

**DISEÑO DE UN ALGORITMO PARA EL CÁLCULO DE
TRAYECTORIAS DE ROBOTS SERIADOS CON EL FIN DE
LOGRAR EL PRE-AGARRE EN POSICIÓN Y DIRECCIÓN DEL
EFECTOR FINAL, MEDIANTE VISIÓN DE MÁQUINA.**

ALEXIS YAMID GHISAYS ABRIL

UNIVERSIDAD MILITAR NUEVA GRANADA

FACULTAD DE INGENIERIA

INGENIERIA EN MECATRÓNICA

BOGOTÁ DC.

2013

**DISEÑO DE UN ALGORITMO PARA EL CÁLCULO DE
TRAYECTORIAS DE ROBOTS SERIADOS CON EL FIN DE
LOGRAR EL PRE-AGARRE EN POSICIÓN Y DIRECCIÓN DEL
EFECTOR FINAL, MEDIANTE VISIÓN DE MÁQUINA.**

ESTUDIANTE:

ALEXIS YAMID GHISAYS ABRIL

**TRABAJO DE GRADO PARA OPTAR EL TITULO DE INGENIERO
EN MECATRÓNICA**

TUTOR:

MAURICIO FELIPE MAULEDOUX MONROY

**DOCENTE TIEMPO COMPLETO PROGRAMA DE INGENIERÍA
MECATRÓNICA.**

UNIVERSIDAD MILITAR NUEVA GRANADA

FACULTAD DE INGENIERIA

INGENIERIA EN MECATRÓNICA

BOGOTÁ DC.

2013

PÁGINA DE ACEPTACIÓN

Nota de aceptación

El trabajo de grado titulado "DISEÑO DE UN ALGORITMO PARA EL CÁLCULO DE TRAYECTORIAS DE ROBOTS SERIADOS CON EL FIN DE LOGRAR EL PRE-AGARRE ÓPTIMO EN POSICIÓN Y DIRECCIÓN DEL EFECTOR FINAL, MEDIANTE VISIÓN DE MÁQUINA" del autor Alexis Yamid Ghisays Abril, cumple con los requisitos para optar al título de Ingeniero en Mecatrónica.

Firma del Jurado

Firma del Lector

Firma del Tutor

Bogotá DC, 2013.

DEDICATORIA

A Dios, que me llena de bendiciones diariamente y siempre me ha ayudado a tomar las mejores decisiones en mi vida.

A mi madre, padre y hermana, que me formaron como ser humano y quienes aportaron de diferentes maneras lo necesario para hoy optar por mi título profesional.

A mi familia, que me han apoyado en todo, y a los cuales les debo todo lo que soy y todo lo que tengo.

A mis amigos con quienes pasé momentos inolvidables y siempre me dieron la motivación de seguir adelante y nunca rendirme.

AGRADECIMIENTOS

A mis profesores quienes aportaron todos los conocimientos necesarios para el desarrollo intelectual de este trabajo de grado.

Al Doctor Oscar Fernando Avilés Sánchez quien siempre me apoyó y estuvo al tanto de mi desarrollo como profesional.

Principalmente al Doctor Mauricio Felipe Mauledoux Monroy, tutor de mi tesis, que me guió en esta última etapa académica, y confió en mí para el desarrollo de este trabajo de grado.

A los ingenieros Samuel Quintero Molina y Sofía Barrios Arboleda por sus explicaciones y conocimientos ofrecidos que ayudaron a la solución de dificultades que se presentaron.

A la Universidad Militar Nueva Granada, que aportó elementos indispensables para la óptima ejecución del trabajo de grado.

TABLA DE CONTENIDO

INTRODUCCIÓN	12
1. JUSTIFICACIÓN.....	13
2. OBJETIVOS.....	14
2.1 OBJETIVO GENERAL	14
2.2 OBJETIVOS ESPECIFICOS	14
3. ESTADO DEL ARTE.....	15
4. MARCO REFERENCIAL.....	18
4.1 MARCO TEORICO.....	18
4.1.1 Visión de Máquina o Visión Artificial :	18
4.1.2 Robótica:.....	18
4.1.3 Robótica teleoperada:.....	19
4.1.4 Técnica de Optimización: Aproximación Chevyshev.	19
4.2 MARCO CONTEXTUAL	20
4.3 MARCO METODOLOGICO	21
4.3.1 Metodología:	21
4.3.2 Dispositivos utilizados:.....	21
4.3.3 Software utilizado:.....	22
5. SOLUCIÓN DE INGENIERÍA.....	23
5.1 Planteamiento de posibles soluciones.	23
5.2 Matriz de decisión.	23
5.3 Desarrollo.	25
5.3.1 Diseño mecánico:	25
5.3.2 Desarrollo del Software.....	27
6. RESULTADOS Y ANÁLISIS DE RESULTADOS.....	40
6.1 RESULTADO SOFTWARE.	40
6.1.1 RESUTADOS DE LA OBTENCIÓN Y MANIPULACIÓN DE LOS DATOS DE VISIÓN.....	40

6.1.2 TRATAMIENTO DE IMÁGENES.	43
6.1.3 RESULTADOS PLANEADOR DE TRAYECTORIA.	45
6.1.4 Imágenes del desarrollo del sistema integrado:.....	51
7. CONCLUSIONES	54
8. RECOMENDACIONES y trabajo futuro	56
9. BIBLIOGRAFÍA	58
10. ANEXOS	62
10.1 Planos	62
10.2 Diagramas de bloques:.....	63

LISTA DE FIGURAS

Figura 1: Gráfico de decisión	24
Figura 2: Diagrama de flujo (Algoritmo) del trabajo de grado	27
Figura 3: Formato STL en bruto.....	28
Figura 4: Formato STL ordenado.....	29
Figura 5: Subvi STLtoXYZ	29
Figura 6: Estructura de Vectores a Planos.	31
Figura 7: Estructura de Plano a Imagen y tratamiento de imagen.	32
Figura 8: Bloque para detección de formas.	33
Figura 9: Diagrama de flujo del árbol de decisión	34
Figura 10: Bloque para el entorno 3D.	35
Figura 11: Estructura del Planeador de trayectorias	37
Figura 12: Código del robot generado mediante Teach Pendant.....	39
Figura 13: Representación real del objeto.	40
Figura 14: Representación de Objeto STL.....	41
Figura 15: Modelo representado en vectores XYZ.....	41
Figura 16: Plano XY.....	42
Figura 17: Plano XZ	42
Figura 18: ImageTop e ImageSide antes del tratamiento de imágenes.....	43
Figura 19: ImageTop e ImageSide después del tratamiento de imágenes.	43
Figura 20: Entorno 3D reconstruido.	44
Figura 21: Características del Planeador de trayectorias.	45
Figura 22: Trayectoria X y Y, para 5 puntos.	46
Figura 23: Gráfica XY - Trayectoria vista desde el Plano Z, para 5 puntos.	46
Figura 24: Trayectoria tridimensional XYZ, para 5 puntos.	47
Figura 25: Código del Robot, para 5 puntos.	47
Figura 26: Trayectoria X y Y, para 100 puntos.....	48
Figura 27: Gráfica XY - Trayectoria vista desde el Plano Z, para 100 puntos.	48
Figura 28: Trayectoria tridimensional XYZ, para 100 puntos.	49
Figura 29: Código del Robot, para 100 puntos.	50
Figura 30: Comparación entorno real con reconstrucción.	51
Figura 31: Posición inicial de sensado 3D.	51
Figura 32: Posición final de pre-agarre al objeto.....	52
Figura 33: Interfaz gráfica programa principal.	52
Figura 34: Interfaz gráfica programa planeador de trayectorias.	53

LISTA DE TABLAS

Tabla 1: Matriz de decisión.....	24
Tabla 2 : Resultados obtenidos Real - Virtual.....	26
Tabla 3: Puntos de trayectoria del robot.	39
Tabla 4: Comparación de dimensiones Real-Virtual.....	44

RESUMEN

Mediante un dispositivo de visión 3D sujeto al efector final de un manipulador de la empresa MITSUBISHI, se logró realizar una reconstrucción tridimensional del espacio de trabajo, y dentro de este se ubicó una figura geométrica que el robot identificaba y detectaba en que parte del espacio de alcance se encontraba dicho elemento. A continuación teniendo el punto inicial y final de la trayectoria del robot, se logró un planeador de trayectorias del manipulador mediante una técnica de optimización conocida como aproximación Chevyshev, definiendo una función matemática, orden de dicha función y un número de puntos para construir la trayectoria. Finalmente después de tener definido todo lo anterior, se generó el código de movimiento del robot.

ABSTRACT

Using a 3D vision device grabbed to the end of the effector of a MITSUBISHI robot, it was possible to make a three-dimensional reconstruction of its workspace and within this a geometric figure was situated, the robot identified the position of the object. Then having the start and end point of the trajectory of the robot, it was achieved a trajectory planner for the manipulator using a optimization technique known as Chebyshev approximation, defining a mathematical function, its order and a number of points to build the trajectory. Finally, after having defined the above, code was generated robot motion.

INTRODUCCIÓN

La integración de la visión artificial o visión de máquina con la robótica , ha venido solucionando el problema básico de este último, que es el de planificación de movimiento para resolver alguna tarea pre-especificada, y luego controlar al robot cuando ejecuta las ordenes necesarias para conseguir esas acciones. Esta planificación hace referencia a la decisión de un curso de acciones antes de actuar. [41]

Poder darle la opción a un manipulador de detectar su entorno tridimensional mediante un dispositivo de visión de máquina, y entrenarlo para tomar decisiones según el espacio que lo rodea, podrían lograrse autómatas más autosuficientes y más seguros para cualquier ambiente, debido a que se disminuiría el riesgo de colisiones con obstáculos cercanos, así como disminución en la utilización de sensores.

Lograr la reconstrucción 3D de objetos reales, da la posibilidad de aumentar las aplicaciones, ya que ahora tendría información de profundidad de objetos y espacio de trabajo. También es importante darle al manipulador posibilidades diferentes a trayectorias rectas, que en varias ocasiones no son posibles, a causa del entorno que rodea al robot.

