que sean resistentes. La unión entre las piezas se realiza con uniones no permanentes (tornillos estándar), incluyendo el ajuste del Kinect sobre la placa inferior. La estructura está compuesta de 4 piezas diferentes (Ver Apéndice A.3 para los planos de construcción). La pieza superior (que se ajusta a la estructura en el techo) está hecha de tal forma que la posición de los tornillos determine el giro de la estructura. Dependiendo de la posición de éstos, se puede girar la estructura en 20° o 40° , dependiendo de las necesidades específicas del vehículo. Para el caso del camión, la estructura se debe posicionar a 20° de giro para poder centrar el campo de visión en el conductor. La estructura también fue diseñada de tal forma que los cables que salen detrás del Kinect puedan pasar fácilmente.

Capítulo 8: Conclusiones y Recomendaciones

8.1. Conclusiones

- Se realizó exitosamente el diseño de una aplicación de reconocimiento gestual según los requerimientos definidos inicialmente y las especificaciones que surgieron del análisis de las posibles soluciones.
- Se realizó una simulación funcional utilizando dos métodos distintos con el programa V-REP.
- Se obtuvieron resultados deseables de la parte de la simulación, realizando detección de gesto correctamente con ambos métodos.
- Se utilizó exitosamente los datos de un sensor real para respaldar los resultados obtenidos en la simulación con un programa funcional utilizando dos métodos distintos con el programa Octave.
- Se obtuvieron resultados deseables de la parte de verificación de datos, realizando detección de gesto correctamente con ambos métodos.
- Se compararon los dos métodos utilizados y se determinó que aunque ambos producen los mismos resultados, tienen distintas limitaciones por causa de movimientos y variaciones del ambiente y de la persona.
- Se diseñó exitosamente una estructura de soporte para posicionar el sensor Kinect en un vehículo, para poder implementar el sistema en una aplicación real.

8.2. Recomendaciones

- Realizar pruebas de la aplicación utilizando una variedad de ambientes y diferentes personas para verificar su funcionamiento.
- Realizar lecturas del sensor en el interior del vehículo y ante condiciones reales de iluminación para verificar un desempeño adecuado.
- Complementar el programa desarrollado con un algoritmo que detecta cuando la mano está abierta o cerrada, para determinar más adecuadamente si el gesto realmente es el que se quiere reconocer.
- Complementar el programa de análisis de histogramas con un algoritmo que detecte cabello y realice una substracción para evitar que éste interfiera con la detección.

- Complementar el programa de análisis de profundidad con un algoritmo para detectar el rostro continuamente y así evitar que los movimientos e inclinaciones de la cabeza afecten la detección.
- Implementar fusión de sensores que hagan el sistema redundante para asegurar que se están realizando lecturas adecuadas de los gestos en todo momento.
- Realizar la construcción de la estructura de soporte para posicionar el sensor firmemente en el interior del vehículo y poder realizar pruebas o realizar la implementación más fácilmente.

Referencias

- [1] Bouwmans, T., Porikli, F., Höferlin, B., & Vacavant, A. (2014). *Background Modeling and Foreground Detection for Video Surveillance*. Chapman and Hall/CRC.
- [2] Braffort, A., Gherbi, R., Gibet, S., Richardson, J., & Teil, D. (1999). Gesture Based Communication in Human-Computer Interaction. *International Gesture Workshop*. France: Springer. doi:10.1007/978-981-4585-69-9_2
- [3] Buss, S. R. (2004). *Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Square methods*. Recuperado el 22 de agosto de 2015, de University of California, San Diego: <http://web.cse.ohio-state.edu/~parent/classes/694A/Lectures/Material/IKsurvey.pdf>
- [4] Cheung, S.-C. S., & Kamath, C. (2007). *Robust techniques for background subtraction in urban traffic video*. Technical Paper, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, California. Recuperado el 27 de setiembre de 2015, de <https://computation.llnl.gov/casc/sapphire/pubs/UCRL-CONF-200706.pdf>
- [5] Dramanan. (2013). Background Subtraction. Recuperado el 27 de setiembre de 2015, de http://www.ics.uci.edu/~dramanan/teaching/cs117_spring13/lec/bg.pdf
- [6] Gallahan, S. L., Golzar, G. F., Jain, A. P., Samay, A. E., Trerotola, T. J., Weisskopf, J. G., & Lau, N. (2013). Detecting and mitigating driver distraction with motion capture technology: Distracted driving warning system. *Systems and Information Engineering Design Symposium (SIEDS)* (pp. 76-81). IEEE. doi:10.1109/SIEDS.2013.6549497
- [7] Guizzo, E. (2011). How Google's Self-Driving Car Works. *IEEE Spectrum*. Recuperado el 18 de setiembre de 2015, de <http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works>
- [8] Killip, C. (n.d.). *Healthy Living*. Recuperado el 04 de octubre de 2015, de Neck Size in Relation to Waist Size: <http://healthyliving.azcentral.com/neck-size-relation-waist-size-14393.html>
- [9] Kulpa, R., & Multon, F. (2005). Fast inverse kinematics and kinetics solver for human-like figures. *5th IEEE-RAS International Conference on Humanoid Robots*, (pp. 38-43).

