UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

“DISEÑO DE UN RECIPIENTE INTELIGENTE ATRAPA BASURA”

TESIS

QUE PARA OBTENER EL TITULO DE:

INGENIERO MECÁNICO

PRESENTA:

ISACC MORALES CASTILLO

ASESOR:

M. en I. HUMBERTO MANCILLA ALONSO

Nezahualcóyotl, estado de México 2016

AGRADECIMIENTOS

A mi familia, y en especial a mis padres, Verónica y Fidencio, por todo el apoyo brindado, a lo largo de toda mi vida y en todos los aspectos de la misma.

A los integrantes del Club de Mecatrónica, por la ayuda y amistad brindadas hacia mi persona y al proyecto presentado, pero en especial a mi asesor, el M. en I. Humberto Mancilla Alonso, por todos los consejos y sugerencias ofrecidas.

A mis mejores amigos Elba, Cuauhtémoc, Daniel y Eric por estar siempre cerca y ser parte indispensable de mi vida.

A todos mis profesores y escuelas, en especial a la Universidad Nacional Autónoma de México, por la formación académica y profesional que de ellos obtuve.

A todas esas personas que en algún momento me hicieran tropezar, porque gracias a ello, aprendí a levantarme.

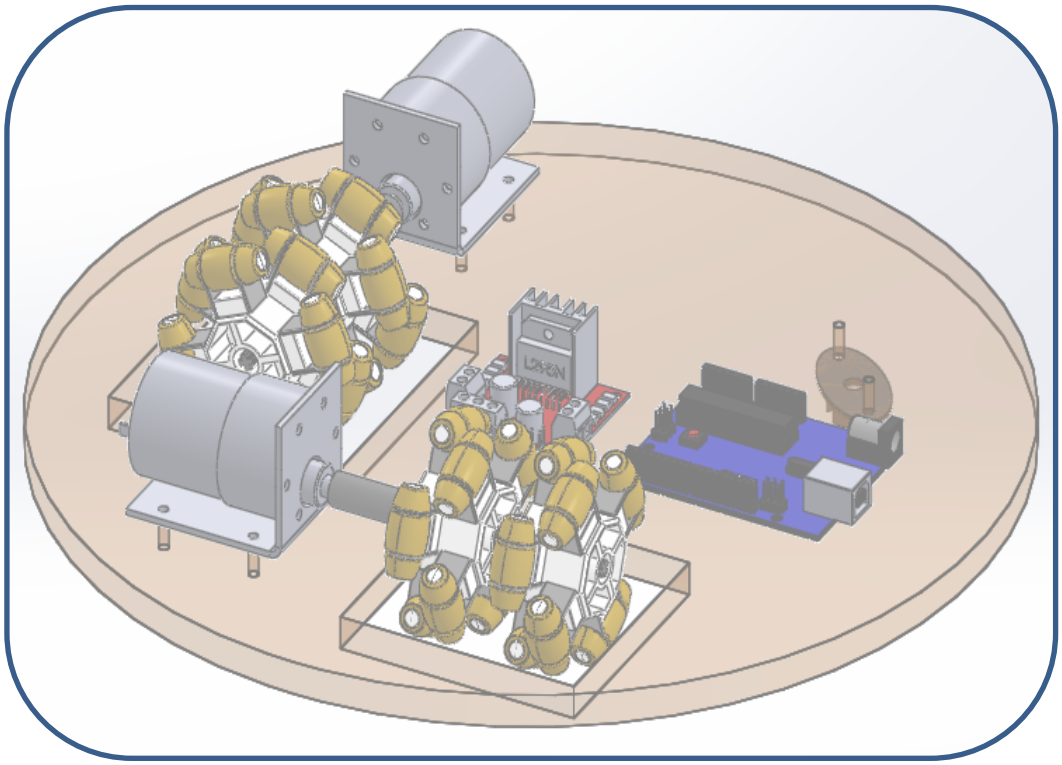
"I DON'T RIDE A BIKE TO ADD DAYS TO MAY LIFE,

I RIDE TO ADD LIFE TO MY DAYS"

DISEÑO DE UN RECIPIENTE INTELIGENTE ATRAPA BASURA

ÍNDICE

CAPÍTULO I INTRODUCCIÓN



"La vida es como andar en bicicleta, para mantener el equilibrio hay que seguir avanzando"
(Albert Einstein)

Justificación

Este proyecto se llevó a cabo porque se pretende disminuir el contacto de los alumnos con la maquinaria involucrada en las pruebas de tensión y compresión, facilitando la toma de lecturas y reduciendo posibles riesgos.

Objetivo

Emplear el sensor de videojuegos Kinect como herramienta de mediciones digitales no invasivas de probetas empleadas en pruebas de tensión y compresión, para establecer una base del sistema de visión artificial para el robot inteligente atrapa basura.

Estado del arte

El propósito con el que se desarrolló Kinect era el de dar a los jugadores de Xbox la posibilidad de que controlaran los juegos utilizando una interfaz natural, utilizando su propio cuerpo, basarse en gestos y en comandos de voz. Sin embargo ya se han encontrado muchas aplicaciones distintas.

En noviembre de 2010 (el mismo mes de su lanzamiento mundial), los drivers de Kinect fueron hackeados por un desarrollador independiente de nombre: Héctor Martín, quien ganó un concurso internacional, que tenía por objetivo hackear el ya mencionado sensor, este concurso fue promovido por Industrias Adafruit, quien ofreció una recompensa por el controlador de código abierto. Esto significa que se obtuvo una solución capaz de utilizar Kinect sin necesidad de pasar por la consola Xbox. De esta forma, el dispositivo puede funcionar también con un ordenador y no solo con la consola de Microsoft. Al desarrollo de Martín le siguieron algunos más.

Esto permitió, entre otras cosas, algunos avances, como crear una imagen 3D y controlar un robot usando Kinect.

Sin embargo, existen muchos otros proyectos, muy curiosos basados en dicha interfaz y no orientados al mercado de videojuegos, se presentan a continuación los 6 considerados más interesantes:

Otros usos del Kinect

Mover una computadora con las manos

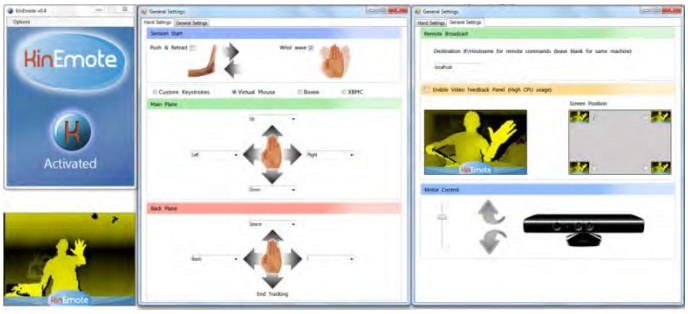


Figura 1.1. Captura de pantalla “Kinemote”

Uno de los usos alternativos más completos es la aplicación para Windows: “KinEmote” (figura 1.1), que permite controlar un aplicación de media center de forma gestual. El código de KinEmote está disponible para descarga para usuarios que quieran experimentar con Kinect y dispongan de un ordenador con Windows 7 (figura 1.2).

Lo más espectacular es la posibilidad de control de un dispositivo de modo similar a como lo realizaba el actor Tom Cruise en la película de ciencia ficción "Minority Report". Varias aplicaciones sirven para simular esta forma de manejo del ordenador, basada en gestos naturales. Es posible controlar una computadora con Windows 7 o bien interactuar solo con fotografías: gestionarlas, aumentarlas o pasarlas de una carpeta a otra.

Figura 1.2. Usuario de “Kinemote”

Una aplicación más potente basada en la gestualidad de "Minority Report" se ha desarrollado por parte de investigadores del Instituto Tecnológico de Massachusetts (MIT), que han conseguido un control más preciso de ordenadores mediante gestos con las manos más discretos y menos teatralizados que en otras soluciones.

Analista de compras en supermercados

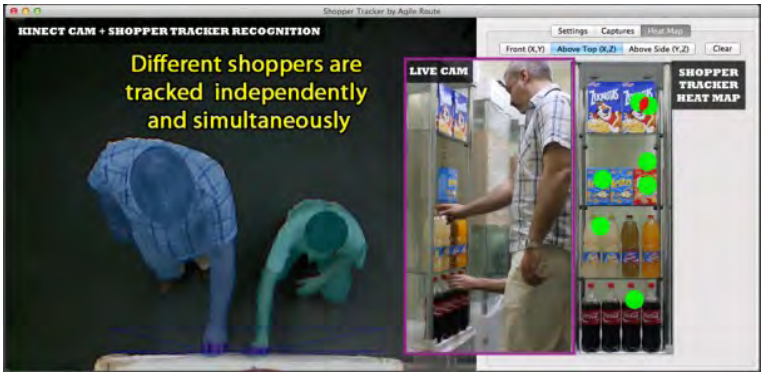


Figura 1.3 Captura de pantalla de “Shopper tracker” (1)

Se trata de un desarrollo argentino realizado por Agile Router, una empresa argentina, que permite utilizar Kinect para analizar comportamientos en tiempo real de clientes en establecimientos tipo supermercado (figuras 1.3 y 1.4). La colocación del accesorio de Microsoft en los sitios que quisiéramos medir permitirá obtener la medición a bajo costo y sin contaminar la experiencia de compra. El software de nombre: “Shopper Tracker”, puede reconocer

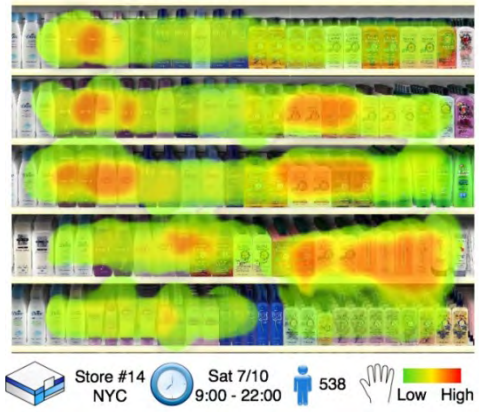


Figura 1. 4 Captura de pantalla de “Shopper tracker” (2)

a más de una persona a la vez, su posición, el lugar al que apunta con la mano y los objetos que elige.

Gracias a los sensores Kinect, la plataforma realiza un reconocimiento espacial en 3D que proporciona información sobre el ciclo de compra. Permite registrar el tiempo que un cliente emplea frente a un lineal de una tienda, qué productos mira, qué producto escoge al final y otras variables.

El registro de datos se realiza sin molestar al cliente, y posteriormente la información es procesada.

Cuadricóptero gestual

En 2011, especialistas del laboratorio "Flying Machine Arena" desarrollaron en el Instituto Federal de Tecnología de Zurich, el software para asociar los controles de un cuadricóptero a los movimientos que un operador realiza frente al Kinect de Microsoft (figura 1.5).

Figura 1.5. Cuadricóptero en funcionamiento (1)

Cuando el investigador levanta un brazo, el cuadricóptero despega. Luego puede controlar el movimiento moviendo su brazo. Un aplauso con las dos manos hace que el aparato aterrice. El cuadricóptero no puede chocar con el usuario gracias a una "zona de no volar" configurada alrededor de él (figura 1.6).

Figura 1.6. Cuadricóptero en funcionamiento (2)

Este experimento se diferencia de otros proyectos en que se utiliza el Kinect para que los robots se manejen solos: que detecten objetos y no choquen. Por el contrario, este permite que las personas sean las que controlen el movimiento de la nave. Se trata de un sistema bastante intuitivo para manejar.

Figura 1.7. Transeúnte posando

Escaneo e impresión 3D

En esta ocasión se utiliza el Kinect para crear figuras 3D en Rambla, una popular avenida barcelonesa, el proyecto ha sido bautizado como “Be Your Own Souvenir”.

Se trata de una propuesta de índole artística y busca generar muñecos con las figuras de la gente que observa (figura 1. 7). Para lograrlo, se realiza un escaneo con tres Kinects cuando alguien se coloca en el centro de la instalación, luego una impresora 3D replica en plástico las siluetas que fueron capturadas (figura 1.8).

Esta interesante aplicación ha sido desarrollada por los investigadores de blablabLAB, utilizando software desarrollado por openFrameworks y openKinect, el concepto se basa en que los visitantes se colocan en centro de un arreglo de tres Kinects para generar un escaneo completo de 360 grados, y después de unos momentos una impresora 3D genera la pequeña escultura con la misma forma de la persona escaneada.

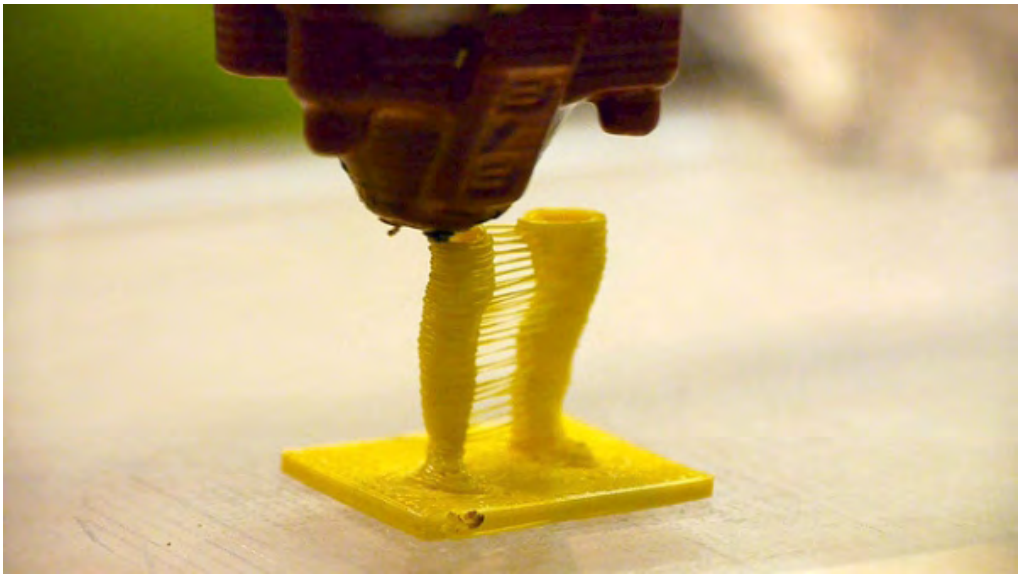


Figura 1.8. Impresión 3D

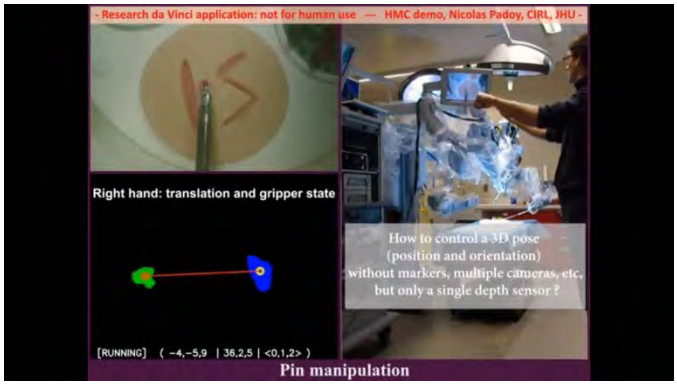


Figura 1. 9. Impresiones de pantalla e investigador manipulando el robot (1)

Uno de los sistemas robóticos quirúrgicos más conocidos es el robot cirujano Da Vinci, qué, desde su lanzamiento en 1999, está presente ya en más de 1200 quirófanos de todo el mundo.

El sistema Da Vinci consta de una consola que es controlada por el cirujano y con la cual maneja los instrumentos del robot, que se sitúa en la mesa de operaciones en la que se encuentra el paciente.

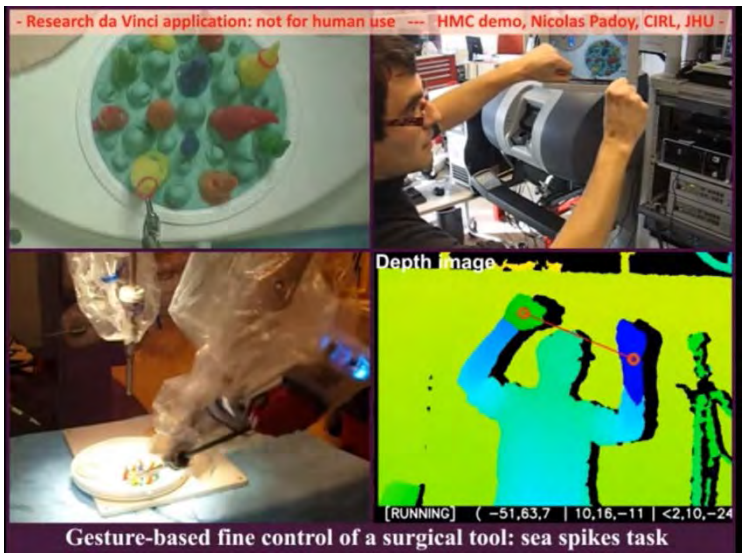


Figura 1.10 Impresiones de pantalla e investigador manipulando el robot (2)

Este proyecto se trata de una adaptación hecha por ingenieros de la universidad de Johns Hopkins, de Baltimore, en Estados Unidos, de los mandos del robot Da Vinci de alta precisión. El robot ya de por sí es una maravilla pero con la ayuda de "Kinect" (figura 1. 9) permite que el médico pueda controlarlo de una manera mucho más intuitiva, sin perder precisión y con la posibilidad de hacerlo de manera remota.