En los trabajos, proyectos y distintas publicaciones que se han revisado sobre esta temática, se encontraron integraciones de visión de máquina en 2D y robótica de manipuladores, pero no se encontró gran cantidad de estado del arte relevante referenciados a la visión artificial 3D con la robótica industrial. Además que en lo encontrado, se daba importancia a llegar al objeto a sujetar de forma directa, sin posibilidades de una trayectoria diferente, debido a que no era necesario en estas aplicaciones.

El proyecto de grado busca la integración de dos áreas del conocimiento como lo son la Robótica enfocado a manipuladores y la Inteligencia Artificial en la parte de visión de máquina o visión artificial y tiene como propósito principal, la utilización de un dispositivo de visión 3D como lo es el PrimeSense Camire 1.09, sujeto a un robot de la compañía MITSUBUSHI de la gama RV-6SQL, para que este logre decisiones dependiendo de su ambiente o espacio de trabajo y finalmente un planeador de trayectorias para la movilidad del robot mediante la utilización de la aproximación Chevyshev como técnica de optimización.

1. JUSTIFICACIÓN

En el área de ensamble y desensamble de productos mediante el uso de manipuladores es una problemática o limitante buscar la trayectoria de estos para lograr el pre-agarre necesario para posicionar y direccionar el efector final a un elemento específico. Industrialmente se utilizan varios robots manipuladores para el ensamble y desensamble, y es muy importante no solo enfocarse en la sujeción de un elemento en particular, sino en la trayectoria para el pre-agarre de este.

La mayoría de los desarrollos en los que se mancomuna el tema de visión de máquina con el tema de robots manipuladores, se enfocan en sujetar distintos elementos sin importar la trayectoria del robot ni como este se aproxime al objeto.

Existen innumerables campos del conocimiento en los cuales el tema de visión artificial y tratamiento de imágenes son totalmente aplicables y aún más sabiendo que en la actualidad el área de robótica tele-operada se encuentra en un gran auge.

El tratamiento de imágenes posee numerosas teorías o teoremas que por falta de conocimiento no se utilizan o se aprovechan, es importante tener presente todo esto para desarrollar una aplicación de visión de máquina exitosa.

Finalmente el diseño de un planeador de trayectorias mediante técnicas de optimización, sería algo innovador para manipuladores seriados y aún más para robots industriales MITSUBISHI.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Plantear un método para el cálculo de trayectorias de robots seriados con el fin de lograr el pre-agarre en posición y dirección del efector final, mediante visión de máquina.

2.2 OBJETIVOS ESPECÍFICOS

- Evaluar distintos dispositivos de visión para encontrar el más adecuado y llevar a cabo el respectivo tratamiento de imágenes.
- Determinar distintas características de objetos reales mediante tratamiento de imágenes y visión de máquina, con la precisión y fiabilidad necesaria para el adecuado desarrollo de la aplicación.
- Relacionar la información recopilada en el desarrollo de visión artificial con el diseño de un planeador de trayectorias para un manipulador seriado utilizando relaciones en el espacio tridimensional.
- Determinar características físicas como el espacio alcanzable, velocidades necesarias de movilidad para la selección de la gama del robot manipulador más adecuado para el cumplimiento del objetivo.
- Encontrar la trayectoria del manipulador seriado mediante alguna técnica de optimización, para lograr el pre-agarre en posición y dirección al efector final.

3. ESTADO DEL ARTE

DESARROLLO DE APLICACIONES INTEGRANDO ROBÓTICA Y VISIÓN EN UN ROBOT INDUSTRIAL KUKA PARA DEMOSTRAR SUS CAPACIDADES:

Proyecto de final de carrera de Pablo Muñoz Rodríguez de la Universidad Politécnica de Valencia presentado en Septiembre del 2011 [1].

En este informe se muestra como mediante el desarrollo de un juego de casino, utiliza a un robot SCARA de la empresa KUKA para sujetar una esfera y soltarla aleatoriamente en un tablero de números con colores, y también para recolectar las fichas ganadas en el desarrollo del juego [1].

La visión artificial se utiliza para detectar las apuestas echar por distintos usuarios, detectando el color define el monto apostado y detectando la posición en el tablero de apuestas define el número al cual el usuario apostó [1].

DISEÑO DE UN AMBIENTE VIRTUAL MEDIANTE VISIÓN ARTIFICIAL PARA APLICACIONES EN TELEROBÓTICA:

Paper desarrollado por Jaime Masiá, Juan Ramón Rufino, Francisco Javier Tormo, José Ferri y Borja Mansilla en la Universidad Politécnica de Valencia, Campus de Alcoi [2].

Se utiliza un robot de 3 grados de libertad, 2 rotacionales y 1 traslacional para el desarrollo del proyecto. Mediante visión artificial, se detecta la mano de un usuario y dependiendo de los movimientos que el operario haga con su mano, el SCARA reaccionará de la misma manera [2].

Los movimientos X e Y del robot son los mismos de la mano frente al dispositivo de visión y el movimiento Z se controla mediante el alejamiento o acercamiento de la mano a la cámara [2].

DISEÑO Y CONSTRUCCIÓN DE UN ROBOT MANIPULADOR TIPO SCARA CON SISTEMA DE VISIÓN DE MÁQUINA, QUE PUEDA SER UTILIZADO COMO HERRAMIENTA DIDÁCTICA EN ASIGNATURAS DE ROBÓTICA Y CONTROL:

Trabajo de grado del Ingeniero César Eduardo Jiménez Robayo de la Universidad Militar Nueva Granada presentado en el año 2009 [3].

En este trabajo de grado, se lleva a cabo la construcción de un robot manipulador tipo SCARA, al cual se le aplicó toda la teoría de robótica para conocer su respectiva cinemática directa, inversa; también la planeación de trayectorias y el control de este [3].

Se utilizó la visión de máquina para agarrar esferas de colores (pimpones) que se encuentren dentro del área de trabajo del manipulador [3].

GENERACION DE TRAYECTORIAS PARA UN ROBOT MANIPULADOR UTILIZANDO PROCESAMIENTO DE IMAGENES Y SPLINES:

Paper desarrollado por J. A. Soto Cajiga, estudiante de posgrado Centro de ingeniería y desarrollo industrial, J. E. Vargas Soto, Director facultad de ingeniería Universidad Anahuac México sur y J. C. Pedraza Ortega, Investigador Centro de ingeniería y desarrollo industrial [4].

En este trabajo se presenta una propuesta para generar trayectorias para un robot manipulador, específicamente el robot manipulador RV-2AJ de Mitsubishi. A partir de una imagen específica que contenga un objeto, se le aplica un procesamiento de imagen para obtener el contorno del objeto [4].

Con ayuda de éste contorno y aplicando una interpolación por splines, se realizan los cálculos de la cinemática inversa para generar la trayectoria del contorno del objeto que debe seguir el manipulador. Se desarrolla un software de simulación en 3D para las pruebas del algoritmo y la visualización de los movimientos del manipulador. Finalmente, se realizan las pruebas reales con el robot manipulador [4].

El desarrollo más cercano y de más relevancia con lo propuesto fue el que se mencionó de último; ya que es una propuesta aplicada a un robot MITSUBISHI y también busca programar trayectorias dependiendo de un objeto mediante visión de máquina; la diferencia se basa en la aplicación de visión artificial, donde ellos utilizan el tratamiento de imágenes para detectar el borde de un objeto en 2D y luego que el robot realice en su trayectoria realice este recorrido.

Según el estado del arte consultado, gran parte de los desarrollos de aplicaciones tridimensionales utilizando dispositivos similares al Carmine 1.09 como lo es el Kinect, se basan en la reconstrucción de caras humanas [17, 18, 20] o en aplicaciones antropométricas para aplicaciones médicas [9].

Ahora, en otras aplicaciones tridimensionales consultadas, se utilizaron teorías de estéreo visión, las cuales consisten en la utilización de 2 cámaras y llevando a cabo una triangulación de píxeles dando así información de profundidad [5, 21, 22, 24], o usando las vistas de varias imágenes de un mismo entorno y relacionarlas entre sí para lograr el mismo fin [6, 12, 13, 15, 16, 19].

Para la detección de formas mediante tratamiento de imágenes, lo más utilizado fueron desarrollos que aplicaban la teoría de la transformada de Hough [26], la cual es de alta fiabilidad pero que al implementarla podría causar un alto nivel de complejidad a la aplicación, debido al extenso código necesario para la buena ejecución de este, además de ser inoficioso para figuras geométricas básicas que son las tratadas en este trabajo de grado.

Finalmente los artículos consultados referentes a aplicaciones con manipuladores MITSUBISHI se basaron en simulaciones virtuales de su movimiento además del cálculo de su cinemática inversa y directa para la movilidad simulada [23].

Pero no se encontraron desarrollos de aplicaciones de visión de máquina en 3D con el cálculo de trayectorias diferentes a recorridos lineales de un punto inicial a otro final. Además de la integración de un dispositivo novedoso como el Carmine 1.09 a un manipulador MITSUBISHI.

4. MARCO REFERENCIAL

4.1 MARCO TEORICO

4.1.1 VISIÓN DE MÁQUINA O VISIÓN ARTIFICIAL :

Los sensores proporcionan al robot información sobre su entorno que él puede usar para guiar sus acciones. La visión es un sentido muy importante porque puede dar información de una resolución relativamente alta a distancias relativamente grandes. Se han desarrollado muchos tipos de sistemas de visión para robots, los cuales entran en uso práctico cada vez con más frecuencia [42].

Los sistemas de visión para máquinas comenzaron a aparecer en el decenio iniciado en 1950, asociados con aplicaciones no robóticas, como lectura de documentos, conteo de especímenes biológicos en la sangre, y reconocimiento aerofotográfico de aplicación militar [42].

En el campo propiamente robótico el desarrollo comenzó a mediados del decenio iniciado en 1960, con el establecimiento de laboratorios de inteligencia artificial en instituciones como el MIT, la U. de Stanford y el Stanford Research Institute. Para 1980 muchas compañías estaban desarrollando sistemas de visión para robots y para otras aplicaciones industriales [42].

4.1.2 ROBÓTICA:

El conjunto de conocimientos teóricos y prácticos que permiten concebir, realizar y automatizar sistemas basados en estructuras mecánicas poliarticuladas, dotados de un determinado grado de "inteligencia" [43].