- Recuperado el 20 de setiembre de 2015, de
<http://www.irisa.fr/mimetic/GENS/fmulton/pdfs/Humanoids2005.pdf>
- [10] Laboratório de Robótica Móvel. (2015). *LRM ICMC/USP*. Recuperado el 13 de setiembre de 2015, de <http://lrm.icmc.usp.br/web/index.php?n=Port.Pesquisa>
- [11] Manzano, F. (2014, enero 17). *USP Universidade de Sao Paulo*. Recuperado el 22 de agosto de 2015, de Com auxílio das unidades, site reúne memória dos 80 anos de USP:
<http://www5.usp.br/39012/com-auxilio-das-unidades-site-reune-memoria-dos-80-anos-de-usp>
- [12] Microsoft. (2015). *Kinect for Windows SDK 1.8*. Recuperado el 12 de octubre de 2015, de Skeletal Tracking: <https://msdn.microsoft.com/en-us/library/hh973074.aspx>
- [13] Molchanov, P., Gupta, S., Kim, K., & Pulli, K. (2015). *Multi-sensor System for Driver's Hand-Gesture Recognition*. Technical Paper, NVIDIA Research, California. Recuperado el 25 de setiembre de 2015, de
<http://people.csail.mit.edu/kapu/papers/DriverHandGestureFG2015.pdf>
- [14] Murphy-Chutorian, E., Doshi, A., & Trivedi, M. M. (2007). Head Pose Estimation for Driver Assistance Systems: A Robust Algorithm and Experimental Evaluation. *Intelligent Transportation Systems Conference, 2007* (pp. 709-714). IEEE. Recuperado el 22 de setiembre de 2015, de
<http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4357803&url=http%3A%2F%2Fieeexplore.ieee.org%2Fstamp%2Fstamp.jsp%3Ftp%3D%26arnumber%3D4357803>
- [15] Ohn-Bar, E., & Trivedi, M. M. (2014). Hand Gesture Recognition in Real Time for Automotive Interfaces: A Multimodal Vision-Based Approach and Evaluations. *Intelligent Transportation Systems, IEEE Transactions*. 15, pp. 2368-2377. IEEE. Recuperado el 23 de setiembre de 2015, de
<http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6883176&url=http%3A%2F%2Fieeexplore.ieee.org%2Fstamp%2Fstamp.jsp%3Ftp%3D%26arnumber%3D6883176>

- [16] Open CV. (n.d.). *Open Source Computer Vision*. Recuperado el 26 de setiembre de 2015, de How to Use Background Subtraction Methods:
http://docs.opencv.org/master/d1/dc5/tutorial_background_subtraction.html#gsc.tab=0
- [17] Organización Mundial de la Salud. (2009). *Informe sobre la situación mundial de la seguridad vial 2009*. Recuperado el 29 de agosto de 2015, de
http://www.who.int/violence_injury_prevention/road_safety_status/report/web_version_es.pdf?ua=1
- [18] Osório, F. (2015). USP-ICMC-LRM. *Disciplina de Programação de Robôs Móveis*. Sao Carlos, Brasil. Recuperado el 30 de agosto de 2015
- [19] Osório, F., & Berri, R. (2015, octubre). *Kinect-Celular*. Recuperado el 5 de noviembre de 2015, de OneDrive: <http://1drv.ms/1NtqUbE>
- [20] Pavlovic, V., Sharma, R., & Huang, T. (1997). Visual Interpretation of Hand Gestures for Human-Computer Interactions: A Review. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 19, págs. 677 -695. IEEE. Recuperado el 15 de octubre de 2015, de <http://www.cs.rutgers.edu/~vladimir/pub/pavlovic97pami.pdf>
- [21] Premaratne, P. (2014). Historical Development of Hand Gesture Recognition. En *Human Computer Interaction Using Hand Gestures* (págs. 5-29). Singapore: Springer. doi:10.1007/978-981-4585-69-9_2
- [22] Sintonen, T. (2014, setiembre 9). *Xbox ONE Kinect Sensor*. Recuperado el 25 de octubre de 2015, de GrabCad: <https://grabcad.com/library/xbox-one-kinect-sensor-1>
- [23] Solomon, C., & Wang, Z. (2015, diciembre). Driver Attention and Behavior Detection with Kinect. *Journal of Image and Graphics*, 03(03), 84-89. doi:10.18178/joig.3.2.84-89
- [24] Srilatha, P., & Saranya, T. (2014). Advancements in Gesture Recognition Technology. *IOSR Journal of VLSI and Signal Processing*, 4(4), 1-7. Recuperado el 16 de octubre de 2015, de www.iosrjournals.org
- [25] Tamersoy, B.; Aggarwal, J.K.(2009) Robust Vehicle Detection for Tracking in Highway Surveillance Videos Using Unsupervised Learning, *Advanced Video and Signal Based Surveillance*, 529-534, doi: 10.1109/AVSS.2009.57

- [26] Thippur, A., Ek, C. H., & Kjellstrom, H. (2013). *Inferring Hand Pose: A Comparative Study of Visual Shape Features*. Stockholm: IEEE. Recuperado el 15 de octubre de 2015, de http://www.csc.kth.se/~hedvig/publications/fg_13.pdf
- [27] Vacavant, A., Chateu, T., Wilhelm, A., & Lequière, L. (2013). A benchmark dataset for outdoor foreground/background extraction. *Computer Vision-ACCV 2012 Workshops*, (págs. 291-300). Springer. doi:10.1007/978-3-642-37410-4_25
- [28] Yan, J., Zhang, X., Lei, Z., Yi, D., & Li, S. (2013). *Structural Models for Face Detection*. China: IEEE. Recuperado el 15 de octubre de 2015, de <https://www.computer.org/csdl/proceedings/fg/2013/5545/00/06553703.pdf>
- [29] Yim, Y. (12 de 2003). Three-feature based automatic lane detection algorithm (TFALDA) for autonomous driving. *Intelligent Transportation Systems, IEEE Transactions*, 4(4), 219-225. Recuperado el 16 de setiembre de 2015, de <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1260588&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Ficp.jsp%3Farnumber%3D1260588>
- [30] Zeng, W. (2012). Microsoft Kinect Sensor and its Effect. *Multimedia at Work*, 4-10. Recuperado el 29 de setiembre de 2015, de <http://research.microsoft.com/en-us/um/people/zhang/Papers/Microsoft%20Kinect%20Sensor%20and%20Its%20Effect%20-%20IEEE%20MM%202012.pdf>
- [31] Zhang, C., Yang, X., & Tian, Y. (2013). *Histogram of 3D Facets: A Characteristic Descriptor for Hand Gesture Recognition*. New York: IEEE. Recuperado el 16 de octubre de 2015, de <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6553754>
- [32] Zhang, X., Yin, L., Cohn, J., Canavan, S., Reale, M., Horowitz, A., & Liu, P. (2013). *A High-Resolution Spontaneous 3D Dynamic Facial Expression Database*. New York: IEEE. Recuperado el 12 de octubre de 2015, de <http://www.pitt.edu/~jeffcjohn/biblio/3DSponDB.pdf>

Apéndices

Apéndice A.1: Histogramas de píxeles para diferentes distancias y posiciones del sensor Kinect