Las posiciones en 3D de las manos del cirujano son analizadas utilizando la información de profundidad obtenida del Kinect (figura 1. 10) y se utilizan para controlar la posición 3D de la herramienta robótica. Este enfoque permite traducir los 3 grados de libertad de las manos en los gestos que pueden controlar un dispositivo que cuenta con seis grados de libertad.

Simulación topográfica

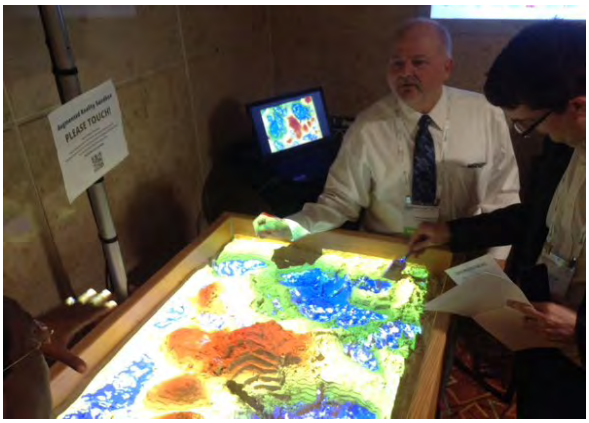


Figura 1. 11. Caja de arena de realidad aumentada (1)

La revista militar estadounidense "Marine Corps Times" desvela que "Design Interactive", una de las contratistas para las fuerzas armadas estadounidenses, está usando Kinect para un programa de realidad aumentada (figura 1.11).

La tecnología se estaría aplicando específicamente a la simulación de terrenos arenosos para los soldados estadounidenses y su primera demostración se ha realizado en la Modern Day Marine Expo celebrada en la base de los marines de Quantico (Virginia).

Figura 1.12. Caja de arena de realidad aumentada (2)

La tecnología se ha mostrado bajo el nombre de "mesa de arena de realidad aumentada" y usa un cajón de arena, un proyector y un sistema Kinect de Microsoft (figura 1.12). El sensor detecta las características de la arena y proyecta un mapa topográfico que se corresponde con la información tridimensional captada por Kinect.

Esto permite, por ejemplo, captar imágenes de Google Earth u otro sistema de satélites con tecnología similar y visualizar el terreno con información topográfica precisa durante ejercicios militares. El objetivo es que pueda usarse finalmente en operaciones reales y no solo en simulaciones y entrenamientos.

Todo esto se realiza en tiempo real, viendo cómo afectaría a la geografía del entorno distintos parámetros, como flujo de agua forestación, viento entre otros. Algo que a simple vista sería un juguete para niños se transforma en un portal al futuro con una pequeña dosis de tecnología.

Titiritero

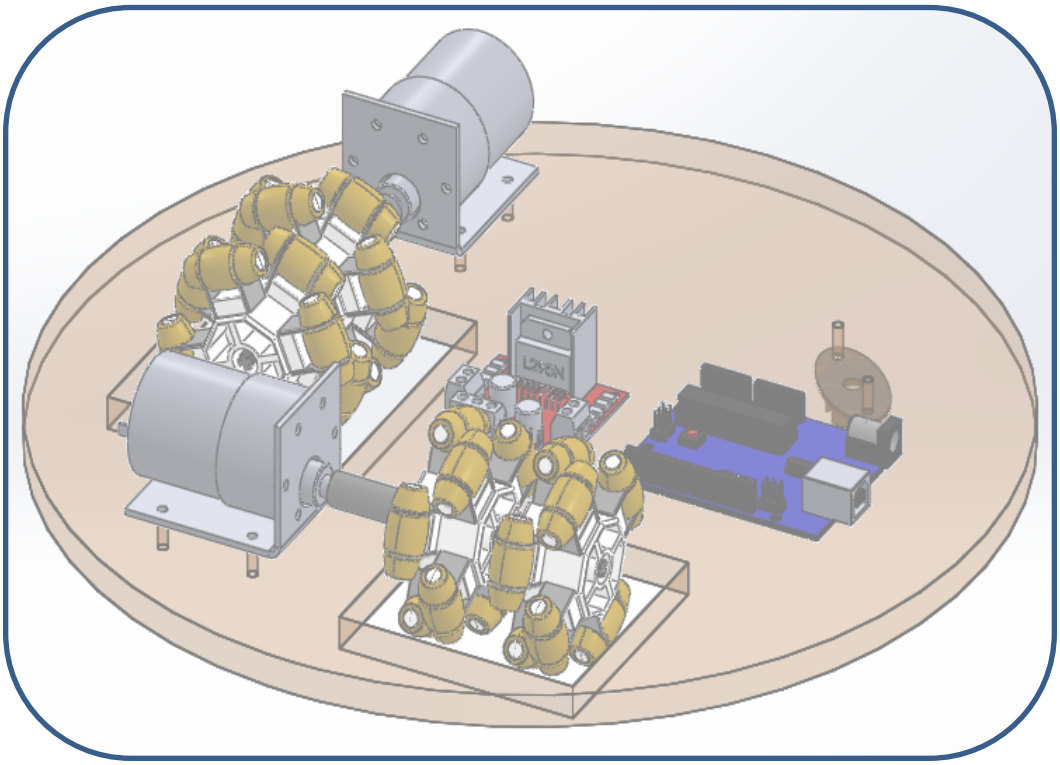
Figura 1.13. Captura de pantalla que muestra el control del títere.

Emily Gobeille y Theo Watson, diseñadores de "Design I/O", una empresa que se especializa en instalaciones interactivas para eventos, han creado un prototipo de títeres de sombras usando el Kinect (figura 1.13), este desarrollo combina el accesorio con un proyector tradicional produciendo un interesante espectáculo de marionetas. La persona que se ubique frente a la cámara infrarroja podrá controlar con los gestos de sus manos las figuras creadas por el ordenador, la proyección puede realizarse sobre quien hace la interpretación o en otra pared.

El sistema openFrameworks rastrea el codo, la muñeca, el pulgar y las puntas de los dedos para asignar un esqueleto sobre el movimiento y la postura de un muñeco animado (figura 1.14).

Figura 1.14. Persona manipulando títere de sombras.

CAPÍTULO II BASES TEÓRICAS



"Nuestra gloria más grande no consiste en no haberse caído nunca,
sino en haberse levantado después de cada caída"
(Confucio)

Sensor Kinect

El Kinect es un dispositivo que funciona como mando o control de videojuegos, fue puesto a la venta en Noviembre de 2010 por la empresa Microsoft, como accesorio de la consola “Xbox 360”. La característica principal de este accesorio es que permite a los usuarios interactuar con la interfaz en pantalla propia de cada videojuego, o de la misma consola sin necesidad de ningún control físico, sino a través de la interfaz de usuario natural, capaz de procesar gestos y comandos de voz. La apariencia de este dispositivo es la se muestra en la figura 2.1.

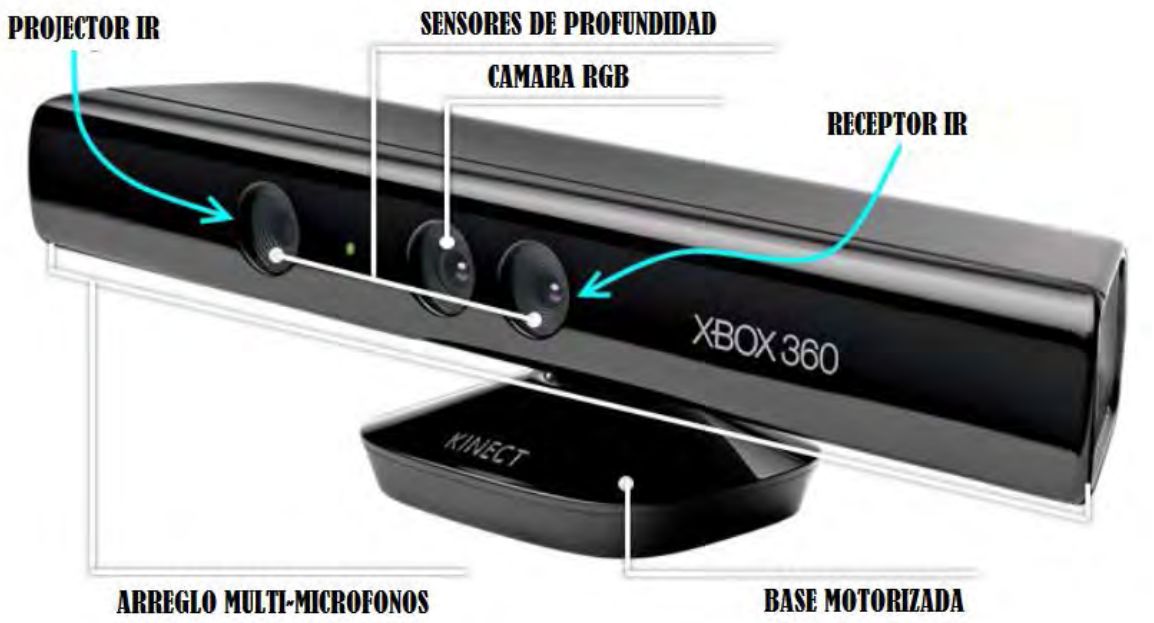


Figura 2.1. Sensor Kinect

Datos del Kinect

Los elementos principales que componen el sensor son cuatro: una cámara RGB, un sensor de profundidad que consta de un proyector IR y un sensor receptor IR; un motor para controlar la inclinación del dispositivo y un arreglo de cuatro micrófonos distribuidos de manera interna a lo largo del sensor. En este proyecto se utilizan principalmente el sensor de profundidad y la cámara RGB, cuyas características se describen a continuación.

Cámara RGB: Dependiendo del software de desarrollo, SDK por sus siglas en inglés (Software Development Kit) que se utilice, es posible trabajar con dos formatos de imagen: formato RGB (resolución de 640x480 (VGA) o 1280x1024 píxeles, hasta 30 imágenes por segundo (fps por sus siglas en inglés (frames per second))) y formato YUV (resolución de 640x480 píxeles 15fps).

Sensor de profundidad: Kinect está basado en el uso de luz estructurada de infrarrojos para su funcionamiento. La fuente de luz infrarroja es un láser más una rejilla de difracción que proyecta un patrón de puntos mismos que son interpretados por un sensor de infrarrojos monocromático (CMOS). Se detectan los puntos reflejados en los objetos del escenario y se estima su profundidad a partir de la intensidad y la distorsión de los mismos, sin embargo no hay ninguna publicación oficial en relación al algoritmo que se encarga de estimar la profundidad de cada uno de los puntos. En la figura 2.2 se muestra un patrón de puntos proyectados por el sensor Kinect los cuales se utilizan para realizar la medición de profundidad, según ROS (Robot Operating System), el algoritmo comienza calculando la profundidad de un plano de referencia a partir de los nueve puntos que aparecen más marcados. Después, la profundidad para cada pixel se calcula eligiendo una ventana de correlación pequeña (9x9 ó 9x7) y se compara el patrón local en ese píxel con el patrón memorizado en ese píxel y los 64 píxeles vecinos en una ventana horizontal.

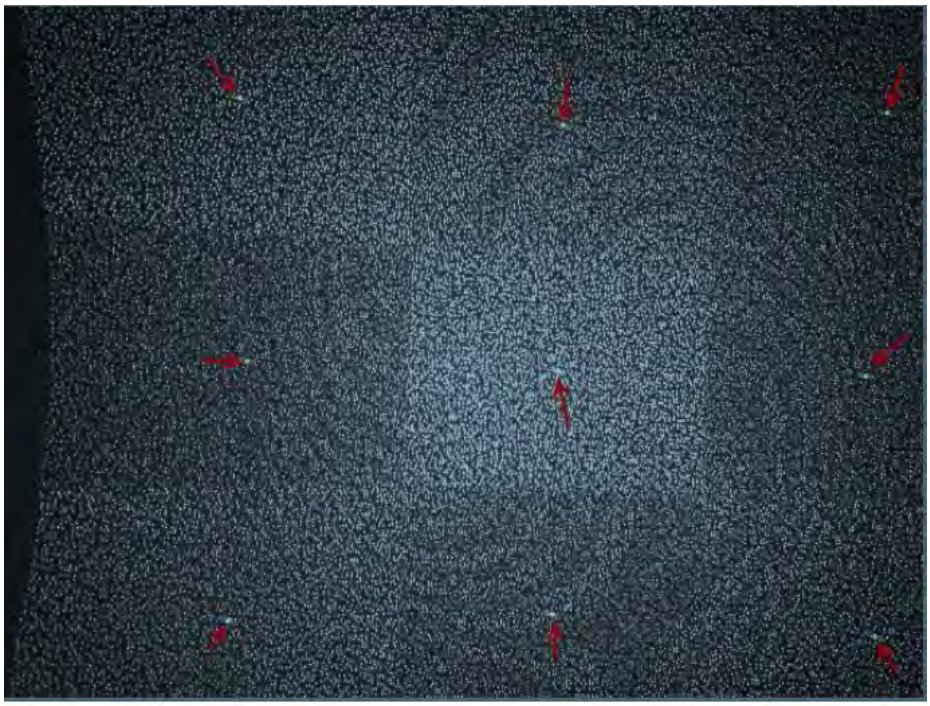


Figura 2.2. Patrón de puntos emitidos por el sensor Kinect

Para un funcionamiento correcto, deben satisfacerse las siguientes condiciones teóricas:

- Máxima (mínima) distancia del objeto al sensor: 3.5 (.4) metros.
- Los objetos deben estar dentro del ángulo de visión: ±43º verticalmente y de ±57º horizontalmente.

Sin embargo, el sensor tiene ciertas limitaciones relacionadas con el tipo de objeto, tales como su forma (cóncavo, cilíndrico, esférico, etcétera) o su apariencia (brillante, translucido, etcétera).

Limitaciones: El sensor Kinect tiene varias limitaciones que hacen que la profundidad de ciertas regiones de la escena no se pueda estimar o si se estima, la fiabilidad de los datos no es aceptable. Estas limitaciones vienen condicionadas tanto por factores internos, debidos a la arquitectura del dispositivo; como externos, debidos a la naturaleza de la escena. En el primer grupo (factores internos) se tienen las siguientes limitaciones:

Los puntos de luz no cubren de forma continua la superficie de los objetos, lo que conlleva a que algunos píxeles de la imagen de profundidad tienen que ser interpolados. Esto implica que el valor de profundidad de un píxel determinado tiene asociado un margen de error. Este margen es mayor cuanto más alejado está el objeto, puesto que, para una misma superficie, los puntos de luz están más separados. A mayores distancias, los valores de profundidad devueltos para objetos cercanos entre sí tienden a ser muy similares. Sin embargo, si el objeto está a demasiada distancia del sensor, no se calcula ninguna distancia para ese punto. Esto ocurre así, porque la potencia de luz del haz de infrarrojos se atenúa en el trayecto recorrido, haciendo que sea imperceptible para el sensor receptor de infrarrojos.

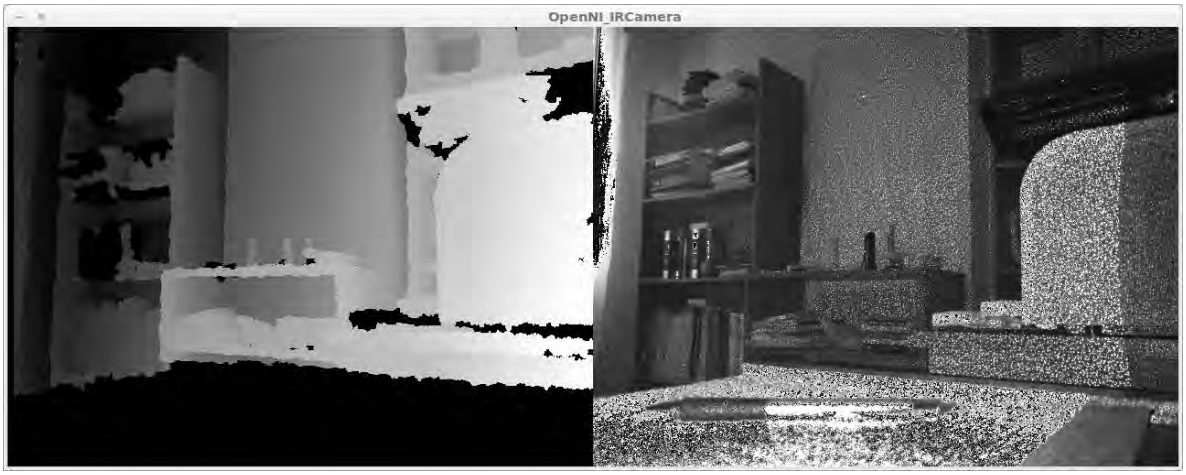


Figura 2.3. IZQ.: imagen de profundidad con sensor Kinect.
DER: imagen en escala de grises

En la figura 2.3 se muestra una imagen de profundidad para distancias muy grandes e inclinaciones que distorsionan la percepción de profundidad.

Del problema descrito en el punto anterior se deriva también una imprecisión en los bordes de los objetos.