En 1979, el Robot Institute of America definiría un robot como: 'Un manipulador reprogramable y multifuncional diseñado para trasladar materiales, piezas, herramientas o aparatos específicos a través de una serie de movimientos programados para llevar a cabo una variedad de tareas. [44]

Veinte años más tarde, la Encarta de Microsoft da la siguiente definición: 'Máquina controlada por ordenador y programada para moverse, manipular objetos y realizar trabajos a la vez que interacciona con su entorno. Los robots son capaces de realizar tareas repetitivas de forma más rápida, barata y precisa que los seres

humanos'. El cambio más sustancial incorporado al concepto en estos veinte años es el hecho de contemplar una interacción del robot con su entorno, que permite comportamientos adaptativos e inteligentes. [44]

Se considera a un robot como un agente autónomo inteligente (AAI) cuando cumple los siguientes requisitos: [44]

- Autonomía: El sistema de navegación reside en la propia máquina, que debe operar sin conexión física a equipos externos.
- Inteligencia: El robot posee capacidad de razonar hasta el punto de ser capaz de tomar sus propias decisiones y de seleccionar, fusionar e integrar las medidas de sus sensores.

4.1.3 ROBÓTICA TELEOPERADA:

La palabra Telerobótica está compuesta por "tele" y por "robótica". Por una parte, tele significa transmitir datos o control a distancia y por la otra, robótica, dando una definición general, la técnica para el diseño y estudio de robots. Es decir, Telerobótica es la ciencia o técnica para el diseño y estudio de robots a los cuales se les transmite datos o son controlados a distancia. La Telerobótica es un campo muy importante y que se está expandiendo rápidamente a la par del desarrollo de procesadores más rápidos, nuevos algoritmos y expectativas mayores. Hay muchas aplicaciones en Telerobótica en uso hoy en día, desde exploraciones espaciales, aplicaciones biomédicas hasta exploraciones en áreas de alto peligro biológico. La Administración Nacional de Aeronáutica y del Espacio (NASA por sus siglas en inglés) es el líder en investigación en el campo de la Telerobótica, sin embargo existen otros entes en otros campos de aplicación como la medicina que se encuentran desarrollando aplicaciones para el uso de la Telerobótica. [45]

4.1.4 TÉCNICA DE OPTIMIZACIÓN: APROXIMACIÓN CHEVYSHEV.

Para un número natural n dado, la siguiente ecuación representa aproximadamente la función $f(x)$.

$$f(x) = c_0 T_0(x) + \dots + c_n T_n(x)$$

donde $T_0(x), \dots, T_n(x)$ son los primeros polinomios de Chebyshev. Se puede calcular la c_0, \dots, c_n como sumas de la forma:

$$c_0, \dots, c_n = \sum_{k=1}^n f(x_k) T_j(x_k)$$

donde:

$$x_k = \cos\left(\frac{\pi * k}{n}\right); \text{ para } k = 1, \dots, n.$$

Para llevar a cabo la aplicación de esta técnica dentro del programa a desarrollar, se utilizó la herramienta Chebyshev Approximation, ubicada dentro de la paleta de bloques de optimización de LABVIEW; la cual determina una función dada mediante polinomios de Chebyshev.

4.2 MARCO CONTEXTUAL

El marco en el que se desarrolló el proyecto de grado tiene principalmente un lugar de desarrollo.

Todo el aspecto mecánico se llevó a cabo en la instalación del taller de mecánica de la Universidad Militar Nueva Granada usando los esmeriles, fresadoras, taladros de árbol, seguetas y otras herramientas necesarias.

La manipulación y programación del robot se ejecutó en el laboratorio de Automatización Industrial de la Universidad Militar Nueva Granada en el cual se encuentra el robot MITSUBISHI RV-6SQL con su respectivo Teach Pendant. La interfaz gráfica y programación se pudo desarrollar en las instalaciones de la universidad.

La planeación y ejecución de este proyecto de grado estuvo comprendido entre Junio y Diciembre del año 2013.

4.3 MARCO METODOLOGICO

4.3.1 METODOLOGÍA:

El trabajo de grado se desarrolló primero identificando el estado del arte, y todas las problemáticas que se presentaron en la ejecución, elaboración e integración de visión de máquina con robots seriados.

De acuerdo a esto se plantearon las distintas soluciones validas que pudieron llegar a desarrollarse en la ejecución del trabajo de grado.

Conociendo la solución más viable, se procedió a desarrollar todas las técnicas y teorías de visión de máquina necesarias para lograr una óptima detección de objetos reales y sus características.

Finalmente, se buscó la forma más adecuada de vincular lo desarrollado en visión de máquina y la planeación de trayectorias del robot seriado.

4.3.2 DISPOSITIVOS UTILIZADOS:

4.3.2.1 Robot Mitsubishi RV-6SQL:

El manipulador utilizado para el desarrollo del trabajo de grado es el robot Mitsubishi RV-6SQL; el cual posee entre sus características importantes que soporta una carga máxima de 6Kg, posee 6 articulaciones y tiene un alcance de 902 mm.

4.3.2.2 PrimeSense Carmine 1.09:

Este dispositivo es un sensor infrarrojo similar al dispositivo Kinect. Sensa entornos 3D con un alcance mínimo de 0,35m y un alcance máximo de 1,4m. Su alimentación de poder es vía USB, lo que lo hace un dispositivo de fácil uso e integración, y tiene un tamaño de imagen de profundidad de 640 x 480 (VGA).

4.3.3 SOFTWARE UTILIZADO:

4.3.3.1 Skanect:

Este programa se utilizó para lograr la adquisición 3D del objeto real. Se utilizó este, debido a que es un programa de fácil utilización y aunque se usó la versión de prueba que posee grandes limitantes, fue suficiente para lograr la reconstrucción 3D deseada.

4.3.3.2 Labview:

La programación en labview facilitó el desarrollo del trabajo de grado ya que esta herramienta posee un módulo dedicado al tratamiento de imágenes, además de separar los procesos de manera gráfica, mediante la utilización de subvis, que son pequeños subprogramas que reducen la complejidad del programa.

4.3.3.3 SolidWorks:

Se utilizó SolidWorks para la construcción del CAD tanto del sensor PrimeSense como de la caja diseñada para lograr la sujeción con el robot Mitsubishi.

4.3.3.4 Dia:

Software utilizado para la fácil construcción de diagramas de flujo o flujo gramas necesarios para la explicación y entendimiento de lo realizado en el desarrollo del trabajo de grado.

5. SOLUCIÓN DE INGENIERÍA

5.1 PLANTEAMIENTO DE POSIBLES SOLUCIONES.

Inicialmente se plantearon distintas soluciones para el problema planteado, dando así las siguientes opciones:

Opción 1: Utilización de una cámara USB sujeta al robot y mediante la vista de varias caras del objeto sensado utilizando el alcance del robot, lograr una idea de a que figura geométrica pertenecía el elemento.

Opción 2: Mediante el uso de 2 cámaras sujetas al robot, llevar a cabo el mismo procedimiento anterior, solo que ahora teniendo información de dos vistas por imagen, lograr la triangulación de las imágenes y calcular distancias tridimensionales del objeto.

Opción 3: Usando un Kinect sujeta al manipulador y utilizando su cámara de profundidad para detectar el objeto a sensar e identificar sus propiedades físicas. Era necesaria la clara utilización de las librerías de este dispositivo existentes para el lenguaje C.

Opción 4: Con el dispositivo PrimeSense Carmine 1.09 se tenía pensado llevar a cabo el sensado del entorno donde se encontrará la figura geométrica mediante la utilización de un programa (software) externo, como lo es el Skanect, se obtiene el formato STL de dicho entorno y mediante técnicas de programación y el método de inferencia detectar la forma geométrica a la cual el objeto pertenece.

5.2 MATRIZ DE DECISIÓN.

Para poder detectar la solución más viable, se definieron varias características de importancia y a cada opción se le dio un valor con respecto a la propiedad mencionada, entre las cuales se encuentran:

- Acople al efector del robot.
- Utilización y programación de la información obtenida.
- Alimentación del dispositivo a utilizar.

- Espacio utilizado.
- Valido para aplicaciones tridimensionales.

El valor descrito anteriormente describe el valor de importancia de cada opción frente a cada característica, donde 1 es lo menos adecuado y 10 lo más adecuado o viable (Tabla 1).

Tabla 1: Matriz de decisión.

Características de importancia \ Opciones	Opciones			
	1	2	3	4
Acople al efector del robot.	8	7	8	8
Utilización y programación de la información obtenida.	8	4	7	9
Alimentación del dispositivo a utilizar.	8	6	6	9
Espacio utilizado.	8	8	8	8
Valido para aplicaciones tridimensionales.	1	8	10	10
	4096	10752	26880	51840

Fuente: Autor.

Figura 1: Gráfico de decisión



Fuente: Autor.

Se puede ver (Figura 1) y finalmente deducir que la opción más viable es la número 4.

Los valores de importancia que se le dieron a esta opción, están claramente justificados. Mediante el uso de un dispositivo más ligero y de tamaño más corto que el propio Kinect hace que el acople al efector final del robot sea más fácil, además que la alimentación USB del Carmine1.09 le da ventajas frente a los otros dispositivos, como el Kinect que es de mayor peso y de alimentación AC.

5.3 DESARROLLO.

5.3.1 DISEÑO MECÁNICO:

Debido al uso del dispositivo Carmine 1.09, y a su particular forma, fue necesaria la construcción de una caja, la cual contenga dicho dispositivo y permita una sujeción fácil al efector final del robot manipulador.

5.3.1.1 Materiales.

Polimetilmetacrilato (Vidrio Acrílico): Plástico de ingeniería que se utilizó para construir la caja contenedora del sensor PrimeSense 1.09, que irá sujeta al robot manipulador.

Poliestireno (Icopor): Se utilizó para darle rigidez al dispositivo dentro de la caja, y lograr una posición fija de este, que no varíe aun cuando el robot esté en movimiento. Además de proteger el sensor de golpes, caídas, vibraciones y humedades.