Para poder determinar la relación entre el ancho del cuerpo y del cuello, se realizaron histogramas de píxeles pertenecientes a la figura humana para varias distancias entre el sensor y la persona y posiciones o inclinaciones del sensor. Para cada histograma, se recorrió la fila de píxeles con un contador que se incrementa cada vez que se tiene un píxel que no pertenece al fondo. Al finalizar el recorrido de la fila, el valor del contador se almacenó en un vector. Este vector tiene igual número de posiciones que el número de filas en la resolución Y del sensor Kinect. Para el caso del proyecto, se tiene una resolución en Y de 48 y por lo tanto el vector que contiene los contadores tiene 48 posiciones. Se imprimieron los valores de contador de cada fila y se introdujo en Excel para obtener un gráfico con los valores del histograma. Esto permite observar más fácilmente la relación que existe entre las filas, y determinar por medio de diferentes pruebas, la razón entre la parte más ancha y el inicio del cuello que permite una detección del punto de interés en todos los casos.

A.1.1. Posición frontal del Kinect con respecto a la persona, a una distancia de 0.5 m

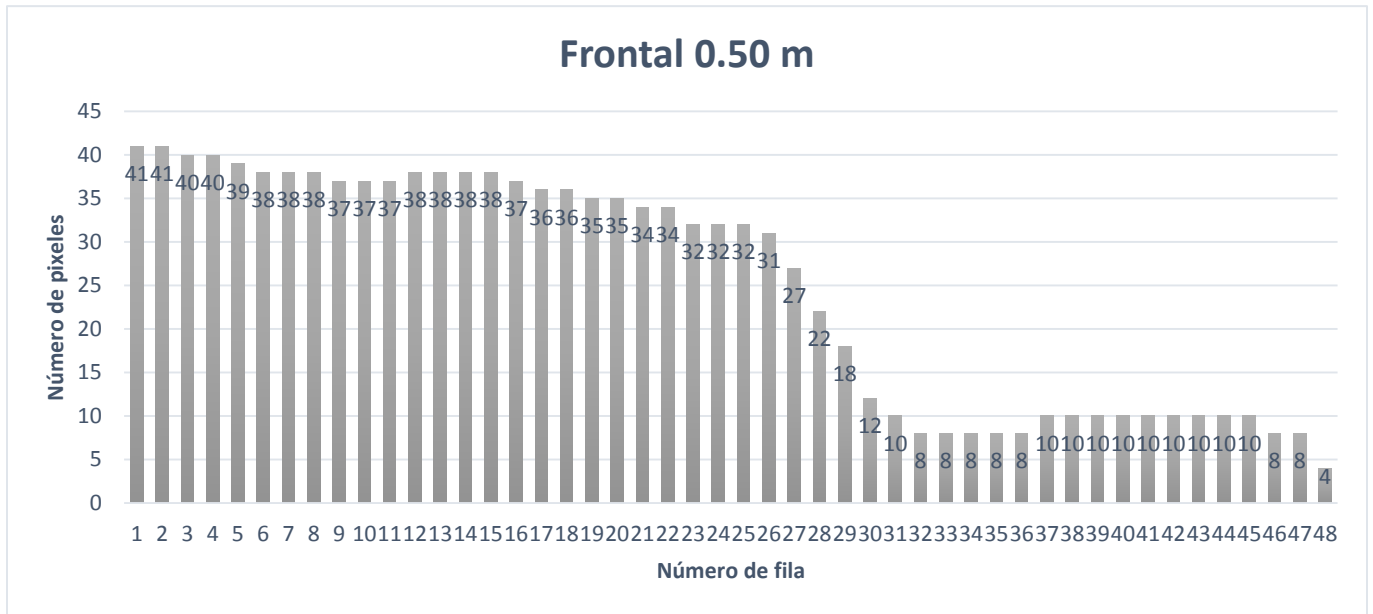


Figura A.1.1: Histograma de pixeles para el sensor en posición frontal, a una distancia de 0.5m (Creado por el autor en Excel)

En este gráfico se puede observar que la cantidad máxima de pixeles se da en la fila 1, con un total de 41. El inicio del cuello se da en la fila 32, con una cantidad de 8 pixeles en este punto. Al calcular la relación entre el máximo y el inicio del cuello se tiene un valor de $41/8 = 5.125$.

A.1.2. Posición frontal del Kinect con respecto a la persona, a una distancia de 0.75 m