Dado que para la estimación de la profundidad considera más de un píxel y al estar éstos dispersos, unas veces se tomará la profundidad del objeto más cercano y otras las del más lejano, lo que genera errores en las mediciones.

Los objetos cóncavos y las cavidades reflectantes pueden producir reflexiones dobles e inter-reflexiones lo que genera puntos negros, o pixeles cuya información es ilegible y se muestra como un pixel de valor 0.

Es decir, un punto de luz que caiga sobre una superficie cóncava, puede rebotar en otra zona del mismo objeto (cóncavo), lo que produce un solapamiento de los puntos de luz, haciéndolos nuevamente irreconocibles para el sensor.

Las zonas cuya profundidad no pudo ser resuelta, aparecerán como zonas negras (píxeles de valor cero) en la imagen de profundidad. Aparte, como ya se mencionó anteriormente, también hay que tener en cuenta la atenuación que sufre la luz, por lo que para objetos muy lejanos, tampoco se podrá determinar su profundidad.

Asimismo, la inclinación de la superficie de los objetos respecto al proyector del haz de luz limita la detección de la profundidad. Si el rayo de luz es casi paralelo a la superficie no podrá incidir sobre la misma, haciendo imposible la estimación de la profundidad. En la figura 2.3 se puede ver este efecto en la mesita de la parte inferior de la pantalla.

Procesamiento digital de imágenes

El procesamiento digital de imágenes es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar su aspecto y hacer más evidentes en ellas ciertos detalles que se desean hacer notar.

El procesamiento de las imágenes se puede hacer por medio de métodos ópticos, o bien por medio de métodos digitales.

Para interés del proyecto solo se dará una breve explicación del procesamiento digital. Este se efectúa dividiendo la imagen en un arreglo rectangular de elementos, cada elemento de la imagen se conoce con el nombre de pixel. El siguiente paso es asignar un valor numérico a la luminosidad promedio de cada pixel. Así, los valores de la luminosidad de cada pixel, con sus coordenadas que indican su posición, definen completamente la imagen. Existen varias técnicas de obtención de imágenes 3 D, las más comunes se describen brevemente a continuación:

Multicámara

El objetivo de esta técnica es captar el objeto o la escena con varias cámaras calibradas para obtener diferentes puntos de vista y así generar datos de profundidad (figura 2.4).

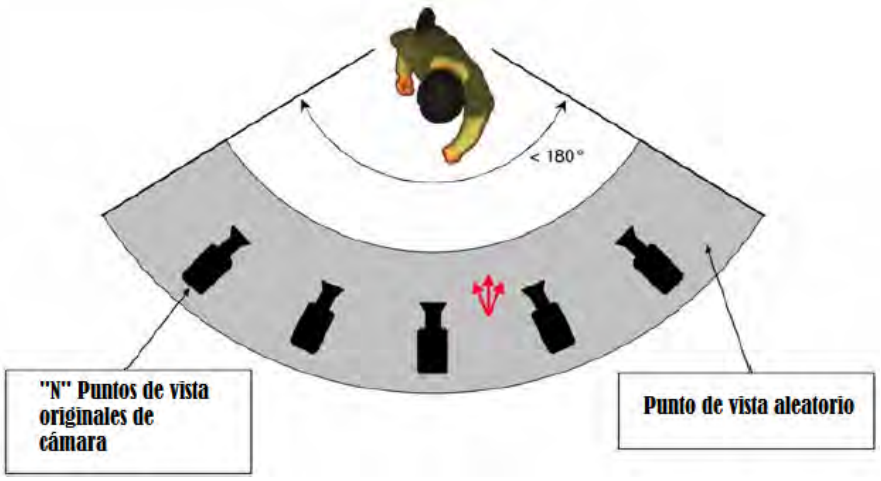


Figura 2.4. Arreglo multi cámaras para generación de imágenes 3D

Cámaras estereoscópicas

Figura 2.5. Cámara estereoscópica

Una cámara estereoscópica es nombrada así debido a la visión estereoscópica humana; es una cámara capaz de capturar imágenes en tres dimensiones. La visión binocular humana, produce dos imágenes, una por cada ojo, que luego se mezclan en el cerebro creando la imagen 3D. Estás cámaras, intentan imitar este comportamiento (figura 2.5), utilizando dos objetivos o dos cámaras separadas estratégicamente, captando la fotografía en el mismo instante, y como resultado se obtienen las imágenes 3D.

Escáner de luz estructurada

El escáner de luz estructurada es un dispositivo capaz de capturar la forma y características de un objeto mediante la proyección de un patrón de luz y su registro en un sistema de adquisición, mismo que está compuesto por una fuente de luz (que proyectara un haz luminoso) y una cámara (que captará los puntos en las superficies) separados entre sí (figura 2.6).

Para escanear el objeto, se define un sistema de coordenadas esféricas para determinar cada punto del espacio tridimensional que se está capturando.

Los sistemas de luz estructurada se basan en estudiar la deformación que sufre un patrón de luz al ser interceptado por cualquier objeto. Una de las mejoras soluciones es emplear un haz láser. Debido a sus características de coherencia, divergencia, y direccionalidad.

Figura 2.6. Capturas de imagen con luz estructurada

Time of flight (tiempo de vuelo)

Es un método para extraer la información de profundidad de una escena para así poder crear una imagen 3D. Consiste en que la cámara emite una señal modulada en el espectro infrarrojo. Esta señal incide sobre la escena y vuelve rebotada sobre la cámara. En cada pixel de la cámara se puede interpretar la señal, y a través de esa interpretación detectar la distancia. La cámara genera una imagen en escala de grises que nos da la información de profundidad.

En la figura 2.7 se describe de forma gráfica el funcionamiento de este método.

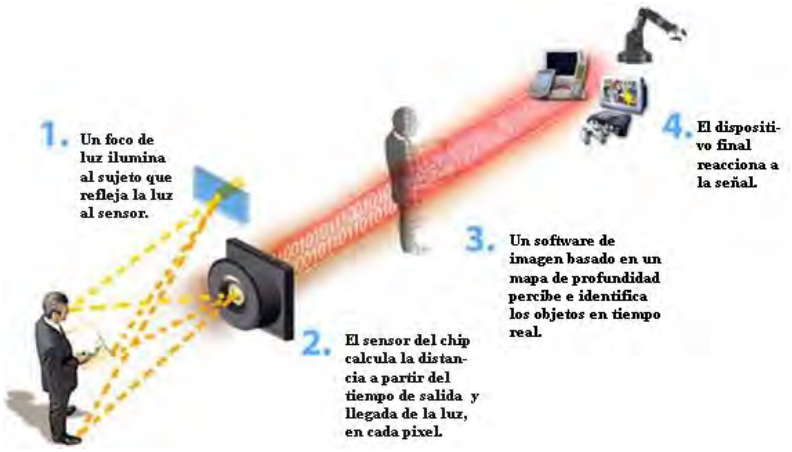


Figura 2.7. Time of flight

La versión más simple de este tipo de cámara utiliza pulsos de luz. La iluminación se conecta durante un tiempo muy corto, el pulso de luz resultante ilumina la escena y es reflejada por los objetos dentro del campo de visión. El lente de la cámara recoge la luz y las imágenes que se generan con este reflejo, Dependiendo de la distancia, la luz entrante experimenta un retraso, y este retraso genera la información para obtener la distancia de los objetos.

Como ya se había mencionado antes, el sensor Kinect utiliza un sensor de infrarrojos monocromático CMOS (semiconductor complementario de óxido metálico), también conocido como sensor de pixeles activos, que es un sensor que detecta la luz, basado en el efecto fotoeléctrico. Está formado por numerosos fotodiodos, uno para cada pixel, que producen una corriente eléctrica que varía en función de la intensidad de luz recibida.

Posterior al procesamiento de imágenes se deriva la visión artificial, pues el primero tiene su uso principal en las artes gráficas auxiliadas por computadora, y la visión artificial tiene un mayor uso en la industria, tal es así que, incluso es denominada visión máquina.

Visión artificial

Las primeras investigaciones en visión artificial comenzaron en los años 70's, siendo una tecnología militar en primera instancia usada para analizar las imágenes que se captaban vía satélite, fue hasta mediados de los años 90's que se le dio un uso de tipo industrial, esto debido a que los costos de implementación bajaron.

La visión artificial se define como un campo de la inteligencia artificial que, permite la obtención, procesamiento y análisis de información obtenida mediante la captura de imágenes digitales.

Consiste en la adaptación de imágenes digitales mediante cámaras especializadas y su posterior tratamiento a través de técnicas de procesamiento, permitiendo así poder intervenir en un proceso o producto para el control de calidad y seguridad de toda la producción.

Los objetivos de un sistema de inspección por visión artificial suelen ser comprobar la existencia de ciertos requisitos en un a pieza o conjunto de piezas, tales requisitos pueden ser características de la o las piezas como dimensiones números de serie, presencia de componentes, colores, etc. Como se puede ver en la figura 2.8.

Figura 2.8. Inspección por visión artificial de botellas

Automatizar tareas o procesos repetitivos de inspección o ensamblaje, para mejorar el control de calidad de un producto.

Incrementar la seguridad en la inspección de objetos peligrosos al evitar el contacto físico del operador con dichos objetos.

Realizar inspecciones del 100% de la producción, lo que a su vez generara calidad total a una gran velocidad y por lo tanto una reducción en el tiempo de procesos automatizados.

Algunas de las características más relevantes de los sistemas de visión artificial son: Detectar bordes y formas, analizar el color, medir la cantidad de luz; esto se puede realizar sin necesidad de tener contacto con la pieza, evitando así la deformación del material, además de tener la capacidad de realizar el análisis mientras el objeto está en movimiento llevando a cabo todo esto de forma automática, se incrementa la velocidad del o los procesos, y, debido al uso de software se tiene la ventaja de que los sistemas de inspección sean flexibles, puesto que el software se puede alterar dependiendo de las necesidades.

Cuando un sistema de visión artificial se aplica a algún proceso industrial sigue un esquema básico, descrito a continuación en la figura 2.9.

Figura 2.9. Ciclo básico de la visión artificial usada en industria (imagen Propia basada en: Visión artificial una tecnología industrial)

La primera etapa de un sistema de visión artificial, es la captura de una imagen, para su posterior análisis, del cual, se envían resultados al o los actuadores correspondientes y en su caso a un operador, mediante una interfaz, para que los mencionados ejecuten una acción determinada, después de la cual se espera por un nuevo objeto para que el ciclo reinicie.

Dado que la visión artificial se basa en el uso de cámaras para poder captar las imágenes que se requieren, se debe hablar también de algunas características clave de las mismas, que deben ser tomadas en cuenta para poder diseñar un sistema de visión artificial que pueda cumplir con los objetivos y requerimientos que se hayan establecido.

Una de estas características es el campo de visión, que es el área completa que puede ver la cámara digitalmente, esta área depende del lente que se ocupe. Esto se muestra en la figura 2.10.

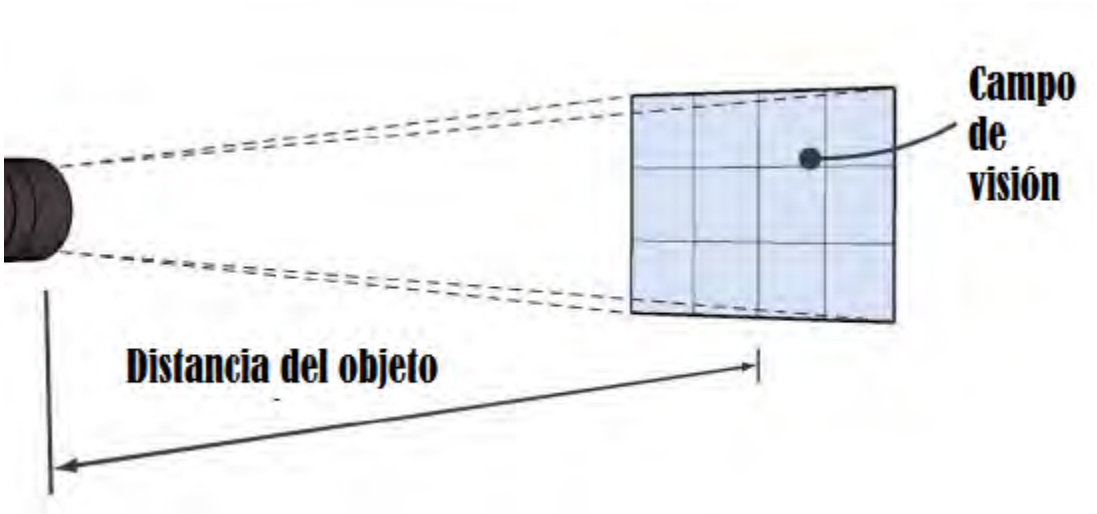


Figura 2.10. Campo de visión

Otra característica notable es la distancia focal, que es la distancia entre el centro de la lente (figura 2.11) y el punto focal (punto en el que convergen los haces de luz procedentes del objeto deseado, después de haber atravesado la

lente), sí el punto focal coincide con el centro del sensor se obtiene una imagen nítida.

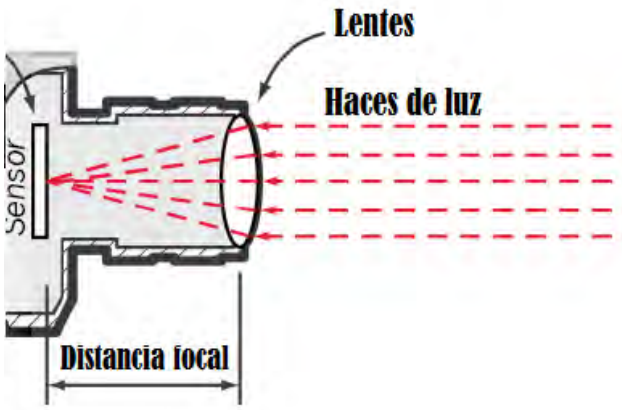


Figura 2.11. Distancia focal

Otro punto importante a considerar es el mostrado en la figura 2.12; la distancia mínima a la cual se puede ubicar el objeto que se desea analizar, pues si el objeto está demasiado cerca no podrá enfocarse y básicamente será invisible. Esta separación entre el objeto y la lente de la cámara está relacionada con la distancia focal, mientras mayor sea la distancia focal, mayor deberá ser la distancia entre la lente y el objeto.

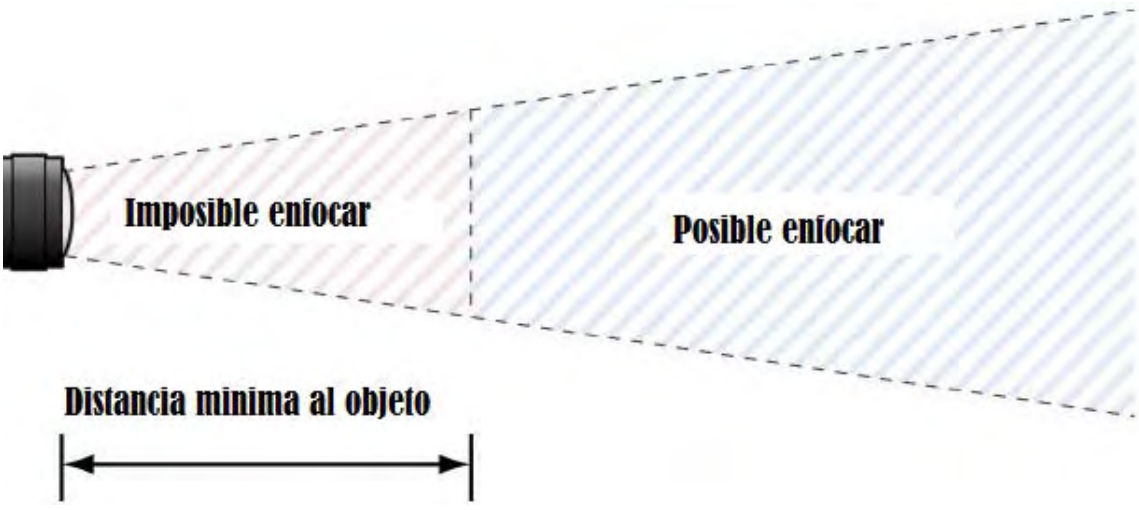


Figura 2.12. Distancia mínima entre objeto y cámara

Por último podemos hablar del plano focal, que es la distancia en la que se encuentra la mayor nitidez posible, y se denomina profundidad de campo al espacio previo y posterior al plano focal donde la nitidez obtenida aún se considera óptima o suficiente. Estas dos condiciones se muestran en la figura 2.13.