5.3.1.2 Diseño CAD mecánico.

Inicialmente para el diseño mecánico de la caja, fue necesario la construcción en CAD del dispositivo, con la herramienta SOLIDWORKS. Esto para conocer sus

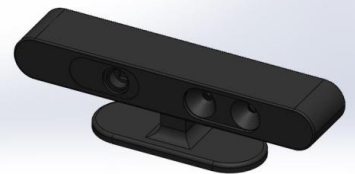

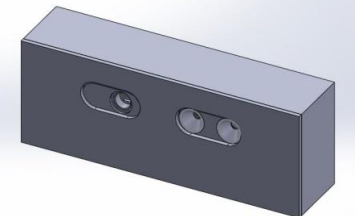

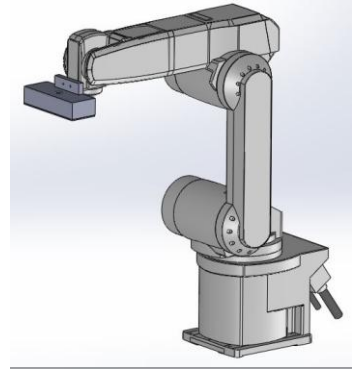

dimensiones tanto de ancho, alto y profundidad; además de la separación de sus lentes y el diámetro de estos.

5.3.1.3 Resultado diseño mecánico.

A continuación se compara el dispositivo real con el desarrollado virtualmente (Tabla 2).

El primero hace referencia al dispositivo Carmine 1.09. El segundo, a la caja contenedora de este. El tercero, al ensamble de los dos anteriores con el robot Mitsubishi.

Tabla 2 : Resultados obtenidos Real - Virtual.

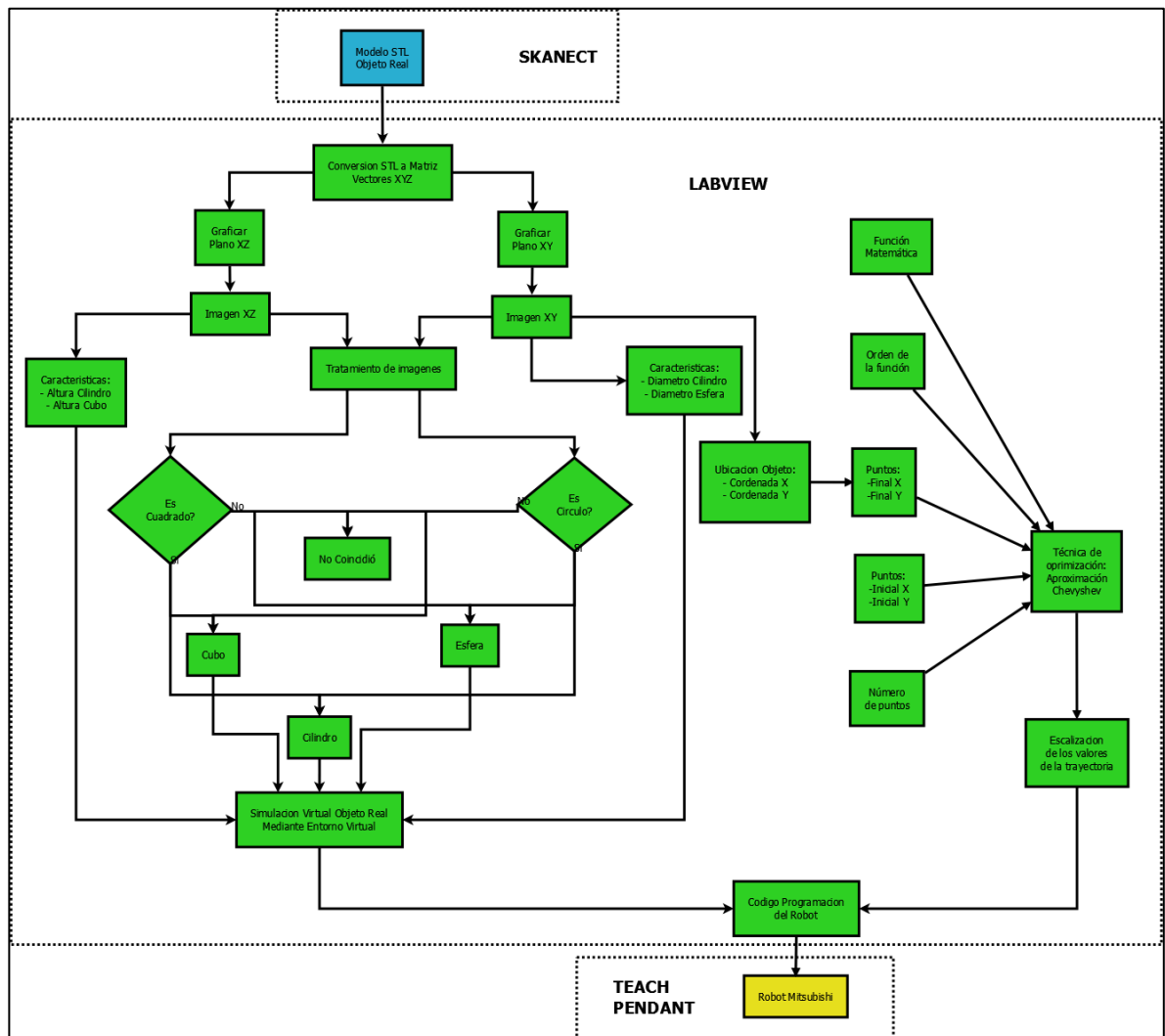
Elemento	Virtual CAD	Real
Sensor PrimeSense		
Caja Acrílico		
Ensamble		

Fuente: Autor.

5.3.2 DESARROLLO DEL SOFTWARE.

En el desarrollo del trabajo de grado se estableció un algoritmo que contempló una serie de pasos, que se pueden evidenciar en la Figura 2.

Figura 2: Diagrama de flujo (Algoritmo) del trabajo de grado



Fuente: Autor.

Cada parte del desarrollo mostrado en el gráfico anterior, se explicará a continuación y al final del capítulo se mostrarán los respectivos resultados obtenidos tanto del diseño mecánico como del diseño de software o programación.

5.3.2.1 OBTENCIÓN Y MANIPULACIÓN DE DATOS DE VISIÓN:

Como se puede evidenciar en el algoritmo dado, el primer paso fue la obtención del modelo virtual del objeto real, que ofrece el programa SKANECT.

Este software permite exportar este modelo en varios formatos:

- STL (STereo Lithography)
- OBJ
- VRML (Virtual Reality Modeling Language)
- PLY (Polygon File Format)

Para el desarrollo del trabajo de grado se decidió tomar archivos en STL, que es un formato de archivo informático, que fue creado principalmente para la industria del prototipado rápido por lo cual se enfoca en la precisión de las distancias y características reales del objeto y excluye propiedades como color y textura de estos, que para esta aplicación no son relevantes.

Al analizar el formato STL, se encontró que éste programa generaba un archivo que al ser visualizado en block de notas, entregaba la información desorganizada (Figura 3).

Figura 3: Formato STL en bruto.



Fuente: Autor.

Después de haber analizado esto, se buscó patrones que dieran la información necesaria para precisar las coordenadas X, Y y Z del objeto real.

Se encontró que, después de organizar esta información utilizando la herramienta LABVIEW, los datos X, Y y Z reales se podían evidenciar en la siguiente imagen (Figura 4).

Figura 4: Formato STL ordenado



```

solid vcg
facet normal -0.244454 -0.967885 0.0586503
outer loop
vertex -206.935 9.30563 -70.2333
vertex -202.252 8.26159 -67.9472
vertex -206.938 9.37823 -69.0504
endloop

```

Fuente: Autor.

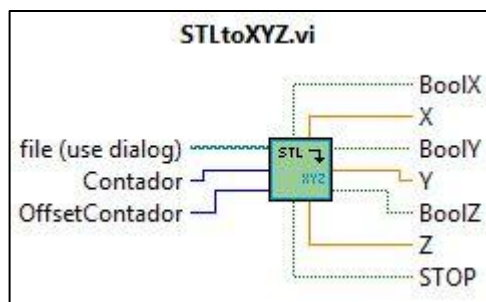
Los valores numéricos después de la palabra "vertex" hacen referencia a las coordenadas deseadas así:

vertex coordenadaX coordenadaZ coordenadaY.

Los valores después de "facet normal" fueron ignorados debido a que no se encontró ninguna relación con el modelo virtual del objeto real, además que siempre presentaban valores entre 1 y -1, lo que ayudo a que posteriormente mediante programación, filtrar estos valores y así pudieran ser eliminados.

El subvi construido en LABVIEW que permitió la clasificación de los valores X, Y y Z fue el siguiente (Figura 5).

Figura 5: Subvi STLtoXYZ



Fuente: Autor.

El diagrama de bloques de este subvi se mostrará en la sección de anexos (10.2) para mejor entendimiento.

A este bloque se le definieron 3 entradas y 7 salidas que corresponden a:

- file: Indica la dirección donde se encuentra el archivo STL.
- Contador: Va conectado, en el programa principal, al contador i de la estructura while. Este valor indica la línea de texto que el programa evaluará para identificar y separar los valores X, Y y Z.
- Offset Contador: Este valor se le introdujo al programa por razones de tiempo de proceso. Al correr el programa con este valor en 0, se encontró que demoró alrededor de 1 minuto en identificar todas las muestras del archivo STL y además se observó que eran alrededor de 35000 muestras y que los valores de Z, que son los más importantes, se guardaban al final de estas muestras. Es debido a esto que para agilizar el programa es ideal inicial el contador en un valor como 20000 o 25000, ignorando estos primeros datos.
- Bool X Y y Z: Son indicadores booleanos, es falso si cada valor se encuentra dentro del rango -1 y 1. En caso que no se encuentre en este rango, será verdadero.
- X Y y Z: Indicadores de los vectores numéricos que posteriormente almacenaran consecutivamente dicha información.