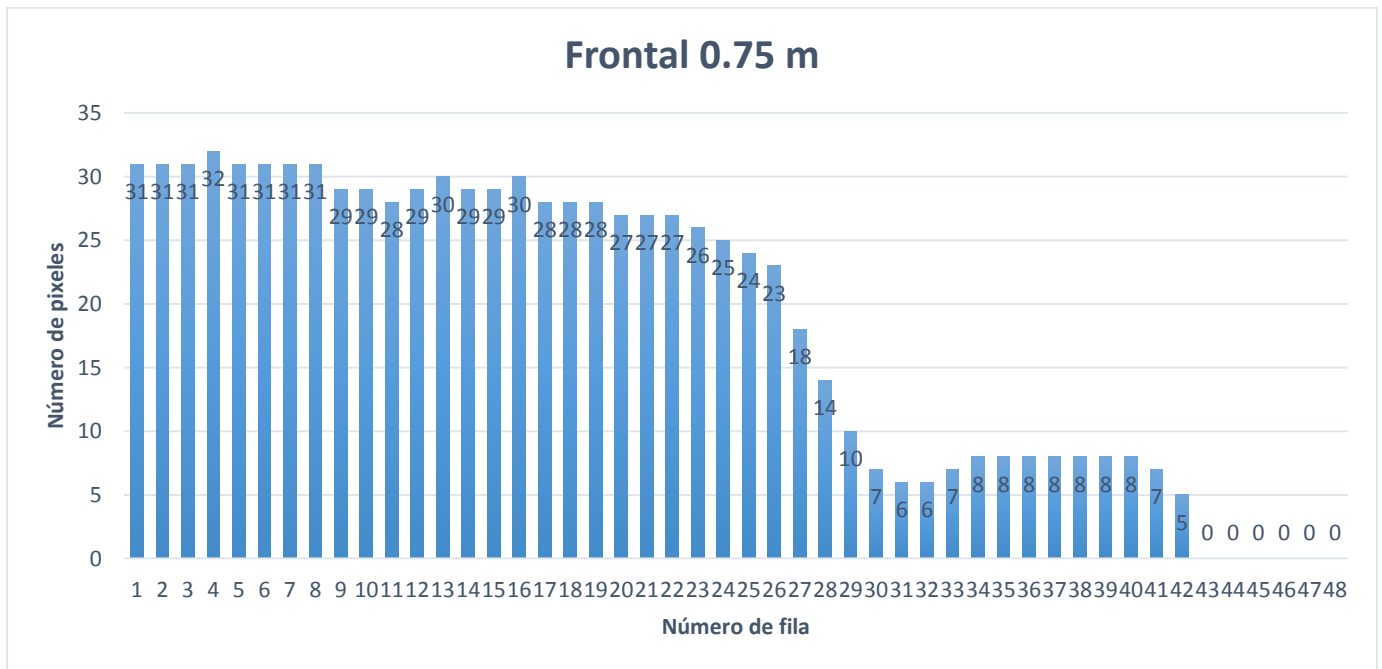


Figura A.1.2: Histograma de píxeles para el sensor en posición frontal, a una distancia de 0.75m (Creado por el autor en Excel)

Para este gráfico, la cantidad máxima de píxeles es de 32 y se da en la posición 4. El inicio del cuello se da en la posición 31 con una cantidad de 6 píxeles. La relación entre el número máximo y el inicio del cuello es de $32/6 = 5.3333$.

A.1.3. Posición frontal del Kinect con respecto a la persona, a una distancia de 0.9 m

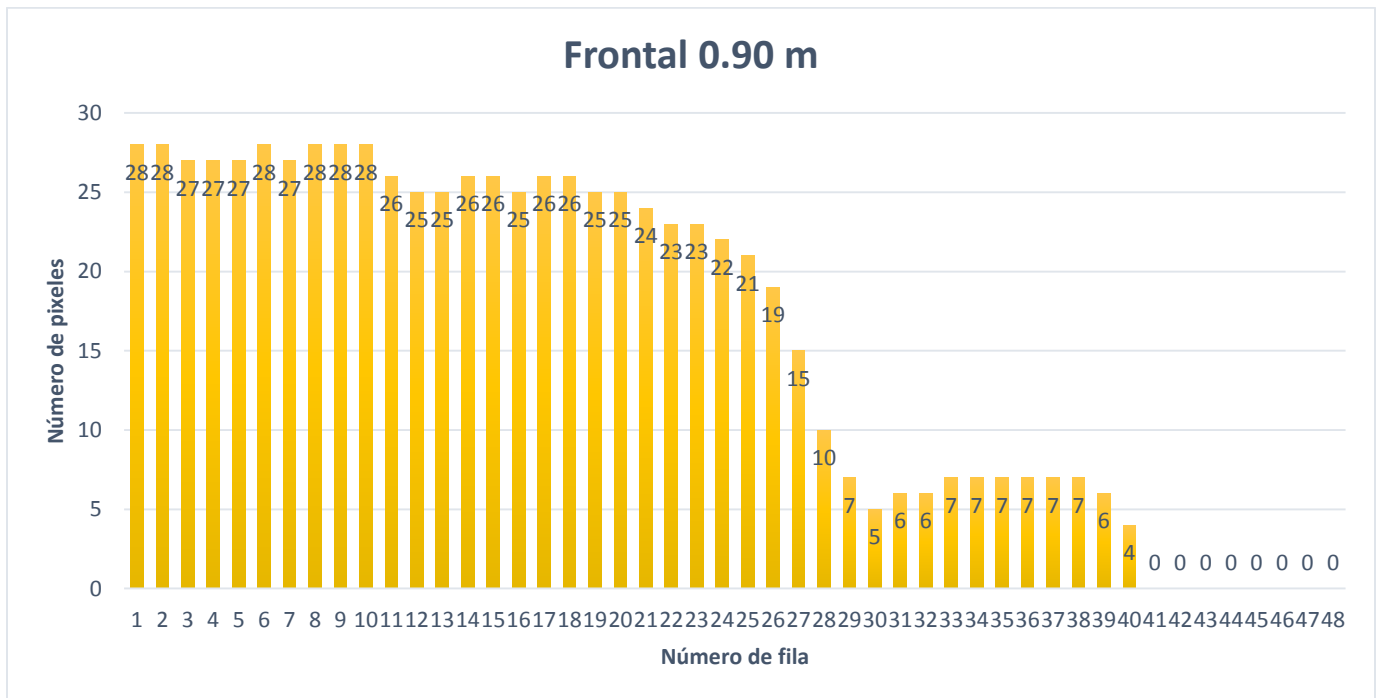


Figura A.1.3: Histograma de pixeles para el sensor en posición frontal, a una distancia de 0.9m (Creado por el autor en Excel)

Para este gráfico, la cantidad máxima de pixeles es de 28 y se da en la posición 1. El inicio del cuello se da en la posición 30 con una cantidad de 5 pixeles. La relación entre el número máximo y el inicio del cuello es de $28/5 = 5.6$.

A.1.4. Posición del Kinect con una inclinación de 20° respecto a la persona, a una distancia de 0.75 m en Y y 0.2 m en X