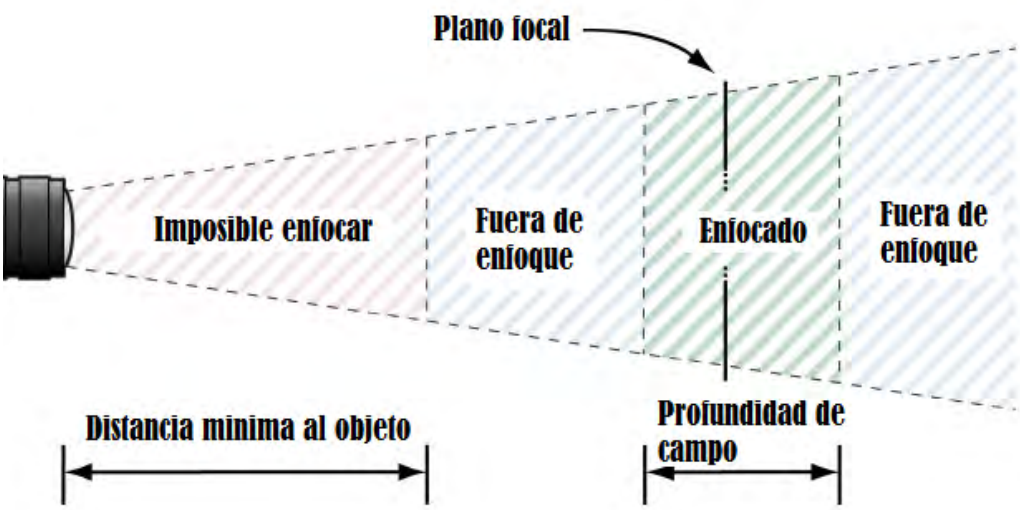


Figura 2.13. Plano focal y profundidad de campo

Infrarrojos

Como ya se ha mencionado, el sensor Kinect, utiliza luz estructurada a base de infrarrojos.

El infrarrojo es un tipo de luz indetectable al ojo humano (figura 2.14), es radiación electromagnética de baja frecuencia y longitud de onda larga más larga que la longitud de onda de la luz visible, pero menor que la de las microondas. Consecuentemente, tiene menor frecuencia que la luz visible y mayor que las microondas. Su rango de longitudes de onda va desde los 780nm hasta 1mm y el rango de frecuencias oscila entre 300 Giga Hertz y 400 Tera Hertz.

Figura 2.14. Longitudes de onda para diferentes tipos de radiación electromagnética

En 1800, el astrónomo inglés (William Herschel) descubrió los “rayos caloricos”. Herschel colocó un termómetro de mercurio en el espectro obtenido por un prisma de cristal con el fin de medir el calor emitido por cada color, Descubrió que el calor era más fuerte al lado del rojo del espectro y observo que allí no había luz. En 1880 se empezó a usar el término “infrarrojo.

Samuel Langley invento el bolómetro, el cual mide las variaciones de voltaje de una resistencia eléctrica durante el calentamiento en un detector absorbente. Los bolómetros fueron los primeros detectores de radiación infrarroja.

El ojo humano es incapaz de detectar este tipo de radiación, sin embargo los seres humanos podemos detectar parte de esta energía por medio de la piel en forma de calor.

Usos de los infrarrojos

La radiación infrarroja es emitida por cualquier cuerpo cuya temperatura sea mayor que -273.15° Celsius, por lo que uno de los usos más comunes de sensores infrarrojos es a medición de temperatura. Un sensor de infrarrojos recibe energía infrarroja generalmente con un valor de 0.0001 volt, del objeto a medir, la amplifica y en un CPU se digitaliza, arrojando datos de temperatura,

Los infrarrojos se utilizan en los equipos de visión nocturna, La radiación se recibe y después se refleja en un a pantalla. Los objetos más calientes se convierten en los más luminosos.

Otro uso común es el que hacen los mandos a distancia, que utilizan infrarrojos en vez de ondas de radio, ya que no interfieren con otras señales como las de televisión.

También son usados para enviar información, por ejemplo de una computadora a un dispositivo externo, o por medio de fibra óptica.

Infrarrojo y Kinect

Como ya se ha mencionado anteriormente, el sensor Kinect, funciona bajo cualquier condición de iluminación, siempre y cuando sea en interiores, pues no está diseñado para trabajar a la intemperie.

Sin embargo se tuvo la inquietud de corroborar que las condiciones lumínicas no afectaran el desempeño del sensor, pues el contenido espectral de la luz solar (o radiación solar) contiene radiación infrarroja lo que podría ser un problema al distorsionar la imagen si el sensor detecta esa radiación, por lo cual se procedió a realizar prueba en distintas condiciones de luz.

Tales condiciones fueron a plena luz solar, bajo iluminación artificial y en total obscuridad, estas condiciones se explican a continuación:

- A plena luz del sol: Se colocó el sensor junto a una ventana sin nada que interrumpiera la iluminación solar, primero deforma lateral, es decir, con el sol de lado (se realizaron dos pruebas, una con el sol a la derecha y otra con el sol a la izquierda) a la línea entre el sensor y la muestra, notando que la percepción del sensor se altera con la luz solar, por lo cual se buscó una posición en la cual el sensor pueda trabajar adecuadamente, y se encontró que si la luz solar cae directamente sobre la muestra esta se hace indetectable para el sensor, pero, si la muestra está en sombra o genera sombra el sensor puede detectarla sin problemas.

  A continuación se presentan las capturas de pantalla de las pruebas realizadas, donde se puede apreciar la alteración en la percepción del sensor.

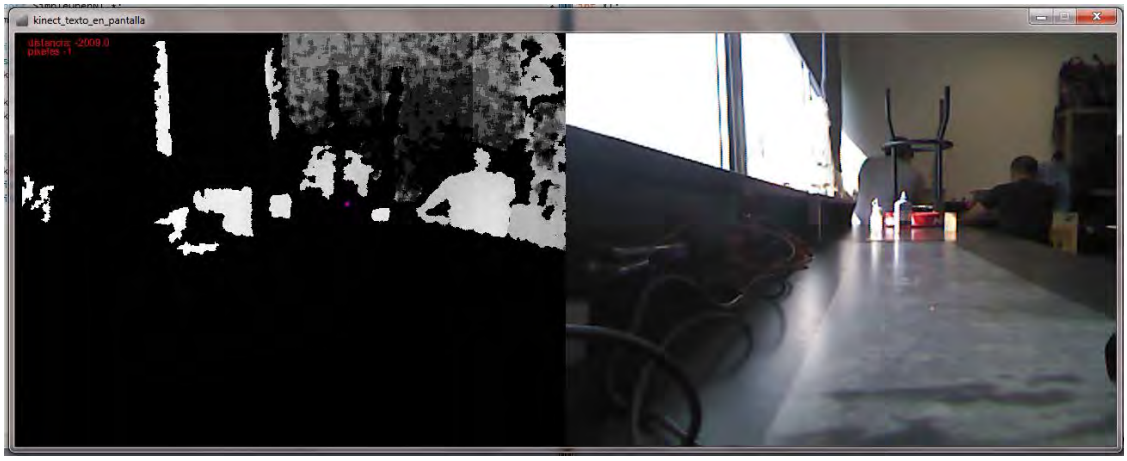


Figura 2.15. Captura de pantalla del sensor trabajando con luz solar a la derecha

En la figura 2.15 se muestra una captura de pantalla con la imagen de profundidad (izquierda) y RGB (derecha) con el sol a la derecha con una

distancia de 2 009.0mm medidos con el Kinect, en la que se nota que ninguna de las muestras es visible con la cámara de infrarrojos.

Figura 2.17. Captura de pantalla del sensor trabajando con luz solar a la derecha.

En la figura 2.17 se muestra una captura de pantalla con la imagen de profundidad (izquierda) y RGB (derecha) con el sol a la derecha con una distancia de distancia igual a 1571.0mm medidos con el Kinect. En la que se puede notar como se alteró la posición de una de las muestras, generando sombre en una de sus caras ante lo cual se hace visible al sensor, las otras muestras siguen sin ser detectadas

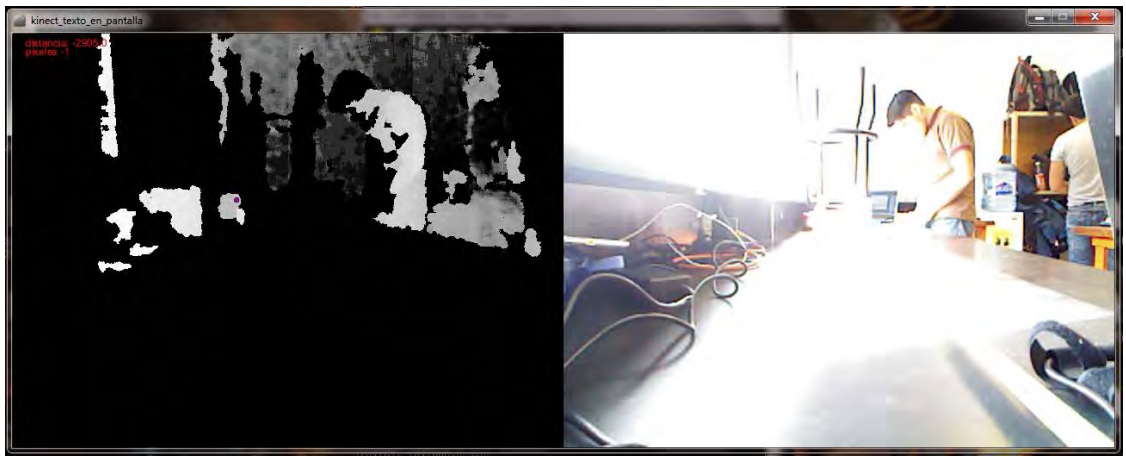


Figura 2.18. Captura de pantalla del sensor trabajando con luz solar casi de frente al sensor.

En la figura 2.18 se muestra una captura de pantalla con la imagen

de profundidad (izquierda) y RGB (derecha) con el sol a la derecha, casi de frente al sensor con una distancia de distancia igual a 1410 mm medidos con flexometro debido a que el sol altera la medición y genera que el sensor no detecte las muestras, solamente los objetos que están en zonas donde la luz solar es escasa.

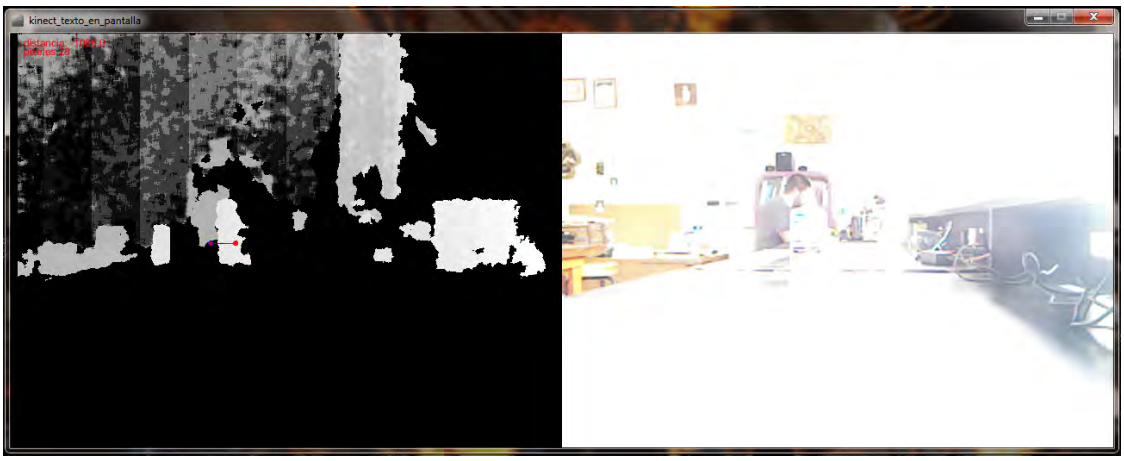


Figura 2.19. Captura de pantalla del sensor trabajando con luz solar a la izquierda.

En la figura 2.19 se muestra una captura de pantalla con la imagen de profundidad (izquierda) y RGB (derecha) con el sol de frente y un poco a la izquierda del Kinect, a un a distancia de 1086 mm medidos con Kinect, debido a que la posición de las muestras genera sombra en una de sus caras, son visibles al sensor.

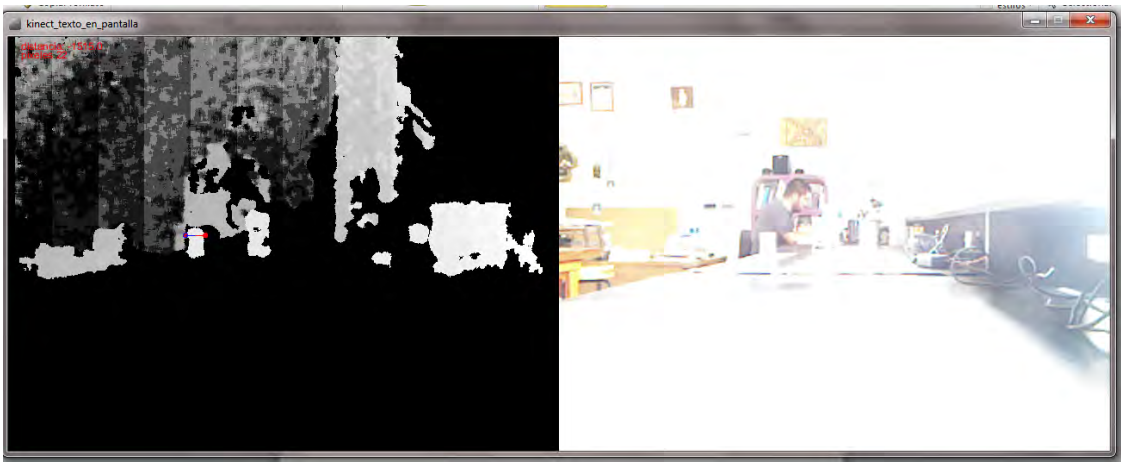


Figura 2.20. Captura de pantalla del sensor trabajando con luz solar a la izquierda.

En la figura 2.20 se muestra una captura de pantalla con la imagen de profundidad (izquierda) y RGB (derecha) con el sol de frente y un poco a la izquierda del Kinect, con una distancia igual a 1515 mm medidos con Kinect en la que se puede apreciar, que la percepción del sensor no cambia con respecto a la distancia de la imagen anterior, siendo más importante la forma en que las muestras se iluminan con la luz solar.

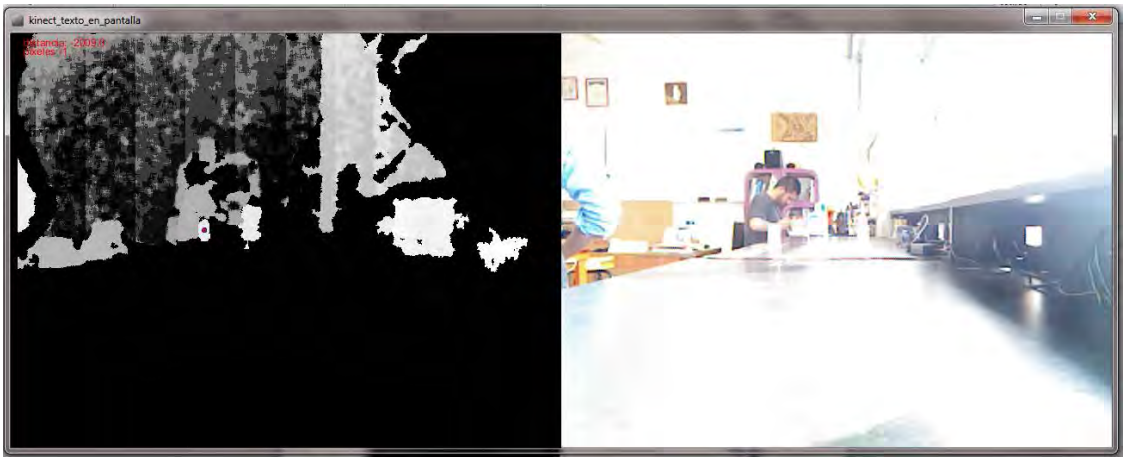


Figura 2.21. Captura de pantalla del sensor trabajando con luz solar a la izquierda.

En la figura 2.21 se muestra una captura de pantalla con la imagen de profundidad (izquierda) y RGB (derecha) con el sol de frente y un poco a la izquierda del Kinect, con una distancia igual a 2009.0mm medidos con Kinect dado que esta es la distancia que se toma como base debido a los datos teóricos, se hace una comparación cambiando las condiciones de iluminación, en la figura 2.22 se puede apreciar esta comparación, en la que se disminuye la cantidad de luz solar mediante el bloqueo de la misma por medio de persianas.

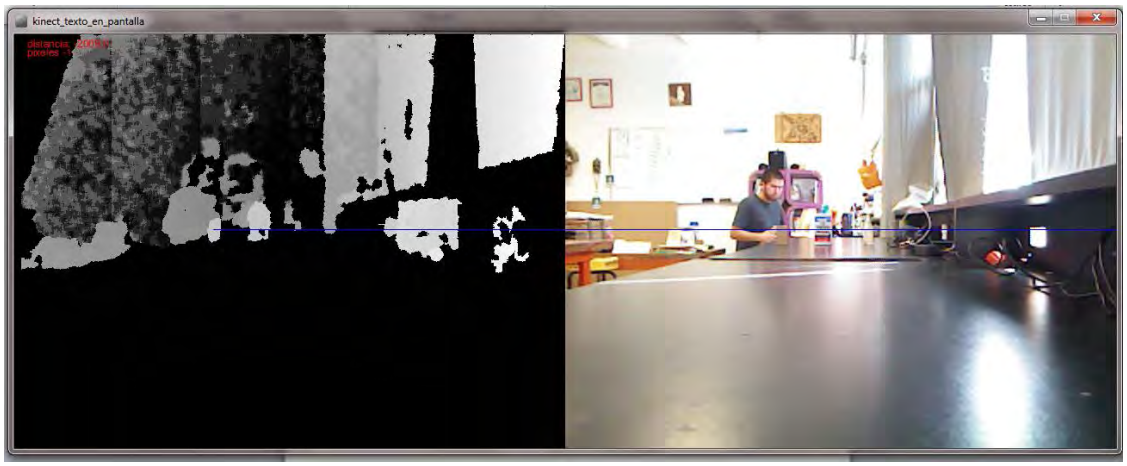


Figura 2.22. Captura de pantalla del sensor trabajando con luz solar a la izquierda.