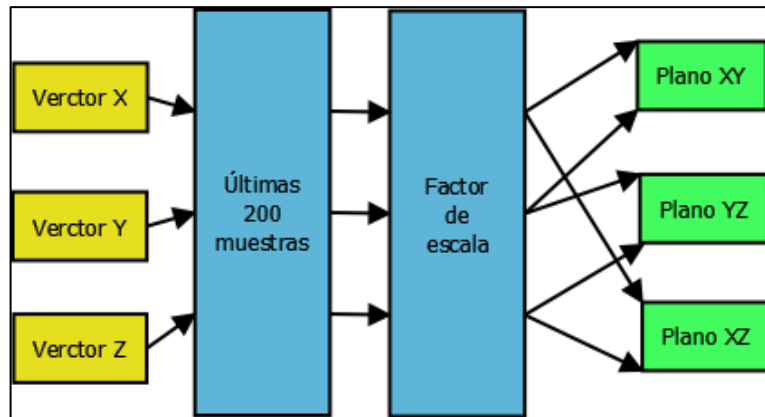
De manera sencilla, lo que el programa hace es identificar la ubicación del archivo STL, leerlo, y línea por línea elimina palabras reservadas del formato como lo son "vertex" y "facet normal", dejando así solo valores numéricos fuera del rango -1 y 1, que serán almacenados en vectores X, Y y Z por aparte.

Esta clasificación de coordenadas en vectores X, Y y Z se mostrará más detalladamente en la sección de anexos, mediante su respectivo diagrama de bloques.

Las imágenes obtenidas y modelos reconstruidos se mostrarán en el capítulo de resultados y análisis de resultados.

Teniendo el modelo ya almacenado en estos vectores, se pueden construir matrices con la información XY y XZ del objeto y graficarlos, esto necesario para el siguiente paso del algoritmo (Figura 6).

Figura 6: Estructura de Vectores a Planos.



Fuente: Autor.

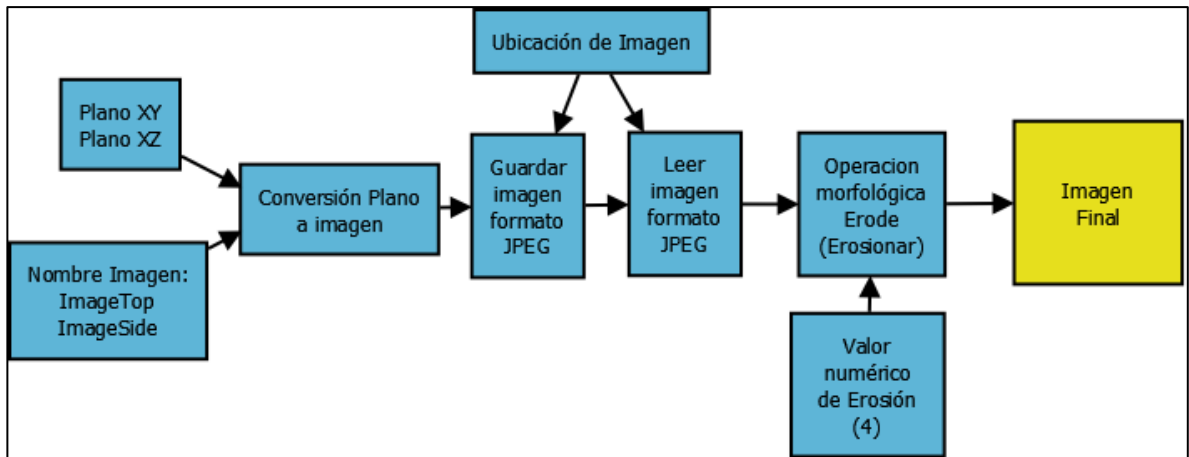
El diagrama mostrado en la Figura 6, hace referencia al programa que utiliza la información de los vectores XYZ, para graficar los 3 planos básicos. Se puede notar que hay un factor de escala, esto para hacer el objeto más grande y facilitar después la detección del objeto. Además se toman solo las últimas 200 muestras de los vectores, esto debido a que es estos últimos datos se encuentra la información del objeto a analizar; este valor puede variar dependiendo del objeto.

5.3.2.2 TRATAMIENTO DE IMÁGENES.

Vale la pena resaltar que el tratamiento de imágenes se logró con el modulo NI-IMAQ, el cual posee una gran cantidad de herramientas que fueron utilizadas y facilitaron el desarrollo de este paso.

Inicialmente para poder llevar a cabo el respectivo tratamiento de imágenes, fue necesario convertir la información de los planos en imágenes que LABVIEW pudiera reconocer (Figura 7).

Figura 7: Estructura de Plano a Imagen y tratamiento de imagen.



Fuente: Autor.

La vista superior nos indica el plano XZ del objeto por lo cual este plano se guardo como una imagen de formato JPG y con nombre ImageTop (ImageSide para el plano XY), que inmediatamente después es leída, ya como imagen, para aplicarle una operación morfológica a esta.

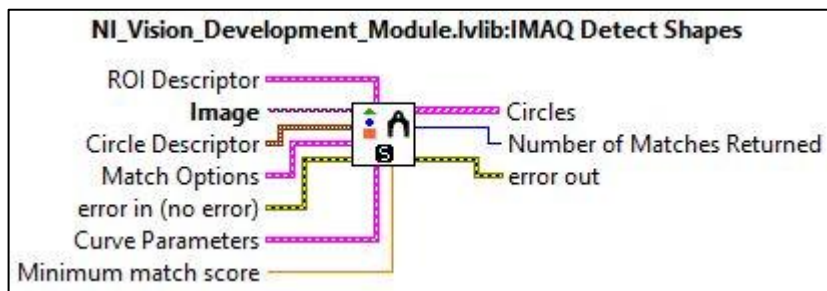
La operación morfológica utilizada fue Erode, que hace que, con el valor especificado, en este caso 4, recorre la imagen pixel por pixel, y si encuentra un pixel diferente a un color blanco, lo agranda 4 veces en todas sus direcciones; de esta forma se garantiza que la forma del objeto de análisis siga siendo la misma.

Tanto para el plano XZ como el XY se realizó el mismo proceso.

Los resultados reales se mostrarán en el capítulo de resultados y análisis de resultados, al igual que los diagramas de bloques contruidos ubicados en la sección de anexos.

Para la identificación de formas geométricas se utilizó el bloque del modulo NI-IMAQ llamado IMAQ Detect Shapes (Figura 8).

Figura 8: Bloque para detección de formas.



Fuente: Herramienta de ayuda de LABVIEW.

Este bloque no solo detecta círculos sino también rectángulos, líneas o elipses; y aplicando esta operación a ambas imágenes (ImageTop y ImageSide) poder aplicar el árbol de decisión.

5.3.2.3 ÁRBOL DE DECISIÓN MEDIANTE MÉTODO DE INFERENCIA.

Para tomar la decisión de si el objeto sensado es un cilindro, esfera o cubo, se plantearon inicialmente varias posibles soluciones.

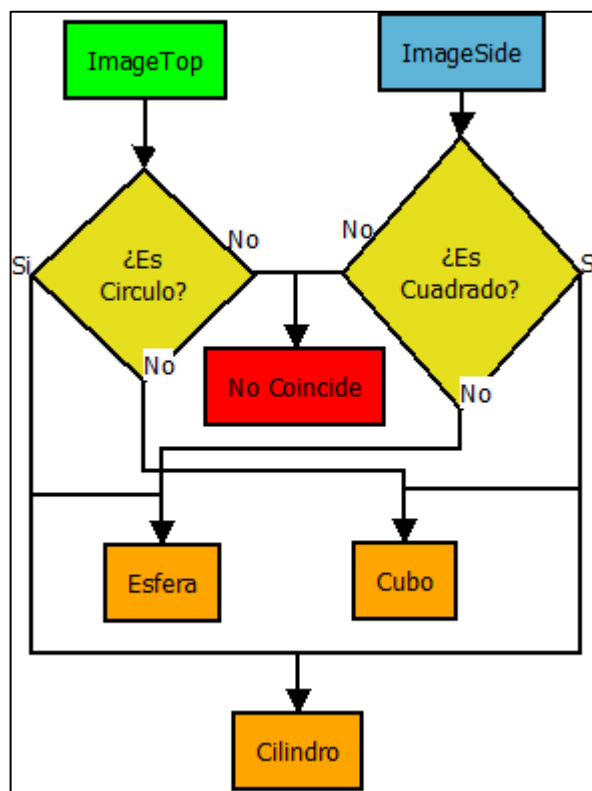
Inicialmente se pensó utilizar redes neuronales para la toma de decisión, pero se descartó dicha solución, debido a que la programación de estas redes son inoficiosas para la aplicación deseada y sería utilizar una herramienta de un alto nivel de programación para dar solución a un problema sencillo.

Después se pensó en utilizar el mismo movimiento del robot para observar y sensor más caras del objeto, pero el mismo alcance del robot limitaba algunas vistas y hacia el desarrollo de la aplicación menos eficiente.

Finalmente se utilizó el método de inferencia donde conociendo solo características como la imagen superior y lateral del objeto se pueden descartar opciones dejando la más probable como la alternativa válida.

El árbol de decisión por el método de inferencia se presentan en la Figura 9.

Figura 9: Diagrama de flujo del árbol de decisión



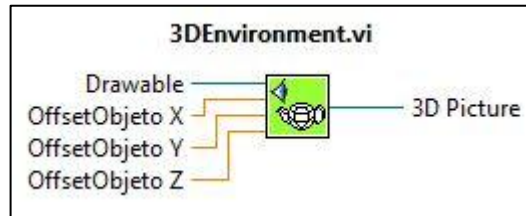
Fuente: Autor.

5.3.2.4 ENTORNO VIRTUAL 3D.

Lo primero para realizar el entorno virtual 3D, donde se reconstruirá el entorno sensado, se construyó un bloque (subvi) que facilitará el desarrollo del programa, reduciendo una gran cantidad de bloques a uno solo (Figura 10).

A este bloque se le han definido internamente 2 estructuras 3D; una de estas es el suelo del espacio de trabajo del robot, definido como un rectángulo de espesor muy pequeño y de lados X y Y calculados como el mayor valor menos el menor valor de los vectores X y Y. Y la segunda estructura definido como el gripper del robot al cual no se le hará ninguna transformación, simplemente se define con fines de simulación.

Figura 10: Bloque para el entorno 3D.



Fuente: Autor.