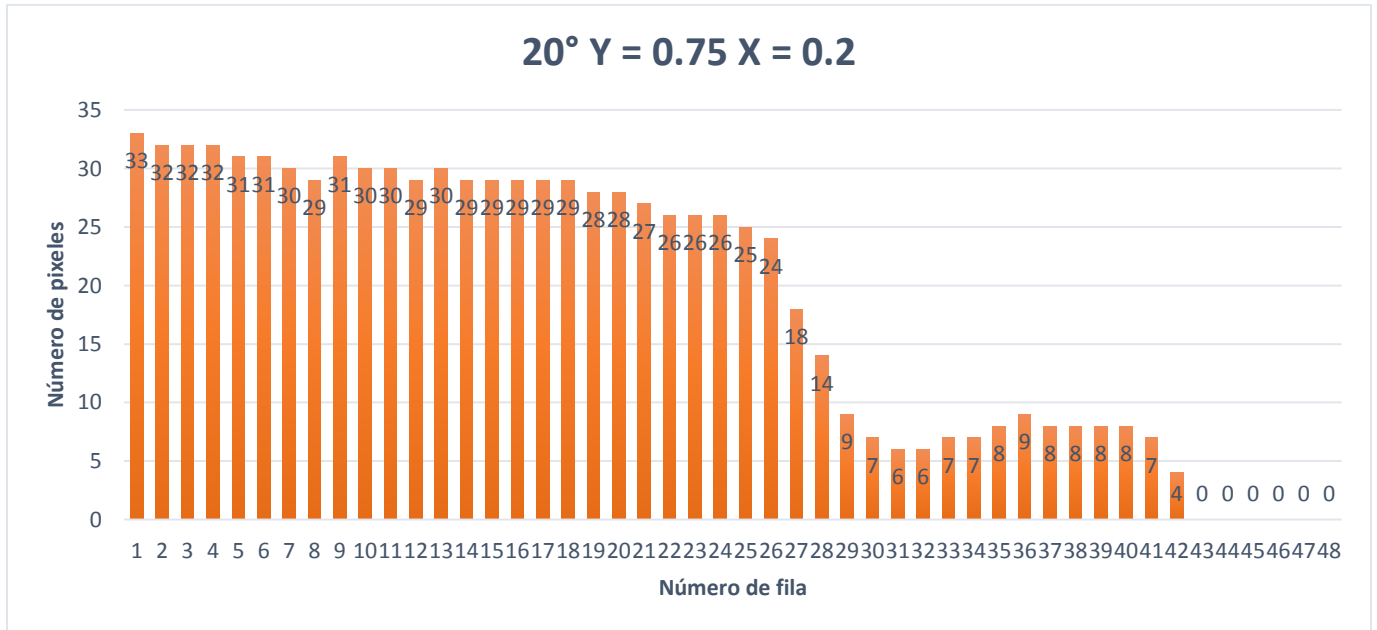


Figura A.1.4: Histograma de píxeles para el sensor en posición de 20° de inclinación, a una distancia de 0.75m en Y y 0.2m en X (Creado por el autor en Excel)

Para este gráfico, la cantidad máxima de píxeles es de 33 y se da en la posición 1. El inicio del cuello se da en la posición 31 con una cantidad de 6 píxeles. La relación entre el número máximo y el inicio del cuello es de $33/6 = 5.5$.

A.1.5. Posición del Kinect con una inclinación de 30° respecto a la persona, a una distancia de 0.75 m en Y y 0.3 m en X

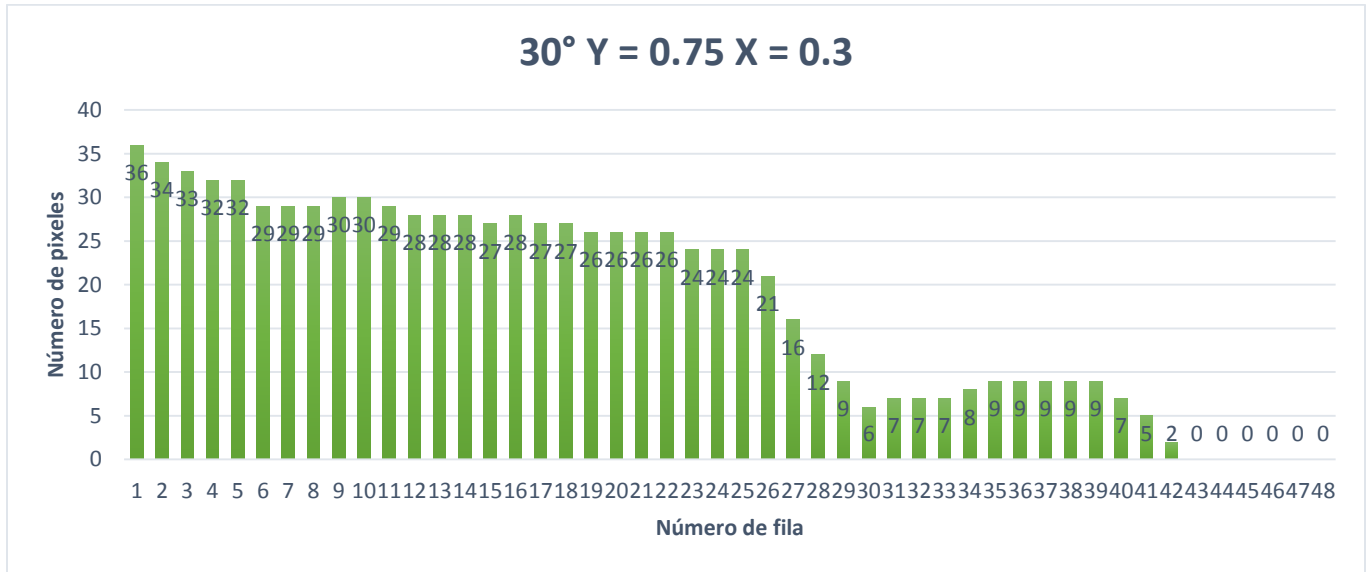


Figura A.1.5: Histograma de píxeles para el sensor en posición de 30° de inclinación, a una distancia de 0.75m en Y y 0.3 m en X (Creado por el autor en Excel)

Para este gráfico, la cantidad máxima de píxeles es de 36 y se da en la posición 1. El inicio del cuello se da en la posición 30 con una cantidad de 6 píxeles. La relación entre el número máximo y el inicio del cuello es de $36/6 = 6$.

Apéndice A.2.: Profundidades medidas desde el punto máximo de la cabeza hasta el inicio del cuello

El segundo método utilizado para encontrar el punto de inicio del cuello consiste en comparar los valores de profundidad medibles por el sensor para encontrar el punto en donde ésta se incrementa, indicando que se tiene el cuello. Se sabe que el punto máximo medido (extremo superior de la cabeza) tiene una profundidad superior a la profundidad que se mide en cualquier punto de la cara. Conociendo este punto, se puede almacenar su valor de profundidad para utilizarlo como referencia. El punto máximo de la cabeza, a su vez, tendrá una profundidad menor que la del cuello, por lo que puede utilizarse este dato para encontrar un punto en donde la razón entre la profundidad medida y la profundidad de la cabeza sea mayor que uno. Para comprobar esta teoría, se midieron los valores de profundidad iniciando desde el punto máximo de la cabeza y descendiendo hasta donde se presume que inicia el cuello para este ejemplo en particular. Los resultados son los siguientes.