Después de realizadas estas pruebas se determinó que la cantidad de luz si afecta la percepción del sensor. Ante lo cual se debió encontrar una solución para que las mediciones pudieran ser siempre en las mismas condiciones. Por lo cual se continuó haciendo pruebas en distintas condiciones de iluminación, tomando tres distancias para establecer igualdad entre las pruebas, esas distancias son: 2009.0, 1571.0 y 1086.0, todas en milímetros medidos con el mismo Kinect.

- Bajo iluminación artificial: Se colocó el sensor en una habitación sin ventanas, y con luminarias fluorescentes, donde se realizaron tres pruebas con las distancias antes mencionadas.

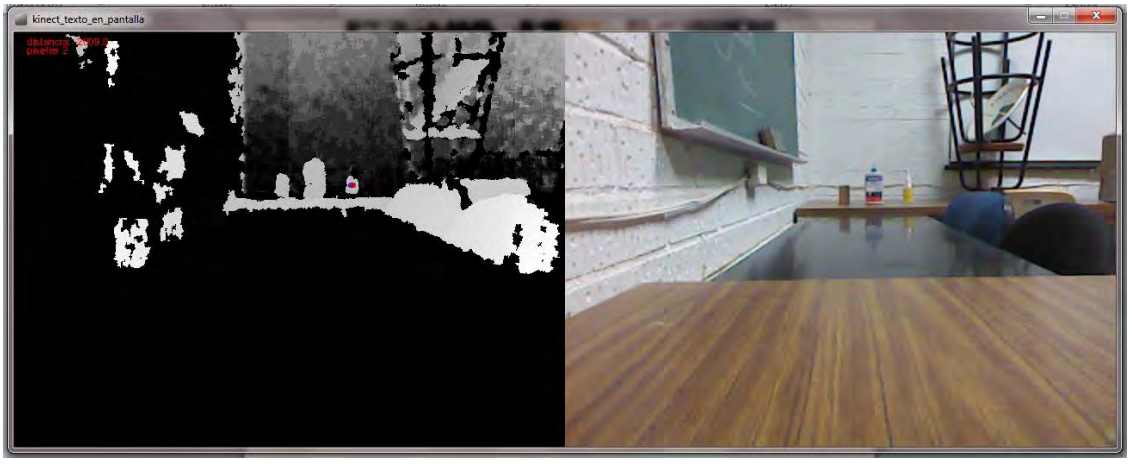


Figura 2.23. Captura de pantalla del sensor trabajando bajo luminarias fluorescentes.

En la figura 2.23 se muestra la imagen generada con una distancia igual a 2009.0 mm medidos con el Kinect, y en la que las tres muestras se pueden observar claramente, tanto en la cámara RGB como en la de infrarrojos. Sin embargo el programa no funciona adecuadamente ya que varían demasiado la medición de pixeles y la distancia respecto al Kinect.

Figura 2.24. Captura de pantalla del sensor trabajando bajo luminarias fluorescentes.

En la figura 2.24 se muestra la imagen generada con una distancia igual a 1571.0 mm medidos con el Kinect, con esta distancia el programa si trabaja a decuadamente, p ues l as mediciones t ienen poc a v ariación entre ellas.

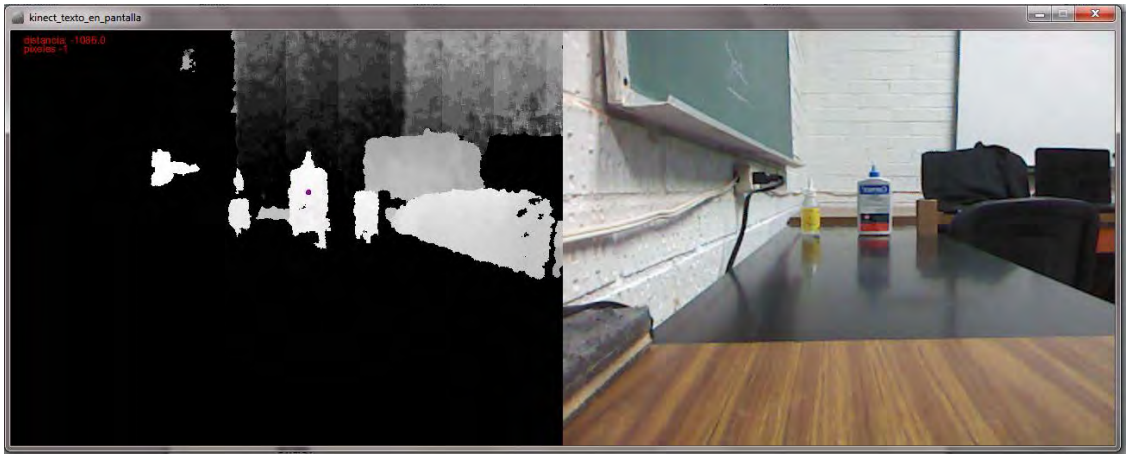


Figura 2.25. Captura de pantalla del sensor trabajando bajo luminarias fluorescentes.

Se disminuyó la distancia a 1086.0 mm, al igual que en las figuras anteriores se puede apreciar en la figura 2.25 que las muestras son visibles de forma clara, y las mediciones siguen teniendo poca variación entre ellas. Por lo cual se deduce que la luz generada por las lámparas fluorescentes no afecta al sensor.

- En ausencia de luz: Se colocó el sensor en una habitación cerrada y sin ningún tipo de iluminación, así como ninguna ventana, al correr el programa no se nota ningún cambio en la cámara de infrarrojos, aunque en la cámara RGB, se pierde por completo la imagen, esto se puede apreciar en la figura 2.26.

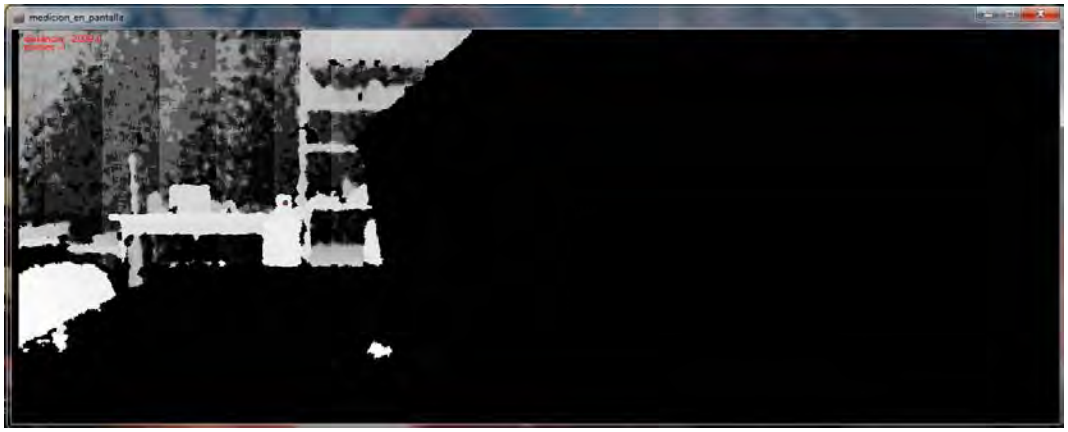


Figura 2.26. Captura de pantalla del sensor trabajando en ausencia de luz.

A pesar de la ausencia de luz el programa tiene dificultades para realizar las mediciones a 2009.0mm, aunque la detección de profundidad se mantiene, pero la medición de sección transversal falla de manera repetida, pues en la mayoría de las ocasiones ni siquiera completa la medición. Ante lo cual se procede a disminuir la distancia al sensor nuevamente.

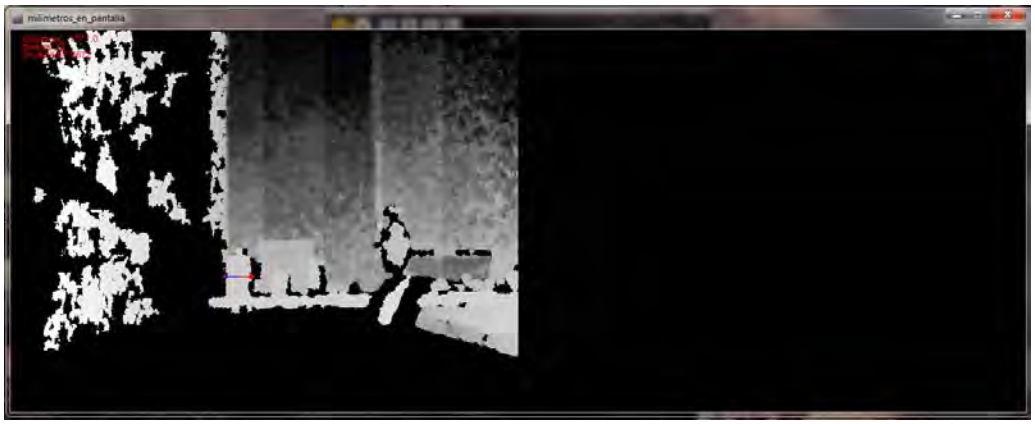


Figura 2.27. Captura de pantalla del sensor trabajando en ausencia de luz.

Al disminuir la distancia a 1571.0mm, las mediciones se vuelven más constantes, y el programa detecta los bordes en la mayoría de las ocasiones, al reducir la distancia a 1086.0mm no se nota gran diferencia con la prueba anterior pues el programa continua trabajando de manera adecuada y los errores son muy pocos.

Figura 2.28. Captura de pantalla del sensor trabajando en ausencia de luz.

Por lo cual se decidió continuar las pruebas y mediciones a esta distancia; en cuanto al tipo de iluminación bajo la cual se realizaron las pruebas, se eligió realizar las mismas bajo luz artificial, pues la luz solar si afecta al desempeño del sensor, y en condiciones de obscuridad no hay una diferencia notable con respecto a la iluminación artificial, así que por comodidad, las pruebas se llevaron a cabo con esas condiciones.

Ensayos de tensión y compresión

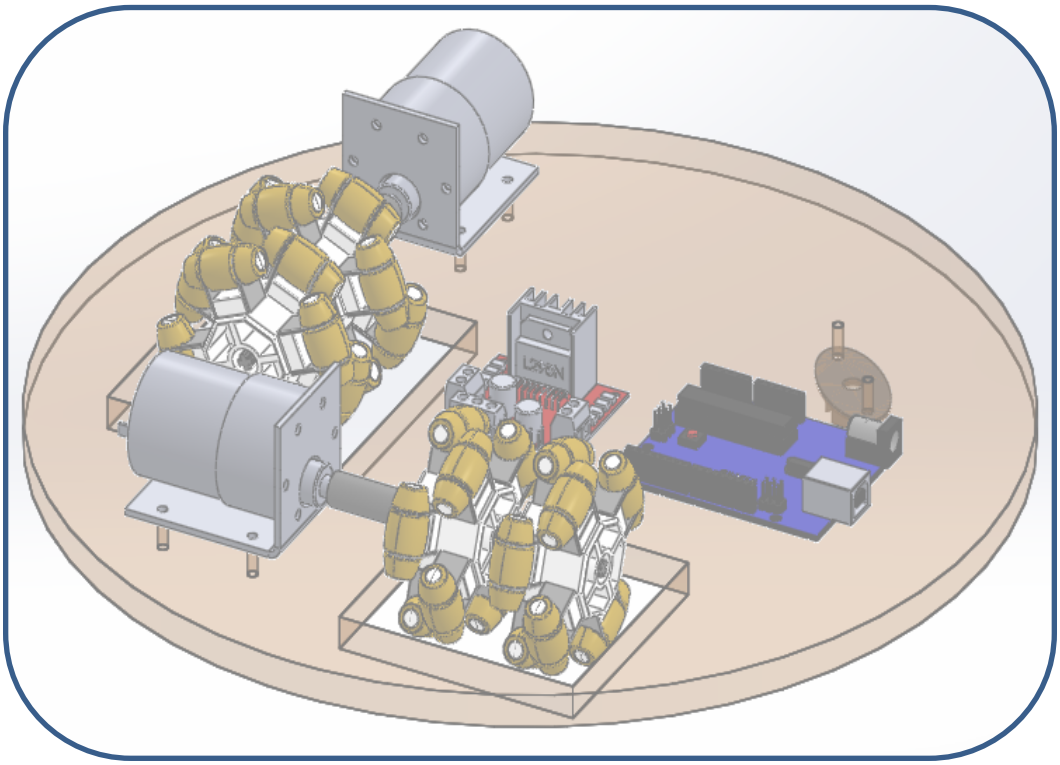
La resistencia de un material depende de su capacidad para soportar una carga excesiva sin presentar deformación o falla. Esta propiedad es inherente al propio material y debe determinarse mediante experimentación. Una de las pruebas más importantes a este respecto es el ensayo de tensión o compresión. Mismo que se utiliza principalmente para determinar la relación entre el esfuerzo y la deformación en muchos materiales de ingeniería.

Para realizar un ensayo de tensión o compresión, se fabrica una probeta con forma y tamaño estándar definidos en base a una norma; que depende del material a estudiar. La probeta puede tener una sección transversal circular, rectangular o plana, siendo constante con extremos más grandes, con el objetivo de que la falla no se produzca en las empuñaduras. Antes de realizar el ensayo, con la ayuda de un punzón se hacen dos pequeñas marcas sobre la longitud uniforme de la probeta. Se hacen mediciones tanto del área de la sección transversal inicial de la probeta, como de la longitud calibrada entre las marcas. Se utiliza una máquina de ensayos, para estirar la probeta a una velocidad lenta y constante hasta que esta falla. La máquina está diseñada para leer la carga que se requiere para mantener este estiramiento uniforme.

Durante la prueba se registran los datos de la carga aplicada a intervalos frecuentes, esta información se lee en la pantalla de la máquina. Además, se mide el alargamiento "$\delta=L-L_i$", entre las marcas hechas en la probeta utilizando un calibrador o bien un dispositivo óptico o mecánico llamado extensómetro. Este valor de δ (delta) se utiliza para calcular la deformación normal promedio en la probeta.

Sin embargo, la sección transversal de la probeta también sufre cambios, al aumentar o disminuir según sea el ensayo realizado, aumentando si es de compresión y disminuyendo si es de tensión.

CAPÍTULO III SOFTWARE EMPLEADO



“Siempre parece imposible
hasta que se hace”
(Nelson Mandela)

Processing

Figura 3.1. Logo de Processing

Es un entorno de desarrollo de código abierto basado en java, que sirve como medio para la ens eñanza y pr oducción de proyectos multimedia e interactivos de diseño digital.

Es un s oftware l ibre, di sponible p ara las plataformas Mac OS, Linux y Windows, fue desarrollado par tir de l 2001 e n el M IT por Casey Reas y Ben Fry.

Es un entorno muy fácil de utilizar y sus primeras pruebas se pueden hacer en tan sólo unos minutos, permite 3 formas de programar: básica, estructurada y orientada a objetos.

Programación Básica

Se entiende como programación básica cuando se usan comando simples tales como son los de suma-resta, multiplicación y división, en pocas palabras, aritmética básica, así como comparadores (<,>, <=,>=, etc.).

Programación Estructurada

La programación estructurada en una teoría de programación que consiste en construir programas de fácil comprensión. Se basa en una metodología de desarrollo llamada “Refinamiento sucesivo”, el cual, plantea una operación, como un todo, dividiéndola en varios segmentos más sencillos, mismos que se resuelven de forma individual, cuando están terminados todos los segmentos del programa, se procede a unificar cada parte de la operación principal, en un solo programa. Si se realizó correctamente, la integración deberá ser sencilla y no

presentar problemas, aunque de ser así, será rápidamente detectable para su corrección.

La representación gráfica de la programación estructurada se realiza a través de diagramas de flujo, el cual representa el programa con sus entradas, procesos y salidas. Para la solución de un problema, se inicia considerando las funciones que debe cumplir el programa en general, y después se va desmembrando en subfunciones más pequeñas, entonces se programan estas sub funciones, de esta manera siempre se pueden construir nuevas unidades, insertando el nombre del módulo donde corresponda y desarrollándola aparte.

La modificación por módulos o unidades es más fácil, y se pueden referenciar cuantas veces se requiera; un programa tiene un diseño estructurado si cumple con dos condiciones:

1. El teorema de Estructura
2. Está debidamente documentado

El teorema de estructura dice: “un programa cumple el teorema de estructura si y solo si es propio y contiene únicamente las tres estructuras básicas de control” que son la secuencial, la alternativa y la repetitiva; un programa es propio si y solo si cumple: con tener un solo punto de entrada y solo un punto de salida, y si entre dos puntos de control del programa exista al menos un camino.