Este bloque posee definidas 4 entradas y una salida:

- **Drawable:** Es donde va el objeto geométrico 3D (esfera, cilindro o cubo).
- **OffsetObjetoX:** Hace referencia al distanciamiento del objeto a su origen respecto al eje X.
- **OffsetObjetoY:** Hace referencia al distanciamiento del objeto a su origen respecto al eje Y.
- **OffsetObjetoZ:** Este valor es la distancia que debemos aumentarle al objeto en el entorno para que este no se vea cortado a la mitad por el suelo del espacio de trabajo ya definido. Es debido a esto que este valor siempre será la mitad de la altura del objeto sensado.
- **3D Picture:** A esta salida va conectado el controlador o visualizador 3D.

Después de tomar la mejor alternativa, se procede a recrear virtualmente el objeto sensado, calculando las dimensiones de este dependiendo de la forma geométrica, así:

- **Cilindro:** Para determinar la altura del objeto, se empleó el vector que guardó todas las coordenadas en Z y se determinó el valor máximo y mínimo, se restaron y así se encontró el valor de altura.
Para el diámetro, se utilizó la matriz XY, y se aplicó el mismo procedimiento que para hallar la altura; solo que ahora teniendo 2 valores (X y Y), se hizo un promedio de estos y se encontró un diámetro promedio.
- **Esfera:** Para encontrar el diámetro se procedió a hacer el mismo método utilizado para la obtención del diámetro en el caso del cilindro.

- Cubo: Las características del cubo como lo son altura y lado, se calcularon del mismo modo que en el caso del cilindro:
Para la altura del cubo utilizando el vector correspondiente a Z, y para los lados del cubo utilizando la matriz XY.

5.3.2.5 MOVIMIENTO Y TRAYECTORIA DEL ROBOT MANIPULADOR

5.3.2.5.1 Posición X y Y finales de la trayectoria del Robot:

Para relacionar la posición del objeto sensado con el entorno de trabajo real se llevó a cabo el siguiente desarrollo:

- Utilizando la imagen ImageTop, que es la figura que indica la posición X y Y del objeto, se detectaron dichos valores en pixeles de la imagen.
- Se ubicó un origen de la imagen virtual en el medio de esta, es decir si se encuentra que la posición X y Y de la figura es 213 para ambos, el elemento se encontrará en el origen o centro del espacio de trabajo del robot.
- En el caso que el objeto se encuentre fuera del origen, se calcula cuanto es la distancia en pixeles tanto para el eje X como para el eje Y y estos valores se escalizan a distancias reales del suelo de trabajo del robot.
Esta escala se llevó a cabo mediante una regla de 3 simple, así:

$$Distancia\ real\ X = \frac{distancia\ en\ pixeles\ X * 447.234\ mm}{426\ pixeles\ en\ X}$$

$$Distancia\ real\ Y = \frac{distancia\ en\ pixeles\ Y * 353.056\ mm}{426\ pixeles\ en\ Y}$$

Los valores de 447.234 mm y 353.056 mm, son las distancias reales que abarca el dispositivo con 426 pixeles respectivamente. Es decir, la imagen obtenida que es de 426 pixeles, equivale en dimensiones reales a 447.234 en X y 353.056 en Y.

- Teniendo los valores reales a los cuales el objeto esta corrido del centro, se procede a aumentarlos o disminuirlos a la posición central del gripper, definida como:

Centro del gripper en X: 390.52.

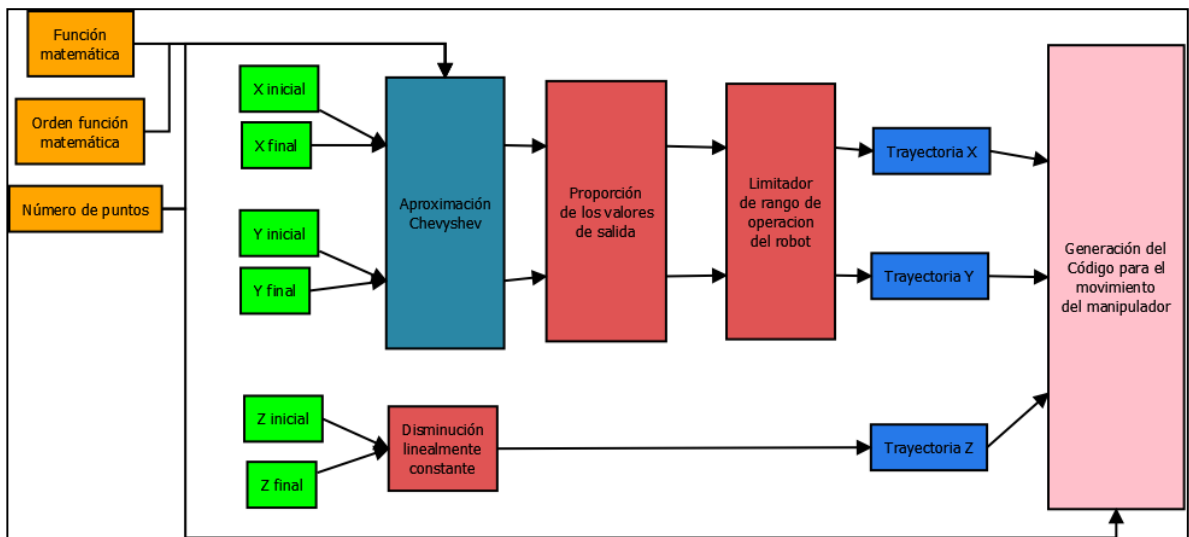
Centro del gripper en Y: -14.83.

- V. Finalmente a el valor real encontrado en X se le debe sumar la distancia real entre el foco del lente y el centro del gripper que es aproximadamente 108.19 mm.

5.3.2.5.2 Planeador de Trayectorias:

Teniendo los puntos X y Y finales a donde el robot debe llegar, y también los puntos X y Y iniciales que son aquellos donde se sensó el objeto, se procede a construir el planeador de trayectorias del manipulador (Figura 11).

Figura 11: Estructura del Planeador de trayectorias



Fuente: Autor.

Como se mencionó en el inicio del trabajo de grado, para optimizar la trayectoria del manipulador se utilizó la técnica de aproximación Chevyshev. Los valores necesarios para el buen funcionamiento de esta técnica son:

- Función matemática: para fines de simplicidad se tomará la función: $f(x)=\text{sen}(x)$.
- Orden de la función: puede tomar cualquier valor, pero hay que tener en cuenta que entre mayor sea el orden, la posibilidad de que los valores de la trayectoria se salgan del rango de movilidad del robot, aumenta.
- Número de puntos: Con este valor se puede demostrar la optimización de la trayectoria, a mayor cantidad de puntos la trayectoria a planear es más cercana a la función, pero aumenta en líneas de código del robot; y a menor cantidad de puntos, la cantidad de líneas se reduce pero la trayectoria deja de ser fiel a la función establecida.
- Puntos iniciales X,Y.
- Puntos finales X,Y.

Los valores después de pasar por la técnica de optimización se deben proporcionar debido a que estos valores tienden a estar entre -1 y 1 que es el rango de una función sinodal. Estos valores se escalan a valores de posiciones reales del robot.

Justo después de este proceso es importante limitar estos valores con los rangos posibles de movilidad del manipulador; esto significa que si alguno de los valores de la trayectoria construida ya sea en X o en Y se salen del rango de movilidad del robot, el programa no permite la construcción de dicho código, a causa que esto podría ser peligroso y el robot podría estrellarse o colisionar con elementos reales de su entorno.

Teniendo estos valores dentro del rango definido, las trayectorias X y Y del manipulador quedan construidas y guardadas en vectores independientes.

Para el caso de la trayectoria en el eje Z, se definió un punto muy cercano al objeto y teniendo el punto inicial se llevó a cabo una disminución constante del punto inicial al punto final. Esto se logró utilizando el número de puntos definido, ya que los 3 vectores finales X, Y y Z deben tener las mismas dimensiones.

Para programar la trayectoria del robot, inicialmente se utilizó el mismo método usado anteriormente para la obtención de datos del formato STL; se programó el robot Mitsubishi mediante su controlador (Teach Pendant) una trayectoria sencilla de puntos definidos, y se buscó algún patrón que hiciera referencia a las coordenadas X Y Z del entorno real (Figura 12).

Figura 12: Código del robot generado mediante Teach Pendant.

```

CAM: Bloc de notas
Archivo Edición Formato Ver Ayuda
1 Mov p0
2 Mov p2
3 Mov p3
p2=(+390.52, -14.83, +619.76, -179.98, +0.04, -129.57)(7,0)
p0=(+592.48, -0.38, +826.72, -153.76, +88.02, -153.78)(7,0)
p3=(+389.82, -14.83, +299.69, -179.98, +0.04, -129.57)(7,0)

```

Fuente: Autor.

Se observó que el código del robot presentaba una estructura clara y asequible, de fácil manipulación (Figura 12).

Inicialmente se guardan los comandos del robot, para este caso, movimiento de p0 (definido como la posición home o inicial del robot), luego p2 (definido como la posición central del gripper en dirección al suelo del espacio de trabajo) y finalmente p3 (punto de manipulación para acercarse al objeto); y después se definen los puntos con sus respectivas coordenadas X, Y y Z, que son los valores que se modificarán en el programa. Los últimos 3 valores corresponden a la orientación de la pinza que ha sido definida siempre perpendicular al suelo del espacio de trabajo (Figura 13).

Tabla 3: Puntos de trayectoria del robot.

Punto	X	Y	Z
P0	+592.48	-0.38	+826.72
P2	+390.52	-14.83	+619.76
P3	+389.82	-14.83	+299.69

Fuente: Autor.

Ahora para la construcción del nuevo código con las trayectorias guardadas en los vectores X, Y y Z, primero se programó una estructura de ciclo for que recorriera dichos arreglos desde 1 hasta el número de puntos definidos inicialmente. Esto para escribir la sección de comandos del robot y la sección de puntos del robot.