Tabla A.2.1: Valores de profundidad para las filas 30 a 48 con el sensor posicionado a una distancia de 0.5m

Profundidades para distancia de 0.5 m		
Número de fila	Valor de profundidad	Razón Pactual/Pcabeza
48	0.2259	1
47	0.2221	0.983178398
46	0.2191	0.969898185
45	0.2179	0.9645861
44	0.2172	0.961487384
43	0.217	0.960602036
42	0.2181	0.965471448
41	0.219	0.969455511
40	0.2163	0.95750332
39	0.2159	0.955732625
38	0.2161	0.956617973
37	0.2162	0.957060646
36	0.2161	0.956617973
35	0.2161	0.956617973
34	0.2164	0.957945994
33	0.2179	0.9645861

32	0.2315	1.02478973
31	0.2321	1.027445772
30	0.2321	1.027445772

Se puede observar para esta tabla que la parte superior del cuello inicia en la fila 32 para un sensor posicionado a 0.5m de la persona. Este es el punto en donde se da una razón mayor que 1. Este será reconocido como el punto del cuello para ser utilizado como punto inferior de la cabeza. Para una distancia de 0.75 m, el punto de inicio del cuello se encuentra en la fila 31, como se muestra en la siguiente tabla.

Tabla A.2.2: Valores de profundidad para las filas 31 a 42 con el sensor posicionado a una distancia de 0.75m

Profundidades para distancia de 0.75 m		
Número de fila	Valor de profundidad	Razón Pactual/Pcabeza
42	0.3064	1
41	0.297	0.969321149
40	0.2916	0.951697128
39	0.291	0.949738903
38	0.291	0.949738903
37	0.2916	0.951697128
36	0.2932	0.95691906
35	0.2941	0.959856397
34	0.2936	0.958224543
33	0.296	0.966057441
32	0.2986	0.974543081
31	0.3099	1.011422977
30	0.3097	1.010770235

Apéndice A.3: Planos de construcción para la estructura de soporte del sensor Kinect

Se diseñó una estructura que sostiene el sensor dentro del camión SCANIA que el laboratorio modificó para hacerlo autónomo. La estructura consiste de 4 partes diferentes que se ensamblan utilizando uniones no permanentes. A continuación se presentan los planos de construcción para dichas piezas.

A.3.1. Placa #1

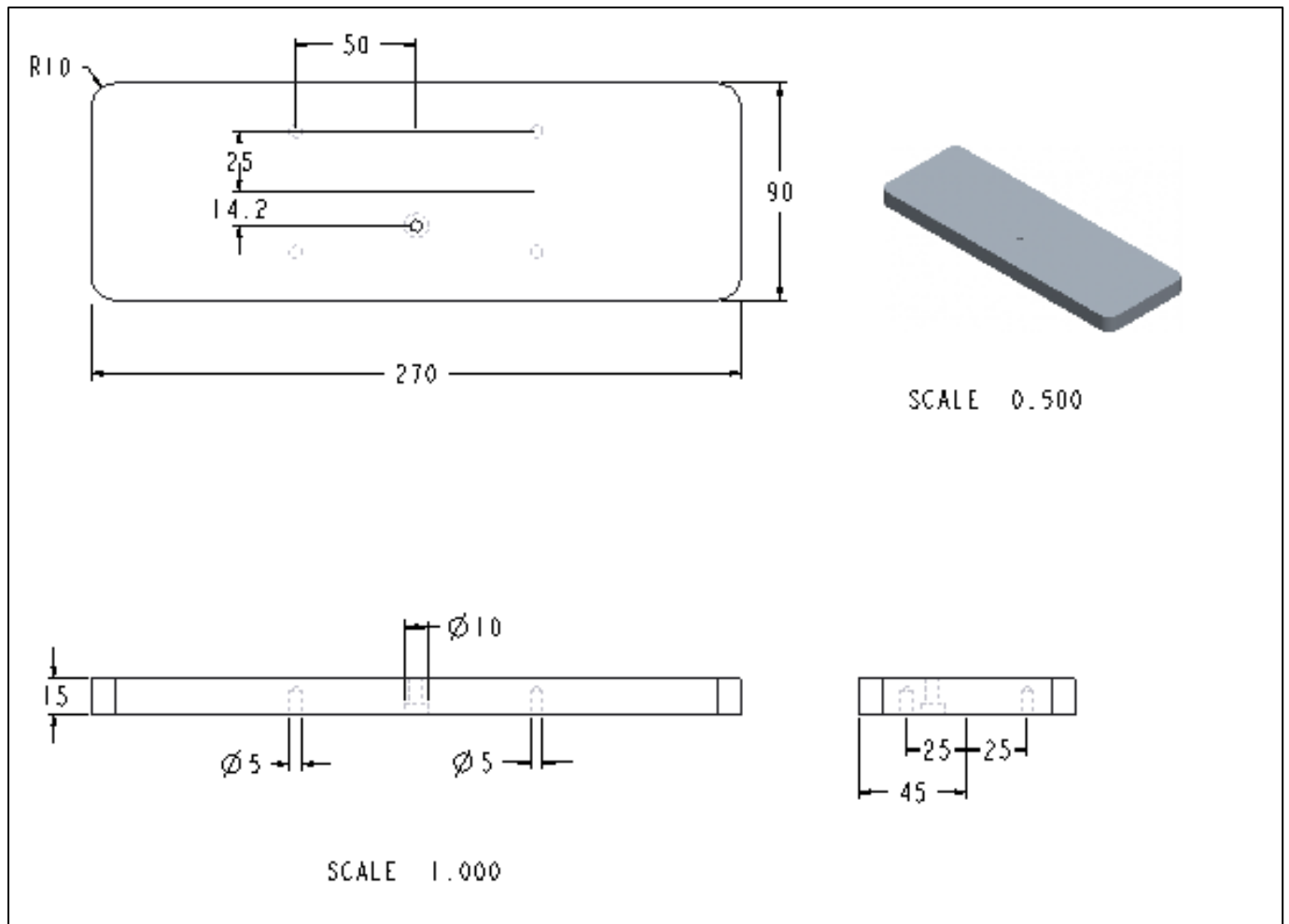


Figura A.3.1: Placa #1 para posicionamiento del sensor Kinect (Creado por el autor en CREO Parametric)