Estructuras básicas

Secuencial

Indica que las instrucciones de un programa s ejecutan después de la otra, en el mismo orden en el cual aparecen en el programa. Se representa físicamente como un recuadro después de otro, ambos con una sola entrada y una única salida, como se muestra en la figura 3.2.

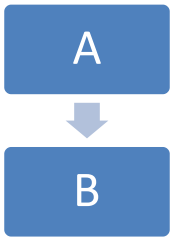


Figura 3.2. Estructura secuencial

Los recuadros A y B pueden ser definidos para ejecutar desde una simple instrucción hasta un módulo o programa completo, siempre y cuando sean apropiados.

Selectiva

También es conocida como estructura "falso verdadero", plantea la selección entre dos alternativas con base en el resultado de la evaluación de una condición; equivale a la instrucción " IF" de los lenguajes de programación y su representación gráfica se muestra en la imagen 3.3.

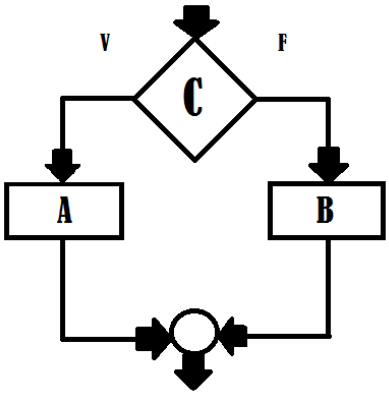


Figura 3.3. Estructura selectiva

En el diagrama de flujo mostrado en la imagen anterior, "C" es una condición que se evalúa; A es la acción que se ejecuta cuando la evaluación de esta condición resulta verdadera y B es la acción que se ejecuta en caso contrario.

La estructura también tiene una sola entrada y una salida; y las funciones A y B pueden ser cualquier estructura o conjunto de estructuras.

Repetitiva

Corresponde a la ejecución repetida de una instrucción mientras que se cumple una determinada condición. Corresponde a la instrucción "while" en lenguajes de programación. El diagrama de flujo de la figura 3.4 muestra gráficamente este tipo de estructura.

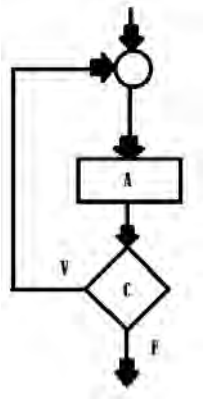


Figura 3.4. Estructura repetitiva

Aquí el bloque "A" se ejecuta repetidamente, mientras que la condición "C" se cumpla o sea verdadera, igual que en las estructuras anteriores "A" puede ser cualquier estructura, módulo o programa.

Programación Orientada a objetos

La programación orientada a objetos es una propuesta de programación que usa objetos en sus interacciones, para diseñar aplicaciones y programas informáticos.

Los objetos son entidades que tienen un determinado "estado", "comportamiento" e "identidad".

- El estado está compuesto de datos o información; serán uno o varios atributos a los que se habrán asignado unos valores concretos.
- El comportamiento está definido por los métodos o mensajes a los que sabe responder dicho objeto, es decir, que operaciones se pueden realizar con él.
- La identidad es una propiedad de un objeto que lo diferencia del resto; dicho con otras palabras, es su identificador, tal como se hace con variables o constantes.

"Objeto" se puede definir como un conjunto complejo de datos y programas que poseen estructura y forman parte de una organización.

El objeto se compone de 3 partes:

- Relaciones: Permiten que el objeto se inserte en la organización y están formadas por punteros hacia otros objetos.
- Propiedades: Distinguen un objeto determinado de los restantes que forman parte de la misma organización y tiene valores que dependen de la propiedad de que se trate. Las propiedades de un objeto pueden ser heredadas a otros que sean consecuentes o descendientes en la organización.
- Métodos: Son las operaciones que pueden realizarse sobre el objeto, que normalmente estarán incorporados en forma de programas o código que el objeto es capaz de ejecutar y que también pone disposición de sus descendientes.

Cada objeto, es una estructura compleja en cuyo interior hay datos y/o programas, todos ellos relacionados entre sí, como si estuvieran encerrados en una capsula, por eso esta propiedad se llama encapsulamiento, y es una característica fundamental de la programación orientada a objetos.

PDE (Portable Development Environment)

Processing es un ambiente de desarrollo portátil (PDE, por sus siglas en inglés), por ello cuando se guarda una aplicación o bosquejo (sketch), se genera una carpeta con el nombre que le designemos, y el directorio de la aplicación se almacena dentro de ésta, con el mismo nombre y la extensión ".pde".

Los nombres de los bosquejos deben ser de máximo 64 caracteres sin espacios; se pueden crear más bosquejos asociados a la misma aplicación y son usados como subrutinas o clases y para que el programa principal pueda leerlas, se deben definir en la sección de configuración principal (setup), estos bosquejos se guardan en la misma carpeta del programa principal, con la misma extensión y para abrir el proyecto se puede iniciar cualquiera de los bosquejos dentro de la carpeta.

En la figura 3.5 se muestra la ventana de "processing", y se muestra las partes principales de la misma.

Figura 3.5. Ventana de Processing

"Ejecutar" y "parar" permiten iniciar y detener la aplicación, "nuevo" permite generar un bosquejo nuevo, aljeno al actual, " Abrir" se emplea para abrir otro bosquejo previamente guardado, " Guardar" permite almacenar el código

generado, “exportar” permite crear una aplicación del bosquejo, que puede ser ejecutada en Mac OS, Linux o Windows.

El nombre del bosquejo es “Sketch_AAMMDDa” (cada letra de la palabra AAMMDDa indica una cifra, A para año, M para mes, D para día, a para indicar la cantidad de bosquejos creados en esa fecha) de manera predeterminada, y se almacena en la carpeta “sketchbook”, pero puede cambiarse tanto el nombre del bosquejo como la carpeta de destino.

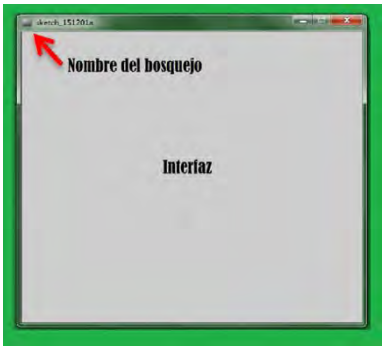


Figura 3.6. Interfaz de processing

La figura 3.6 muestra la ventana donde se muestra la interfaz del bosquejo creado con processing, el tamaño de la ventana se configura en el código, en la sección de configuraciones “setup”.

Labview

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) es un lenguaje de programación, gráfico para el diseño de sistemas de adquisición de datos, instrumentación y control. Permite diseñar interfaces mediante una consola interactiva generada por software.

Figura 3.7. Logotipo de LabVIEW

Los programas desarrollados en LabVIEW son denominados instrumentos virtuales, o VI's, (Virtual Instruments), ya que su apariencia y operación imita a instrumentos físicos pero de forma virtual, tal como osciloscopios, multímetros, etc. Estos VI's se pueden utilizar dentro de otros VI's, convirtiéndose en sub VI's. Los VI's de LabVIEW contienen dos componentes principales: el panel frontal y el diagrama de bloques.

Panel Frontal

Es la interfaz de usuario para el VI. La figura 3.8 muestra un ejemplo de panel frontal. Esta interfaz la construye el usuario, con controles e indicadores, los cuales son las terminales interactivas de entrada y salida del VI.

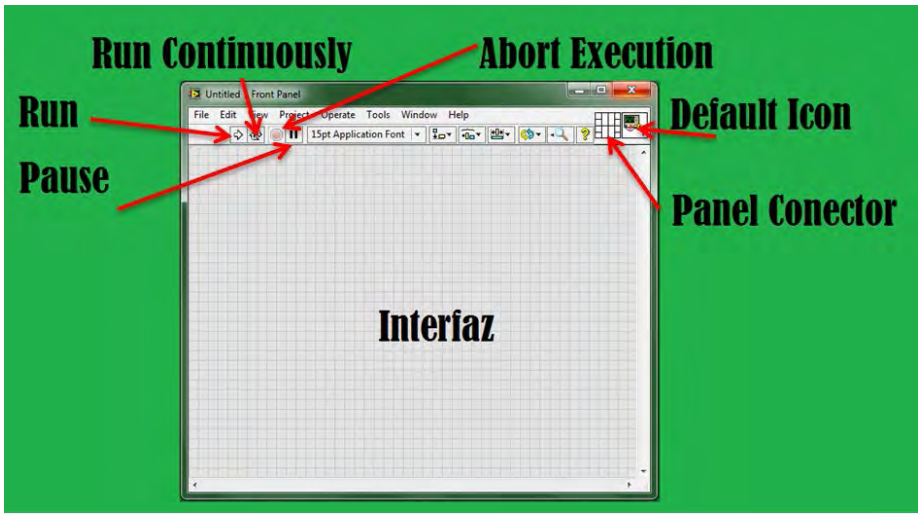


Figura 3.8. Panel Frontal de LabVIEW

El botón “Run” Permite iniciar el VI, una sola vez, hasta el punto donde el programa este elaborado. “Run Continuously” Inicia el VI de forma cíclica, por lo cual, el usuario deberá añadir al programa un botón de paro, para evitar detener el programa mientras se interactúa con algún objeto externo, pues algunos dispositivos requieren tener una secuencia de paro. “Abort Execution” permite al usuario detener el programa en cualquier momento, este botón suele usarse en caso de que el programa entre en un bucle infinito. También se muestran en la esquina superior derecha el “default icon” (icono por defecto), que es una representación gráfica del contenido o tipo de VI en el que se está trabajando, este puede contener texto e imágenes, y contiene un número que indica cuantos VI’s nuevos se han abierto desde que inicio LabVIEW.

El panel conector es un conjunto de terminales que corresponden a los controles y/o indicadores del VI en el que se está trabajando, se utiliza para usar el VI como un sub VI, y debe editarse, desde el panel frontal, pues en el diagrama de bloques no es posible.

Diagrama de bloques

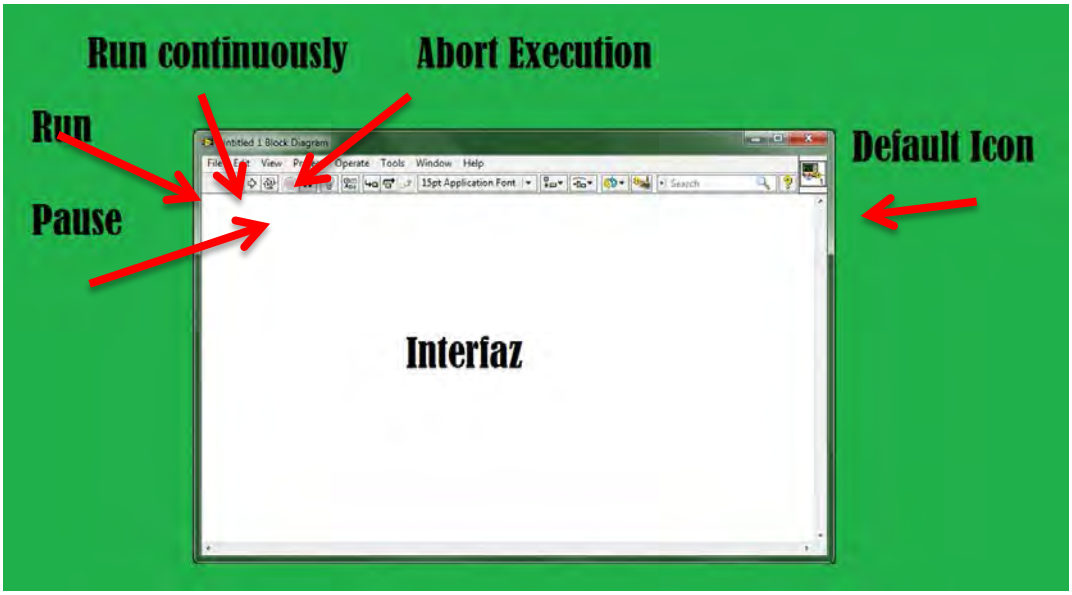


Figura 3.9. Diagrama de bloques de LabVIEW

El diagrama de bloques contiene código fuente gráfico, mismo que es utilizado para controlar los objetos del panel frontal, Estos objetos aparecen como terminales en el diagrama de bloques, a estas terminales se les agrega código en forma de representaciones graficas de funciones. Contiene los mismos controles que el panel frontal con la excepción del panel conector.

La programación en LabVIEW se basa en la asignación de iconos que representan los datos y procedimientos que se necesiten, mediante la conexión de los mismos, utilizando líneas para ello, estas líneas ayudan a identificar el tipo de datos, pues su forma y color varía dependiendo de los datos que las emplean. Esta forma de programar está orientada al flujo de datos, mismo que va de izquierda a derecha en el panel de programación y se determina por las funciones que procesan los datos empleados.

LabVIEW es ampliamente usado en el procesamiento de imagen debido a la gran flexibilidad que posee, pues el mismo hardware y software puede ser empleado para diversas aplicaciones.

En su módulo de visión existen diversas herramientas para desarrollar aplicaciones de visión artificial, como la detección de bordes, o detección de formas, esta última es de interés para el proyecto. En la figura 3.10 se muestran las herramientas disponibles para visión artificial que proporciona LabVIEW, en el kit: “vision and motion”.

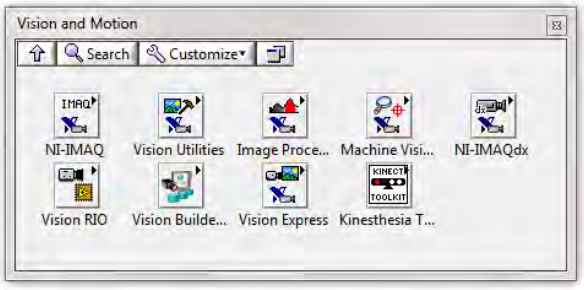


Figura 3. 10. Kit de herramientas “ Vision and motion”

Para poder utilizar el Kinect con LabVIEW, se necesita un kit de herramientas, llamado “ Kinesthesia”, con el cual se puede iniciar la cámara de profundidad de Kinect, sin embargo, al realizar pruebas con ambos kits, se notó que estos, no trabajan juntos, la capacidad de detectar objetos no funciona con la imagen de profundidad adquirida por el Kinect únicamente trabaja con cámaras RGB, y el kit kinesthesia solo puede identificar, las extremidades del cuerpo humano por sí mismo. Por lo cual se descartó este programa para su utilización en el proyecto.

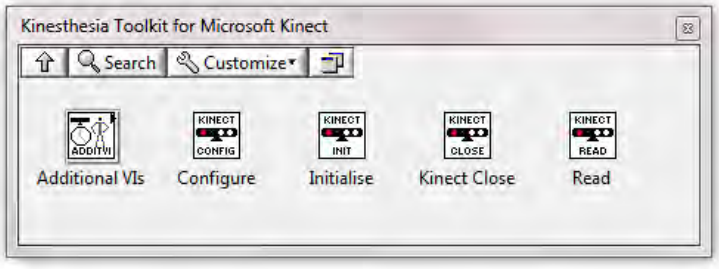
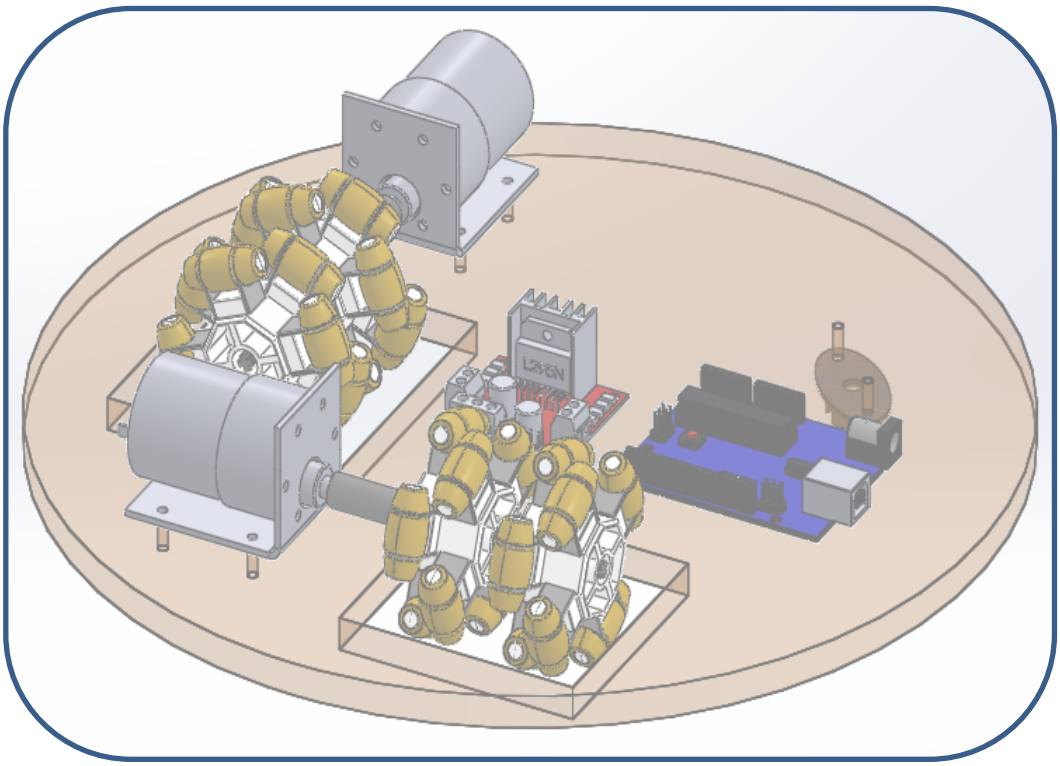


Figura 3.11. Kit de herramientas “kinesthesia”

CAPÍTULO IV PROGRAMACIÓN



“La serenidad no es estar a salvo de la tormenta,
sino estar en paz en medio de ella.”
(Thomas de Kempis)

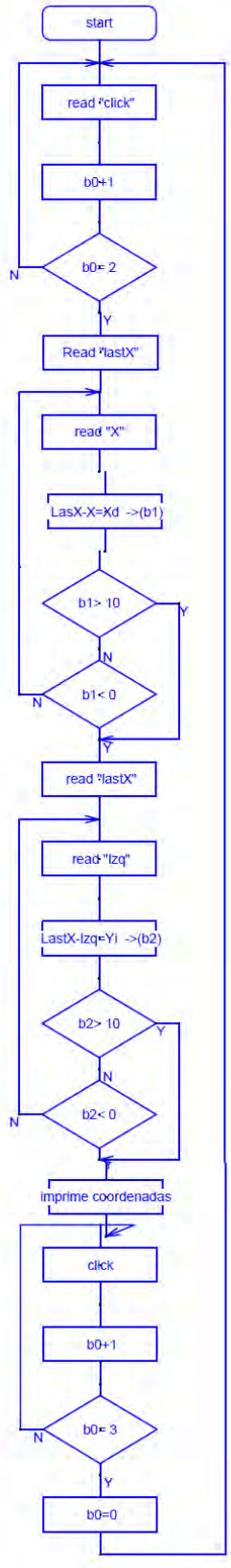


Figura 4. 1. D iagrama de f lujo de l programa

Para hacer que el sensor Kinect obtenga los valores deseados d e l a s ección t ransversal d e un ob jeto s e d iseñó un programa en “Processing”, esto lo va a realizar haciendo comparaciones d e pr ofundidad e ntre c ada punto det ectado por el sensor infrarrojo, y al haber una diferencia de profundidades mayor a 10 milímetros se dará por hecho que se enc ontró u n c ontorno de l ob jeto que s e es tá midiendo, entonces, el pr ograma r egresara al punt o i nicial y comenzara a hacer comparaciones de la misma forma pero ahora en dirección izquierda respecto al punto inicial.