6. RESULTADOS Y ANÁLISIS DE RESULTADOS.

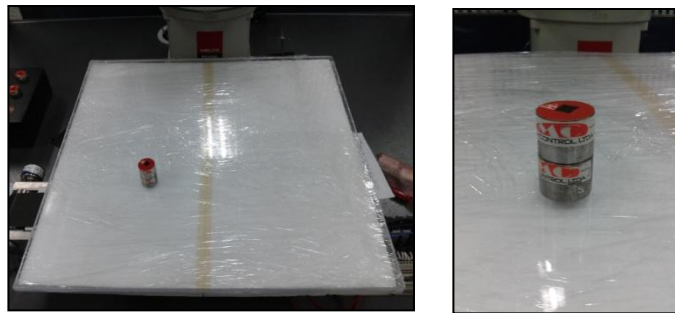
Para presentar los resultados, se definió el uso de una figura geométrica fija (cilindro), y a este elemento se le aplicó todo el procedimiento desarrollado.

En una estructura lisa (lámina de acrílico) ubicada al frente del manipulador a una altura de 20 cm paralelo a el suelo del robot, y de una dimensión de 50 cm de ancho por 50 cm de largo, lo que cubriría el espacio de visión del dispositivo (Figura 13).

El cilindro se ubicó sobre la lámina de acrílico, donde se sensó con el dispositivo de visión y se concluyó de que figura geométrica se trataba, reconstruyéndola de forma tridimensional.

Teniendo esto y detectando la ubicación de este elemento, se procede a generar el código de movilidad del robot, definiendo inicialmente una función matemática, que el manipulador deberá seguir.

Figura 13: Representación real del objeto.



Fuente: Autor.

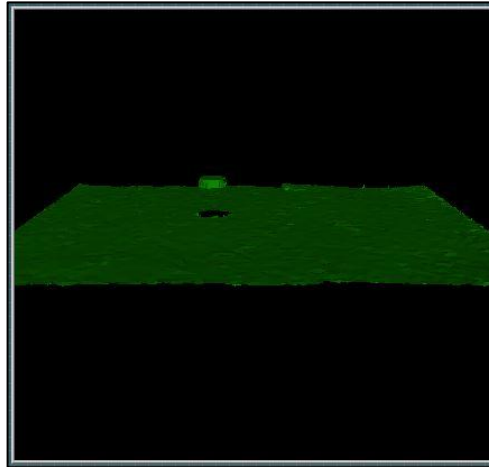
6.1 RESULTADO SOFTWARE.

6.1.1 RESULTADOS DE LA OBTENCIÓN Y MANIPULACIÓN DE LOS DATOS DE VISIÓN.

Para poder hacer la comparación entre el modelo entregado por Skanect y los vectores obtenidos, primero se visualizó este modelo en Labview, dando como resultado la Figura 14. En esta figura se puede observar la similitud entre el modelo real y el modelo virtual. Al sensar el entorno desde la parte superior se

puede ver la presencia de un objeto sobre la base, dando así el agujero que se observa.

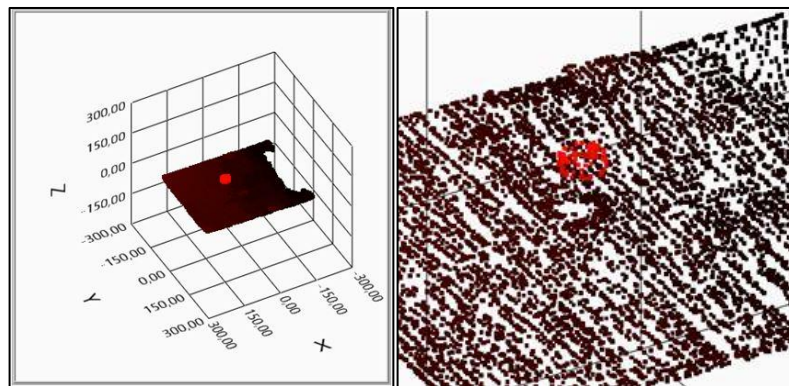
Figura 14: Representación de Objeto STL



Fuente: Autor.

Luego teniendo los vectores X, Y y Z, se graficaron de forma tridimensional dando como resultado una nube de puntos (Figura 15). Cada punto virtual obtenido hace referencia a un punto real de profundidad del entorno sensed. Para facilitar la visualización y entendimiento de esta gráfica, a medida que el punto este más cerca al sensor, su tonalidad de rojo será más clara.

Figura 15: Modelo representado en vectores XYZ.

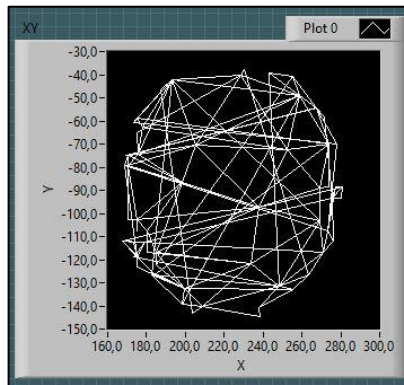


Fuente: Autor.

Teniendo el modelo ya almacenado en estos vectores, se pueden construir matrices con la información XY y XZ del objeto y graficarlos, esto necesario para el siguiente paso del algoritmo.

La figura 16 hace referencia al elemento detectado desde arriba, es decir, desde una vista superior, y se observa que tiende a mostrar la forma de un círculo, siendo esta la vista superior del cilindro.

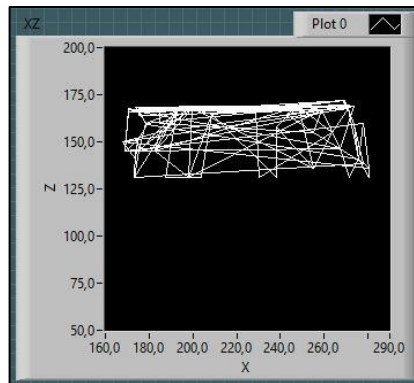
Figura 16: Plano XY.



Fuente: Autor.

La figura 17 muestra la vista lateral del cilindro, dando como resultado una figura rectangular.

Figura 17: Plano XZ



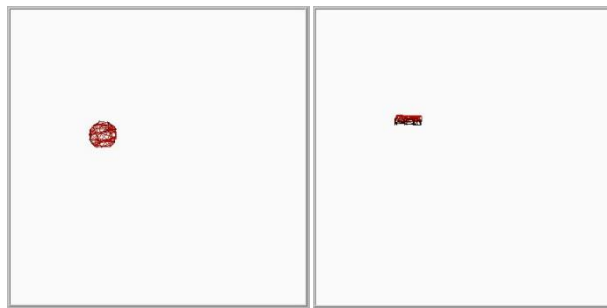
Fuente: Autor.

Las Figuras 16 y 17 son los planos necesarios para poder llevar a cabo el respectivo tratamiento de imágenes, ya que se puede intuir de que figura geométrica se trata.

6.1.2 TRATAMIENTO DE IMÁGENES.

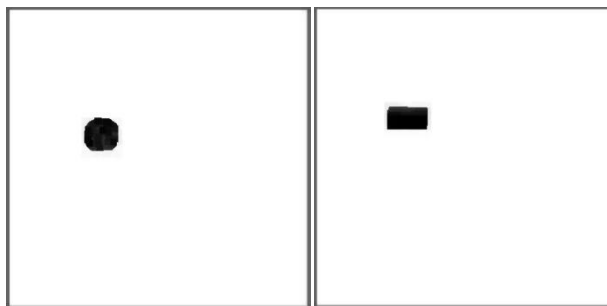
Después de convertir los planos anteriormente mostrados a imágenes (Figura 18) se aplica una operación de dilatación de imagen (Figura 19), esto con el fin de detectar de mejor manera la forma del elemento.

Figura 18: ImageTop e ImageSide antes del tratamiento de imágenes.



Fuente: Autor.

Figura 19: ImageTop e ImageSide después del tratamiento de imágenes.

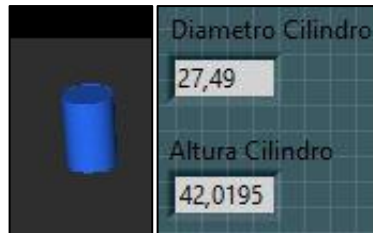


Fuente: Autor.

Analizando estos resultados, se puede decir que con el tratamiento de imágenes aplicado, es suficiente para poder detectar a qué tipo de figura geométrica pertenece el objeto sensado.

Con esta información definida, y al aplicar el árbol de decisión diseñado, se pueden detectar que figura geométrica es el objeto y que características posee este, esto con el fin de lograr una reconstrucción del entorno 3D con valores reales de sus dimensiones (Figura 20).

Figura 20: Entorno 3D reconstruido.



Fuente: Autor.

Para poder lograr una comparación del objeto reconstruido tridimensionalmente con el objeto real se construyó la Tabla 4.

Tabla 4: Comparación de dimensiones Real-Virtual.

	Dimensión Real (mm)	Dimensión Virtual (mm)	Error (%)
Altura Cilindro	40	42,0195	5,04875
Diámetro Cilindro	25	27,49	9,96

Fuente: Autor.

El error porcentual se encuentra por debajo del 10% lo que nos indica que la reconstrucción es fiel al objeto real, teniendo en cuenta que son dimensiones muy pequeñas.

Después de llevar a cabo varias mediciones se encontró una tolerancia promedio de 2 mm. Con esto se define que los elementos como mínimo deben tener dimensiones de 25 mm, esto para que la reconstrucción sea fiel al elemento real.

Es importante mencionar que existe una limitante en el sensado del entorno, dado por la versión demo del programa Skanect. Esta limitante consisten en que al ser

un demo del programa, este solo permite guardar hasta 5000 caras del entorno, lo cual es muy reducido, ya que un entorno normal tiene alrededor de 135000 caras.

De este modo, si se logra obtener la versión completa del programa Skanect, se disminuiría de gran forma la tolerancia mencionada anteriormente, haciendo el desarrollo mucho más exacto y fiel a la realidad.

6.1.3 RESULTADOS PLANEADOR DE TRAYECTORIA.

Para comprobar la optimización de la trayectoria planeada, se mostrarán los resultados de dos casos; el primero para una trayectoria de 5 número de puntos y el segundo para una trayectoria de 100 puntos; de esta forma, se podrán hacer los análisis respectivos.

Para ambos casos se definió una función matemática $f(x) = \sin(x)$, de orden 5, y con los mismos puntos iniciales y finales (Figura 21).

Figura 21: Características del Planeador de trayectorias.