A.3.2. Pieza #2

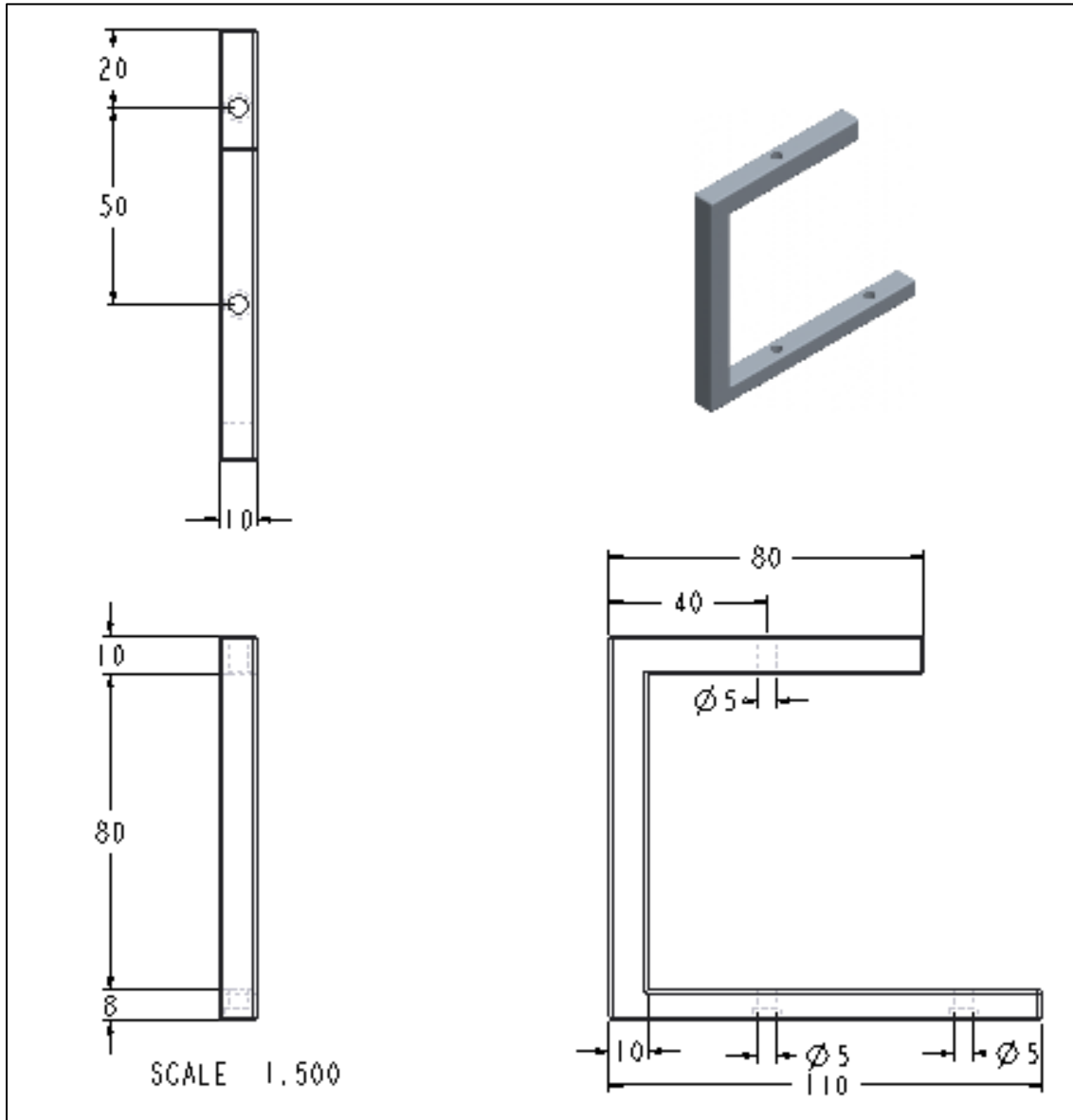


Figura A.3.2: Pieza #2 para soporte de la placa #1 (Creado por el autor en CREO Parametric)