En l a figura 4.1 se muestra e l di agrama d e f lujo qu e explica la lógica usada en el desarrollo del programa.

Durante l a redacción del programa s e tuvieron v arias pruebas, pu es a unque s e s iguió la lógica mostrada e n e l diagrama de flujo de la figura 4.1, el programa no fue completamente útil, al primer intento.

En la primera prueba, se utilizó como base la teoría: “si l a di ferencia de d istancia de p rofundidad de l a v ariable “clickdepth” es menor a 10 milímetros, inicia la operación “if” y l a o peración “while”, s i l a di ferencia es mayor, s e detendrán”.

En la tabla 4.2 se m uestran l os resultados de 4 intentos de esta primera prueba.

Variable	intento 1	intento 2	intento 3	intento 4
depthvalue	221895	3640	0	846
lastX	0	454	435	454
clickedDepth	767	569	567	560
lastclick	0	0	567	560
run	0	0	0	907
dif	0	0	0	560

Tabla 4.2 Valores de primera prueba

El valor de la variable "LastX" inicia donde el puntero hace click, después, con un ciclo "while", cambia su posición desplazándose sobre el eje de las "X"; la variable "run" es la medida de profundidad correspondiente al pixel que ocupe en ese momento la variable "lastX", este mismo valor aumenta de uno en uno. El valor "dif" es la diferencia de l valor "clickedDepth" menos el valor "run", esta diferencia se realizara hasta que su valor sea superior a 10.

Como se puede apreciar, los resultados que arroja el sensor no tienen relación ni coherencia entre ellos, de esto, se notó que el problema se debía a que el programa arranca con un valor inicial para todas las variables igual a cero por lo cual, se genera confusión al hacer las mediciones, y por lo tanto se descartó este programa.

En la segunda prueba, se corrigió el problema del valor inicial de las variables, solicitando al usuario generar un valor inicial a las variables que se utilizan para medir la profundidad, esto se lleva a cabo por medio de un click,

Donde se coloca el cursor y se hace click, el programa arroja valores de profundidad de los pixeles (sobre el eje X) siguientes al click, hasta que la diferencia de profundidades es superior o igual a 10. Esto sucede solo en una dirección; el programa inicia al 3er click, y reinicia (el contador de clicks) con un click más, lo que significa que se debe volver a hacer click 3 veces más para volver a ejecutar el programa.

Este programa funciona, sin embargo, sólo mide el objeto hacia un lado y tiene el problema que si la diferencia es menor a -10 sigue corriendo, por lo cual se descarta.

Al igual que en la segunda prueba, en la prueba numero 3 el programa inicia al 3er click, y se detiene hasta que la diferencia de profundidades del valor medido con el pixel anterior sean superiores o iguales a 10, sin embargo esta comparación se realiza en dos sentidos utilizando para ello un ciclo "while".

En la cuarta prueba se dibujan puntos en los pixeles medidos, lo que a su vez forma líneas, esto con el fin de saber si el programa se detiene donde se requiere, dando como resultado un error, pues el programa no se detiene donde se requiere, por lo que se tuvo que redactar otro programa.

Tomando como base la prueba número 3 se generó la prueba 5, en la cual se emplean dos ciclos "while", para hacer las mediciones, uno para el lado derecho del punto inicial y otro para el lado izquierdo, y de igual manera que en la prueba número 4, se dibujan unas series de puntos en los pixeles que se han medido, y, en este caso se generan dos líneas diferentes, una para cada ciclo "while".

Esta prueba solo funciona la primera vez que se corre el programa, por lo que se necesita volver a redactar otro programa.

En la prueba 6 los ciclos "while" se modificaron de tal forma que la diferencia a tomar en cuenta para detenerlos, debe de estar en un rango de -10 a 10 pixeles, esto se entiende como un borde del objeto y por lo tanto el punto donde se inicia la medición de la sección transversal. Se solucionó el problema del programa de correr solo la primera vez.

Por último se generó un programa que trabaja de igual manera que el de la prueba 6, con la diferencia qué se le agrego la capacidad de imprimir en la interfaz, los datos considerados más importantes, los cuales son: la distancia que hay entre el sensor y la cantidad de pixeles que hay entre los bordes de la

muestra. Esto con el objetivo de facilitar la captura de esos mismos datos, pues anteriormente se tenían que buscar en el área de debugging (área destinada a depurar el programa y donde processing imprime los datos que se soliciten, además de posibles errores) lo que tomaba mucho tiempo. Posteriormente se comenzó a registrar las pruebas de medición.

Se realizaron 8 mediciones de 3 objetos diferentes para observar la precisión del Kinect. Los resultados de estas mediciones se registraron en la tabla 4.3. Los objetos utilizados fueron: dos recipientes plástico, y un trozo de madera (bote, silicón, y madera son los nombres con los que se denominaron los objetos para el llenado de la tabla 4.3).

	med 1	med 2	med 3	med 4	med 5	med 6	med 7	med 8
bote	45	24	32	48	36	47	45	46
madera	31	31	32	31	32	31	32	31
silicón	26	28	28	27	31	27	26	27

Tabla 4.3 Valores de prueba de tres muestras

Todos estos valores se obtuvieron a una distancia de 1086.0 mm medidos con el Kinect y las medidas obtenidas se muestran en pixeles.

Como puede observarse la diferencia entre cada medición es muy pequeña lo que se interpreta como un aumento en la precisión de las mediciones llevadas a cabo con este programa, en contraste con la primera prueba que se hizo.

Basado en las características teóricas del sensor, sabemos que tiene una resolución espacial de 3mm estando el objeto a 2000mm del sensor.

Esto nos permite hacer una equivalencia en las dimensiones obtenidas.

Si cada pixel equivale a una distancia de 3mm estando el objeto a 2000mm, tenemos que:

Sí: $3 \rightarrow 2000$

Entonces: $x \rightarrow 1086$

$$\frac{(3 * 1086)}{2000} = 1.629$$

Este resultado es el valor teórico equivalente en milímetros de cada pixel a una distancia de 1086.0 milímetros del Kinect.

Lo que nos genera la tabla 4.4, ahora con valores métricos, resultado de multiplicar cada valor de la tabla anterior por el valor obtenido en el cálculo de la equivalencia pixel-milímetro.

*1.629	med 1	med 2	med 3	med 4	med 5	med 6	med 7	med 8
bote	73.305	39.096	52.128	78.192	58.644	76.563	73.305	74.934
madera	50.499	50.499	52.128	50.499	52.128	50.499	52.128	50.499
silicón	42.354	45.612	45.612	43.983	50.499	43.983	42.354	43.983

Tabla 4.4 valores en milímetros

Los valores reales en milímetros de los objetos medidos son:

Objeto	Medida (mm)
bote	81.077
madera	45.999
silicón	38.398

Tabla 4.5 valores en milímetros
(medidos con calibrador vernier)

Como se puede apreciar, la diferencia entre mediciones es bastante aunque constante, sin embargo cabe aclarar que esta comparación se realizó con los datos teóricos del Kinect, se pretendía repetir la misma comparación pero utilizando una medición práctica, estableciendo como base la medida a 2000 milímetros de distancia medidos con el Kinect, para establecer una medida práctica base similar a la teórica y repitiendo todo el proceso anterior.

Figura 4.6 Trozo de madera

La medida de base práctica que se pretendía utilizar era de 2009.0mm porque no fue posible obtener los 2000mm exactos, sin embargo debido a que el programa no trabaja adecuadamente a esa distancia, se redujo a 1086.0mm pues con esta separación el programa tuvo un mejor desempeño. Se hizo una comparativa parecida a la que se realizó con los datos teóricos, pero en esta ocasión, se usaron las dimensiones del objeto medido por medios físicos, (utilizando un calibrador vernier), dicho objeto es el trozo de madera rectangular cuya sección transversal, mide 45.999 milímetros, este objeto se muestra en la figura 4.6; después se midió con el Kinect a la distancia establecida y a partir de ello, se determinó una medida base.

Al medir el trozo de madera con el Kinect obtuvimos que este mide 31.375 pixeles en promedio (se utiliza un promedio debido a que el sensor no arroja los mismos valores en cada medición), con estos valores podemos establecer una relación entre la cantidad de pixeles que mide el sensor y la medida en milímetros del objeto, con lo cual obtenemos una base para hacer mediciones de distintos objetos.

Si el objeto mide 45.999 milímetros y el programa imprime 31.375 pixeles, estando el objeto a 1086.0mm, dividimos la cantidad de milímetros entre la cantidad de pixeles para obtener la cantidad de milímetros que medimos en un pixel a la distancia mencionada, entonces tenemos que:

$$45.999\ milímetros \rightarrow 31.375\ pixeles$$

Entonces: $$\frac{45.999\ milimetros}{31.375\ pixeles} = 1.4661\ \frac{milimetros}{pixel}$$

Con estos datos se redactó otro programa que imprime en la interfaz el valor en milímetros del objeto medido, sin importar la distancia a la que se encuentre. Tomando como base el valor obtenido en el cálculo anterior, si cada pixel equivale a una distancia de 1.4661 milímetros estando el objeto a 1086.0mm, tenemos que:

Sí: $$1.4661 \rightarrow 1086$$

Entonces: $$x \rightarrow distancia$$

$$\frac{(1.4661 * distancia)}{1086} = x$$

$$\frac{\frac{mm}{pixel} * mm}{mm} = \frac{mm}{pixel}$$

Siendo $\boldsymbol{x}$ la equivalencia en milímetros de cada pixel dependiendo de la distancia a la que se encuentre, este valor se multiplica por la cantidad de pixeles

medidos y esto nos da el valor en milímetros del objeto, mismo que se imprime en la interfaz.

$$x * pixeles = medida\ en\ milimetros$$

$$\frac{mm}{pixel} * pixel = mm$$

Con estos datos se puede llevar un registro práctico de los valores que se obtienen con el sensor, mismos que se muestran en la tabla 4.7.

	med 1	med 2	med 3	med 4	med 5	med 6	med 7	med 8
madera	45.4491	45.4491	46.9152	45.4491	46.9152	45.4491	46.9152	45.4491

Tabla 4. 7. Valores en milímetros
(medidos con el sensor Kinect)

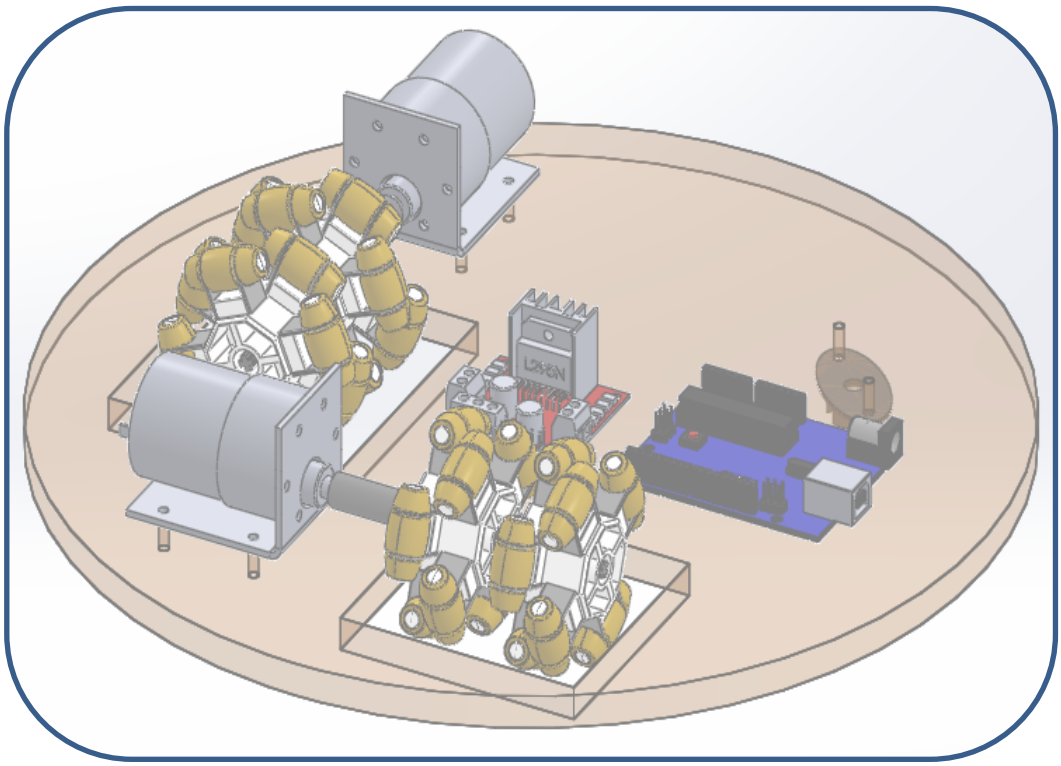
La medida de la sección transversal del trozo de madera es de 45.999 milímetros, se puede notar que el valor que más se acerca es de 45.4491, y la media de los valores es:

$$45.4491 + 45.4491 + 46.9152 + 45.4491 + 46.9152 + 45.4491 + 46.9152 + 45.4491 = 367.9911$$

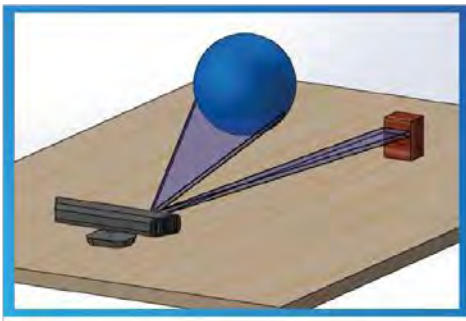
$$\frac{367.9911}{8} = 45.9988$$

El valor real es de 45.999, y el valor promedio de las mediciones realizadas con el Kinect es de 45.9988; como se puede apreciar, la diferencia es mínima, apenas .0002mm. Por lo que se considera confiable.

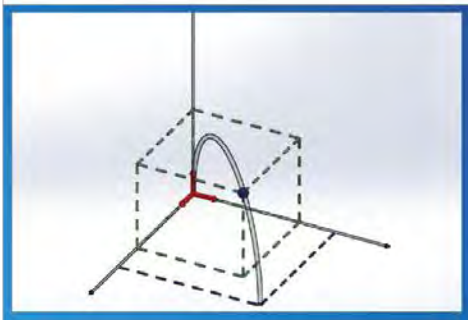
CAPÍTULO V CONCEPTO BASE

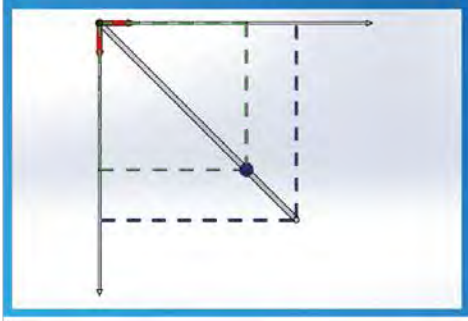


“No escalas la montaña para que todo el mundo pueda verte,
sino para que TÚ puedas ver el mundo”
(Anónimo)



El programa desarrollado puede detectar los bordes de un objeto cualquiera y medir la distancia entre ellos. Lo que es una base para la detección de objetos.