A.3.3. Pieza #3

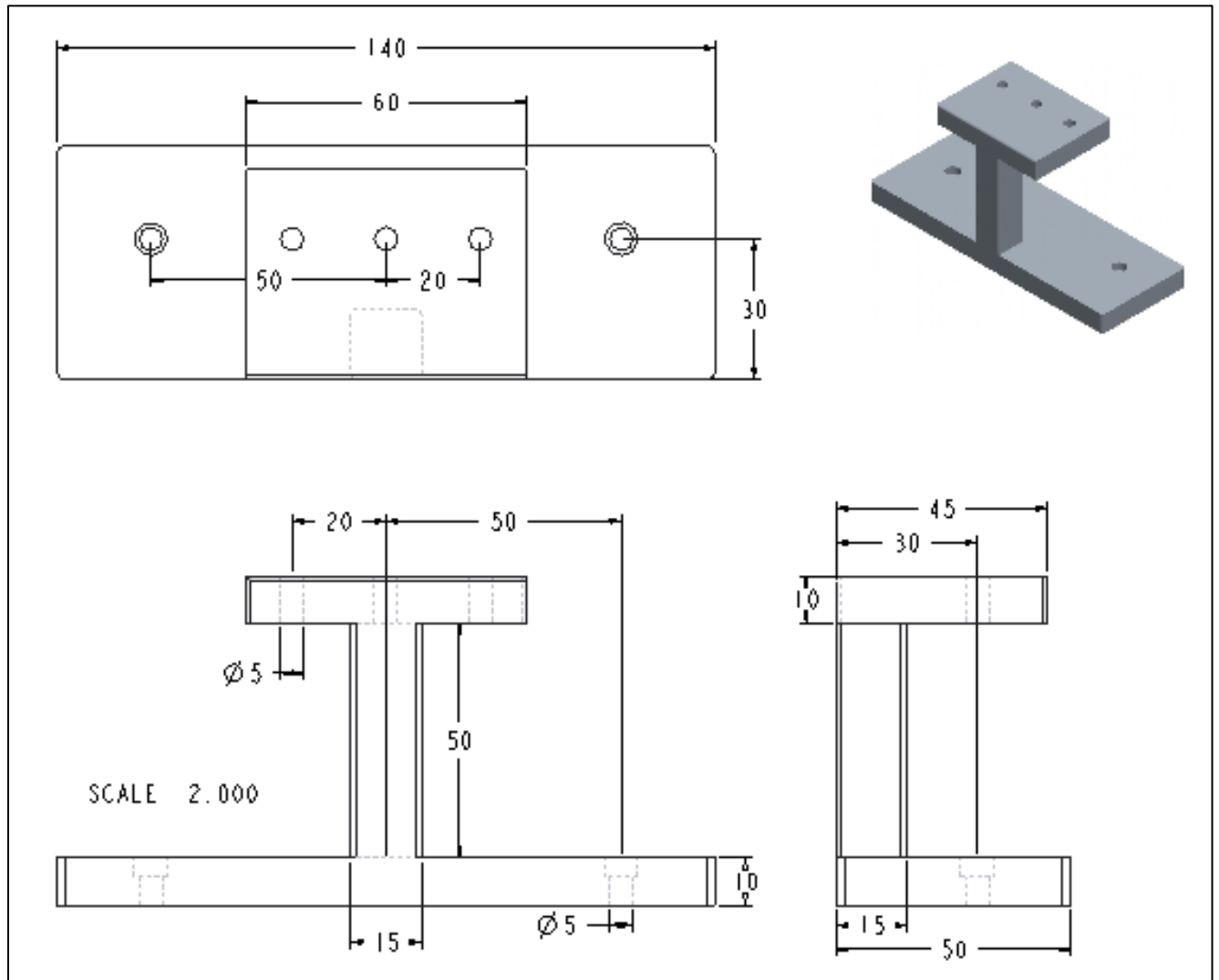


Figura A.3.3: Pieza #3 para unión entre pieza #2 y pieza superior (Creado por el autor en CREO Parametric)

A.3.4. Pieza #4

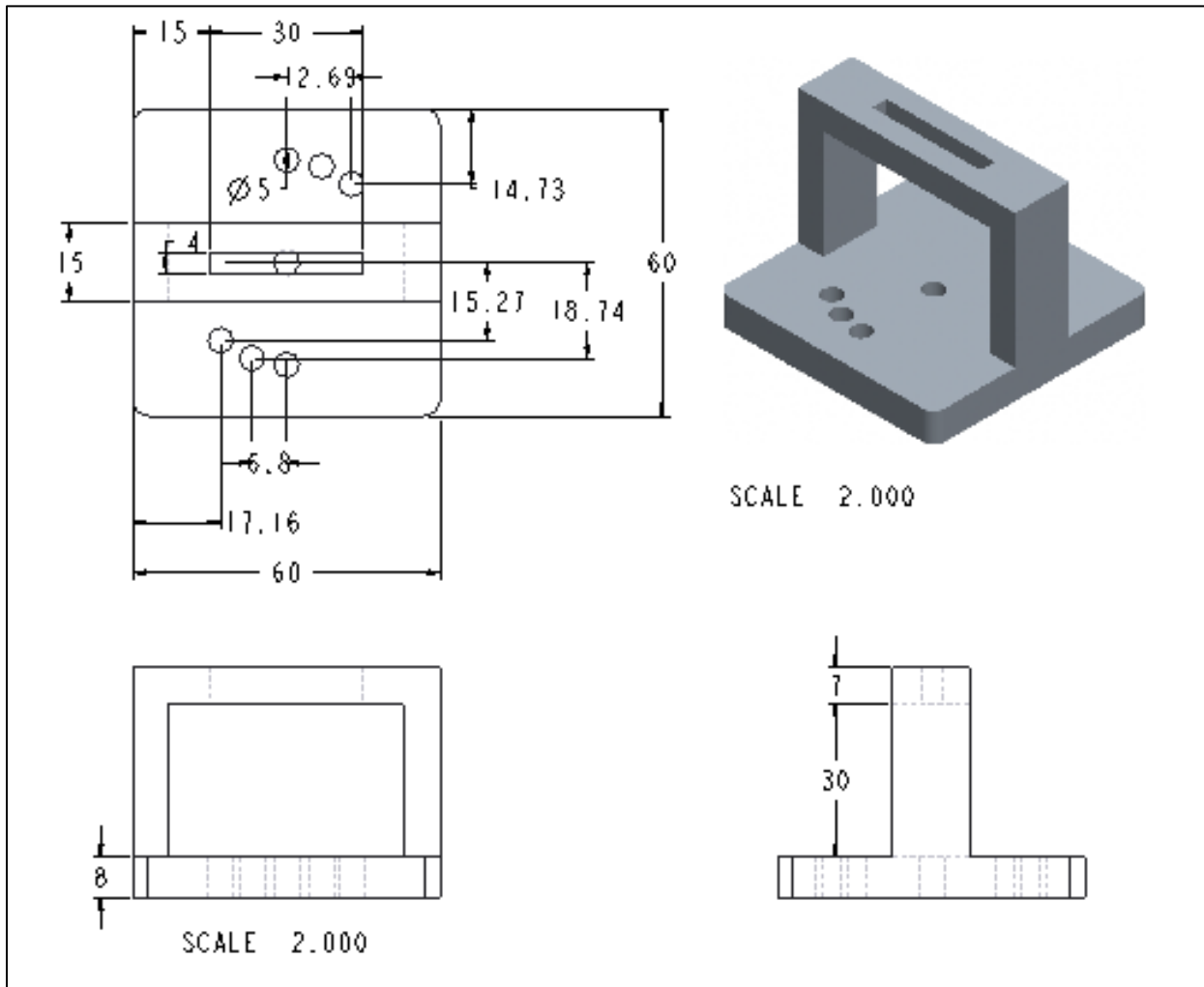


Figura A.3.4: Pieza superior para sostener la estructura en el techo del carro (Creado por el autor en CREO Parametric)