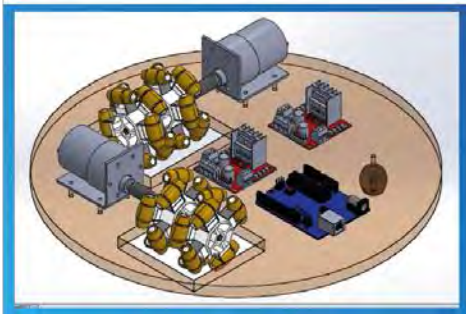


La trayectoria de un objeto cualquiera lanzado al aire, genera un movimiento parabólico.

Las componentes de la trayectoria que genera el movimiento de dicho objeto se pueden descomponer para obtener una proyección en un plano.

La proyección obtenida genera una posición final del trayecto, misma que se puede utilizar para posicionar el robot.

El robot tendrá un arreglo de ruedas omnidireccionales, para generar un movimiento con relación a las coordenadas en el plano, utilizando estas últimas para dirigir un motor para cada eje del plano.



La propuesta del robot es como la mostrada en la figura.

La primera parte de la detección de objetos se basa en la detección de bordes, pues a partir de ellos se puede detectar la presencia de un objeto, al brindar al programa la capacidad de distinguir un objeto de otro utilizando sus límites. Posterior a la detección del objeto debe formularse un algoritmo para detectar la trayectoria del objeto detectado mientras esta en movimiento, esto puede llevarse a cabo mediante el registro de la posición en el espacio y la descomposición en componentes.

El movimiento que se va a analizar será un tiro parabólico por lo cual los datos que se registran se analizaran siempre con la misma metodología, cabe mencionar que los datos que se registren deben ser pocos, para permitir a la computadora realizar cálculos rápidamente, y que la trayectoria sea definida para poder predecir el lugar en el que el objeto detectado caerá y de esta forma traducir el movimiento parabólico en un par de coordenadas para posicionar el robot, dándole suficiente tiempo para desplazarse e interceptar el objeto antes de que toque el piso.

Los datos de detección y posición del objeto se realizaran utilizando processing y Kinect; en este mismo programa se realizaran los cálculos y el control del robot, ya que permite la interacción con la placa de desarrollo Arduino, aunque para esto debe cargarse un programa desde el IDE propio de Arduino en el microcontrolador, y posteriormente y a se pueden cargar programas desde processing.

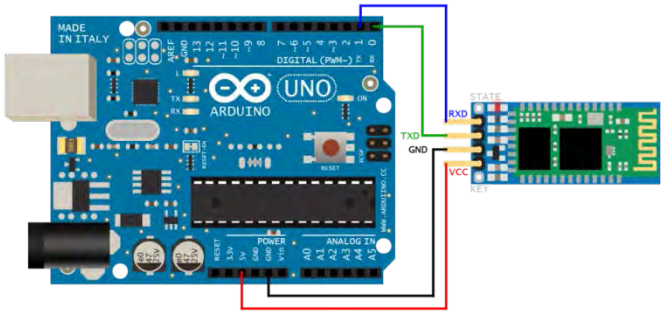


Figura 5.1. Módulo bluetooth HC-06 conectado a placa Arduino

Después de realizados los cálculos, los datos obtenidos para posicionar el robot se enviaran por medio de una conexión bluetooth desde la computadora

hasta un módulo bluetooth HC-06 (figura 5.1) conectado la placa Arduino, misma que servirá para controlar los motores del robot.

El control de los motores se llevara a cabo con ayuda de un dispositivo controlador de motores L298N (figura 5.2) igualmente conectado a la placa Arduino.

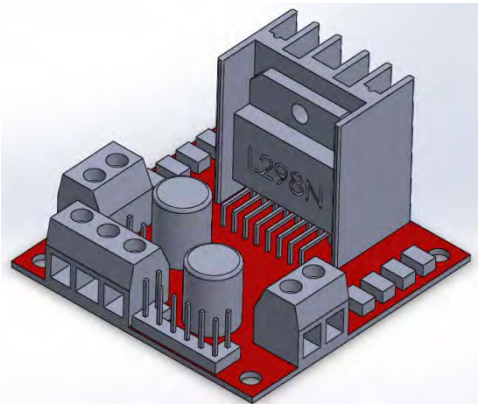


Figura 5.2. Controlador de motores L298N

El controlador L298N es un dispositivo que permite controlar el sentido de giro y la velocidad de motores de corriente directa, con una corriente de hasta 2A, es necesario utilizarlo pues la placa Arduino no soportaría la corriente requerida por los motores por si sola; este dispositivo tiene la capacidad para controlar dos motores, mismos que son los que se propone utilizar en conjunto a las ruedas omnidireccionales para el desplazamiento del robot.

Figura 5.3. Motor Pololu.

Se propone usar un par de motores pololu de 12V y 11000 RPM, pues se necesita la mayor velocidad posible para que el robot tarde lo menos posible en llegar al lugar que se haya calculado, y, esos son los que ofrecen la velocidad más alta, aunque es necesario que posea un juego de engranes reductor de velocidad, para dejar 500 RPM y un torque de 5 kg-cm, este reductor se requiere pues de no contar con él, cada motor genera tan solo 360g-cm de torque lo que resulta muy poco, tomando en cuenta que el peso de cada uno es de 190g lo que significa que se ocuparía casi la mitad del torque generado para mover tan solo el mismo motor, sin contar los demás componentes y el otro motor, además del robot y la fricción de las ruedas.

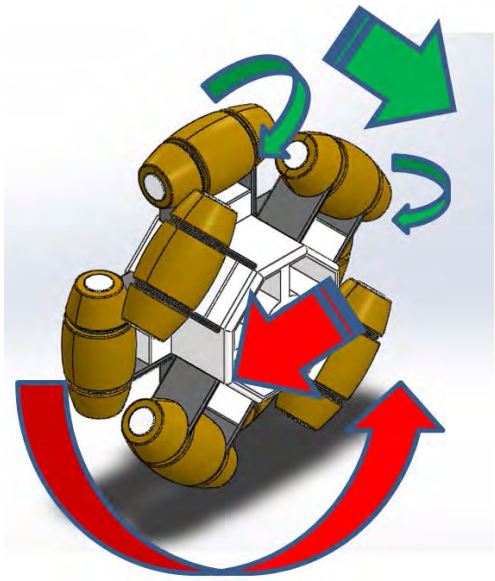


Figura 5.4. La configuración de giro de las ruedas omnidireccionales genera un movimiento distinto.

Las ruedas omnidireccionales giran hacia delante como las ruedas normales, pero tienen la capacidad de deslizarse hacia los lados con muy poca fricción, ya que poseen una serie de rodillos, los cuales están encargados de disminuir esa fricción, lo que nos brinda una segunda dirección de desplazamiento. En la figura 5.4 se muestran los movimienos que pueden generar con la rueda omnidireccional.

La configuración de giro de las ruedas omnidireccionales generara la posición del robot en un plano, por ejemplo si es solo un motor el que se activa el robot tendrá movimiento solamente sobre un eje del plano, como se muestra en la figura 5.5 (a y b), y un movimiento coordinado de los dos motores proporciona un movimiento en diagonal, o dicho de otra forma, en ambos ejes del plano de manera simultánea (figura 5.5-c), el tipo de movimiento que se requerirá, se determinara y obtendrá a partir de las coordenadas obtenidas del tiro parabólico, para que el robot se pueda desplazar al lugar donde el objeto caerá, se pretende generar un movimiento parecido al de un robot cartesiano que se desplaza sobre rieles, aunque en este

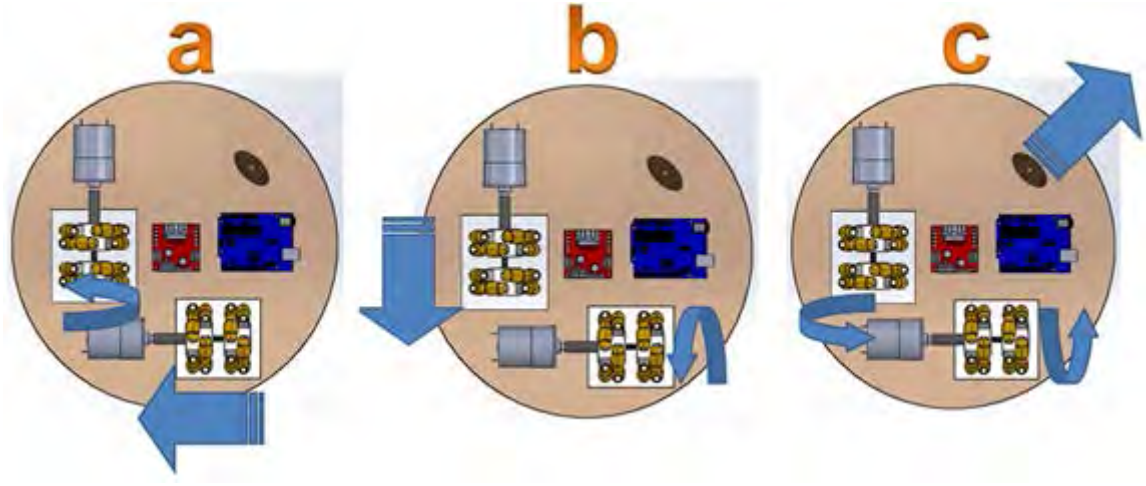


Figura 5.5. Distintas configuraciones de giro de las ruedas omnidireccionales.

caso son las ruedas omnidireccionales quienes dirigirán al robot.

Debido a la posición de los componentes sugerida en este proyecto, se utilizara una rueda loca para que el robot no arrastre sobre el piso, existe la posibilidad de que la fricción del piso con la rueda loca debida a la posición de la misma, mostrada en la figura 5.5 (elipse color café), pudiera alterar el desplazamiento del robot, pero son necesarias pruebas físicas para descartar esta posible desventaja.

CONCLUSIONES

Se desarrolló un programa que utiliza la cámara de infrarrojos del sensor Kinect, muestra la imagen de profundidad en una interfaz de "processing", y permite medir la sección transversal de un objeto cualquiera.

Aunque las mediciones que arroja el sensor de manera individual no son precisas, al hacer varias mediciones, en este caso 8 y obtener una media de ellas, se pudo apreciar que la precisión de dicha media es superior a la de las mediciones individuales.

Después de las pruebas realizadas se notó que la diferencia entre medir un objeto por medios convencionales y utilizar el sensor Kinect es mínima, pues aunque cada medición sufre una variación notable de algunos pocos milímetros con respecto al valor real medido con un calibrador vernier, la media de 8 mediciones realizadas con el sensor Kinect, arroja un valor muy parecido al real, con una diferencia mínima (.0002mm).

Otro aspecto importante a mencionar es el hecho de que la precisión del Kinect depende de la cantidad de luz y del material que se está utilizando, pues como se pudo observar con el trozo de madera las mediciones fueron constantes sin importar la iluminación, sin embargo, con otras dos muestras no fue así, siendo que las mediciones del bote fueran más precisas cuando había luz que cuando se hicieron en obscuridad, y al contrario, con el silicón, la precisión aumentó cuando se apagó la luz, las mediciones de las otras dos muestras se registraron en tablas mismas que se muestran en los anexos.

Aunque la precisión aumento o disminuyo dependiendo de las condiciones de luz, cabe mencionar que es más importante la forma de la muestra, pues la forma del trozo de madera, es un prisma rectangular con bordes rectos, el bote tiene bordes curvos además de una concavidad en la parte medida, y el silicón es

un cilindro, por lo cual, se determina que los problemas del programa para medir de manera precisa esos objetos se debe principalmente a la forma y al material.

A pesar de estas limitantes el sistema se puede considerar apto para la implementación en los ensayos, bajo la condición de que la probeta de ensayo tenga bordes rectos, pues con esa condición trabaja adecuadamente.

Debido a los problemas para realizar mediciones de los objetos restantes se omitieron las anotaciones en este trabajo.

La detección de los bordes que se desarrolló se puede emplear como base para detectar objetos, y por ello, es una base para la visión artificial del robot.

BIBLIOGRAFÍA

1. Russell C . H ibbeler MECÁNICA D E M ATERIALES Ed. Pe arson México 2011. 880 pp.
2. “Visión Artificial”. 2015, de Wikipedia Sitio web: https://es.wikipedia.org/wiki/Visi%C3%B3n_artificial
3. “5 U sos s orprendentes de K inect f uera d e l os v ideojuegos”. 2 011, d e TICbeat. S itio web: http://www.ticbeat.com/innovacion/usos-sorprendentes-kinect-fuera-videojuegos/
4. David López. (2013). Software y características del Sensor Kinect. 2015, de Prezi S itio w eb: https://prezi.com/t0qzq6s8okyk/software-y-caracteristicas-del-sensor-kinect/
5. Córdova Luc ero, F . (2012). Detección de r obo/abandono de o bjetos en interiores ut ilizando c ámaras d e pr ofundidad. Tesis de G rado. U niversidad Autónoma de Madrid, Madrid, España.
6. Agustín Salaberry. (2012) Los 10 mejores hacks de Kinect. 2015, de hipertextual. S itio w eb: http://hipertextual.com/2012/01/los-10-mejores-hacks-de-kinect
7. JJ Velasco. (2012). Inteligencia artificial: 5 robots diseñados para ayudarnos en nuestro día a día. 2015, de Hipertextual. Sitio web: http://hipertextual.com/2012/02/inteligencia-artificial-5-robots-disenados-para-ayudarnos-en-nuestro-dia-a-dia
8. Antonio D elgado. (2011) Los ot ros us os d e K inect. 20 15, d e E roski consumer Sitio web: http://www.consumer.es/web/es/tecnologia/hardware/2011/01/18/198227.php /web/es/tecnologia/hardware/2011/01/18/198227.php
9. Daniel E scandell (2014). Usan Kinect p ara desarrollar t ecnología topográfica. 2 015, de V andal. S itio web: http://www.vandal.net/noticia/1350656011/usan-kinect-para-desarrollar-tecnologia-topografica/

10. Profundidad de C ampo (2015). de T hewebfoto.com S itio w eb: http://www.thewebfoto.com/2-hacer-fotos/211-profundidad-de-campo
11. Ulrich D ruzella (2010). V isión ar tificial un a t ecnología i ndustrial. 2015. de Visión Art ificial Si ck Esp aña. Si tio Web: http://www.jcee.upc.edu/JCEE2010/pdf_ponencies/PDFs/09_12_10/Vision%20Artificial%20_UNI%20TERRASSA%202010.pdf
12. 2012. Aplicación práctica de la visión artificial en el control de procesos industriales. 201 5 Sitio web : http://visionartificial.fpcat.cat/wp-content/uploads/UD_1_didac_Conceptos_previos.pdf

ANEXOS

Tabla 8 mediciones a 2009mm de madera

	med 1	med 2	med 3	med 4	med 5	med 6	med 7	med 8
luz	13	10	12	2	9	2	13	2
obscuro	12	14	13	10	17	14	12	15

Tabla 8 mediciones a 1571mm de madera

	med 1	med 2	med 3	med 4	med 5	med 6	med 7	med 8
luz	22	22	21	22	22	23	22	23
obscuro	23	22	22	22	22	21	22	24

Tabla 8 mediciones a 1086mm de madera

	med 1	med 2	med 3	med 4	med 5	med 6	med 7	med 8
luz	31	31	32	31	32	31	32	31
obscuro	31	31	31	32	32	31	31	32

Tabla 8 mediciones a 2009mm del bote

	med 1	med 2	med 3	med 4	med 5	med 6	med 7	med 8
luz	22	2	2	14	2	2	2	8
obscuro	23	24	24	20	23	2	22	24

Tabla 8 mediciones a 1571mm del bote

	med 1	med 2	med 3	med 4	med 5	med 6	med 7	med 8
luz	32	35	34	35	35	35	34	34
obscuro	34	33	33	33	33	34	33	33

Tabla 8 mediciones a 1086mm del bote

	med 1	med 2	med 3	med 4	med 5	med 6	med 7	med 8
luz	45	24	32	48	36	47	45	46
obscuro	28	28	27	27	28	28	28	27

Tabla 8 mediciones a 2009mm del silicón

	med 1	med 2	med 3	med 4	med 5	med 6	med 7	med 8
luz	6	7	7	10	2	8	9	7
obscuro	10	6	7	8	9	8	12	14

Tabla 8 mediciones a 1571mm del silicón

	med 1	med 2	med 3	med 4	med 5	med 6	med 7	med 8
luz	17	22	21	20	20	21	20	21
obscuro	19	18	19	19	19	19	18	18

Tabla 8 mediciones a 1086mm del silicón

	med 1	med 2	med 3	med 4	med 5	med 6	med 7	med 8
luz	26	28	28	27	31	27	26	27
obscuro	45	45	44	46	45	45	45	44

“NUNCA TE DESANIMES,
PUES HASTA UNA PATADA EN EL TRASERO TE EMPUJA HACIA ADELANTE”
(ANÓNIMO)