The image shows a software interface for a trajectory planner. It has a blue background with a grid pattern. At the top, there are two input fields: 'Función' with the value 'sin(x)' and 'Orden' with the value '5'. Below these, there is a large rectangular box containing six input fields arranged in three rows and two columns. The first row contains 'X Inicial' (448) and 'X Final' (520). The second row contains 'Y Inicial' (-20) and 'Y Final' (-141,83). The third row contains 'Z Inicial' (619,76) and 'Z Final' (319,76). Each input field has a small icon to its left, resembling a coin or a button.

Función	Orden
sin(x)	5

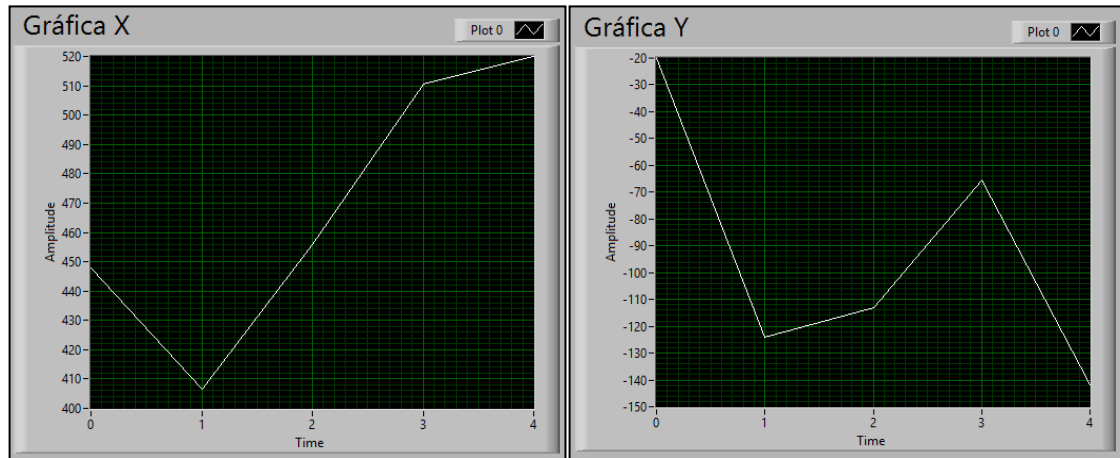
X Inicial	X Final
448	520
Y Inicial	Y Final
-20	-141,83
Z Inicial	Z Final
619,76	319,76

Fuente: Autor.

Caso 1: Numero de puntos = 5.

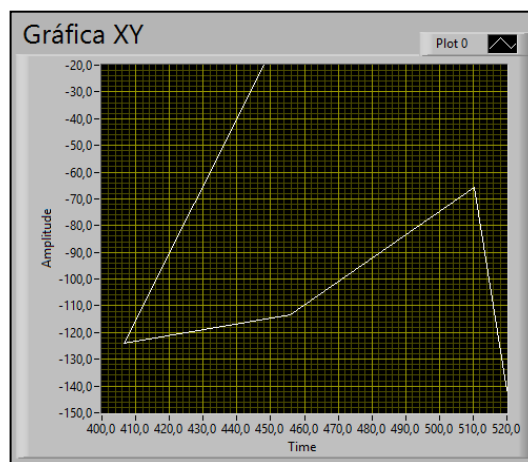
Para este caso la trayectoria en X y en Y que presentó el planeador corresponde a la Figuras 22.

Figura 22: Trayectoria X y Y, para 5 puntos.



Fuente: Autor.

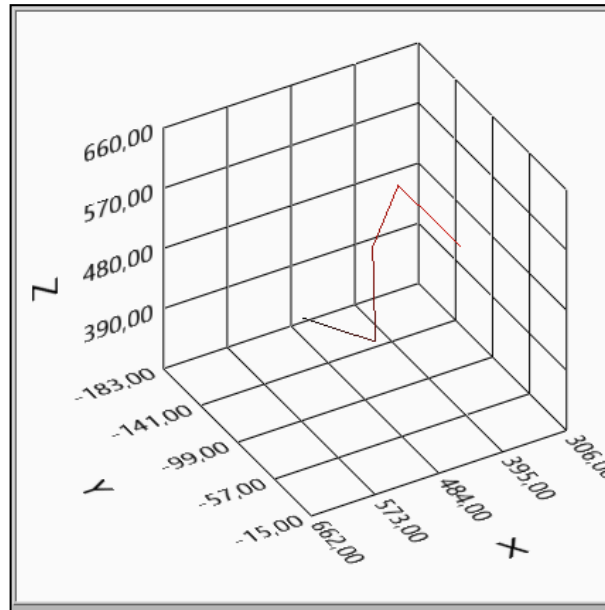
Figura 23: Gráfica XY - Trayectoria vista desde el Plano Z, para 5 puntos.



Fuente: Autor.

La figura 23, muestra la trayectoria del manipulador vista desde el plano Z, es decir desde la parte superior del manipulador.

Figura 24: Trayectoria tridimensional XYZ, para 5 puntos.



Fuente: Autor.

Puede analizarse que con 5 puntos, el recorrido no muestra un comportamiento senoidal, que es lo esperado, más bien tiende a ser muy lineal y brusco en sus cambios de dirección (Figura 24).

Por otro lado, el código generado del robot (Figura 25) posee alrededor de 10 líneas, lo que hace que el robot llegue al punto final más rápido debido a que tiene que cumplir menos instrucciones.

Figura 25: Código del Robot, para 5 puntos.

```
Código Robot

1 Mov P1
2 Mov P2
3 Mov P3
4 Mov P4
5 Mov P5

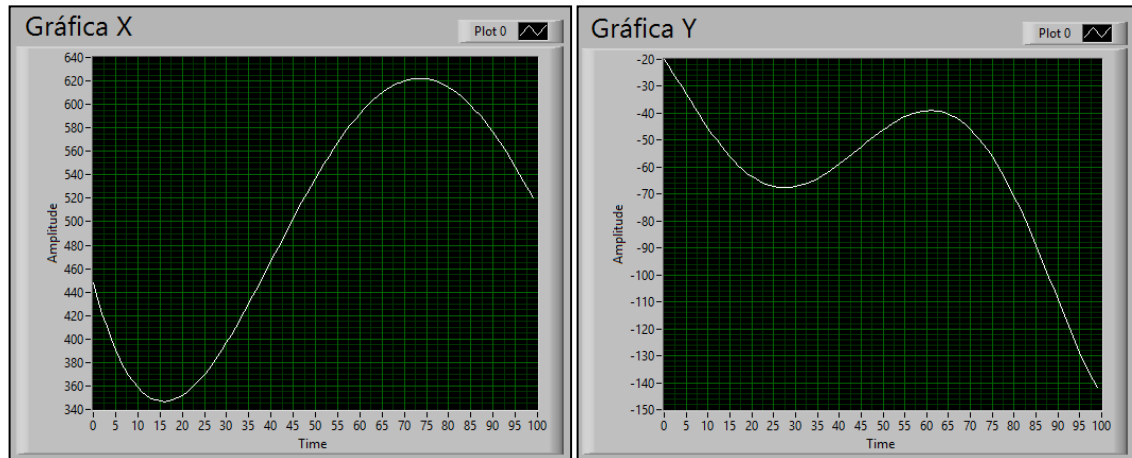
p1=(+448.00,-20.00,+619.76,-179.98,+0.04,-129.57)(7,0)
p2=(+406.45,-124.19,+559.76,-179.98,+0.04,-129.57)(7,0)
p3=(+455.79,-113.22,+499.76,-179.98,+0.04,-129.57)(7,0)
p4=(+510.42,-65.66,+439.76,-179.98,+0.04,-129.57)(7,0)
p5=(+520.00,-141.83,+379.76,-179.98,+0.04,-129.57)(7,0)
```

Fuente: Autor.

Caso 2: Numero de puntos = 100.

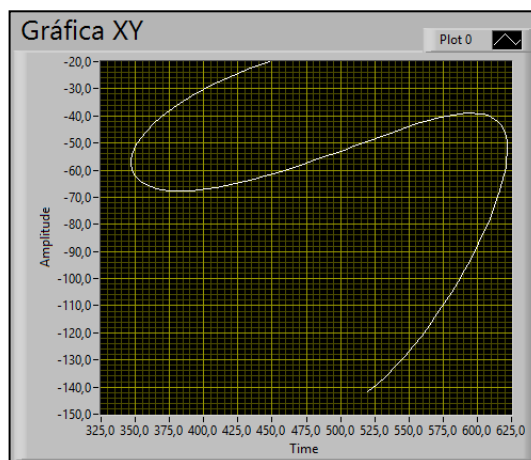
De igual forma que en la caso 1, la Figura 26 hace referencia a las trayectorias X y Y.

Figura 26: Trayectoria X y Y, para 100 puntos.



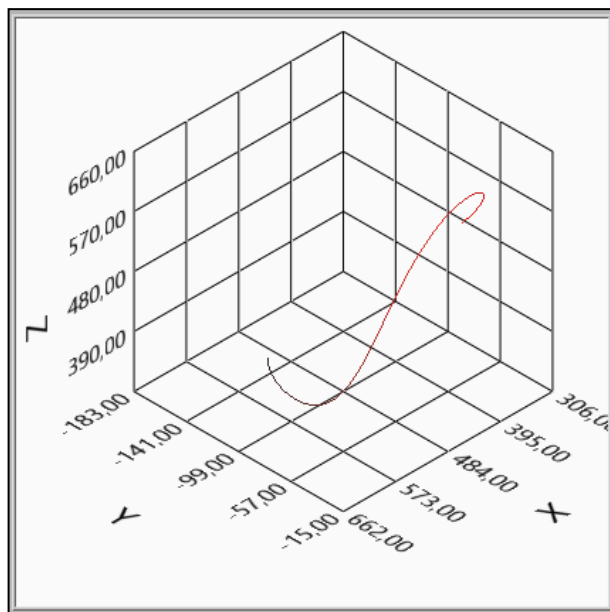
Fuente: Autor.

Figura 27: Gráfica XY - Trayectoria vista desde el Plano Z, para 100 puntos.



Fuente: Autor.

Figura 28: Trayectoria tridimensional XYZ, para 100 puntos.

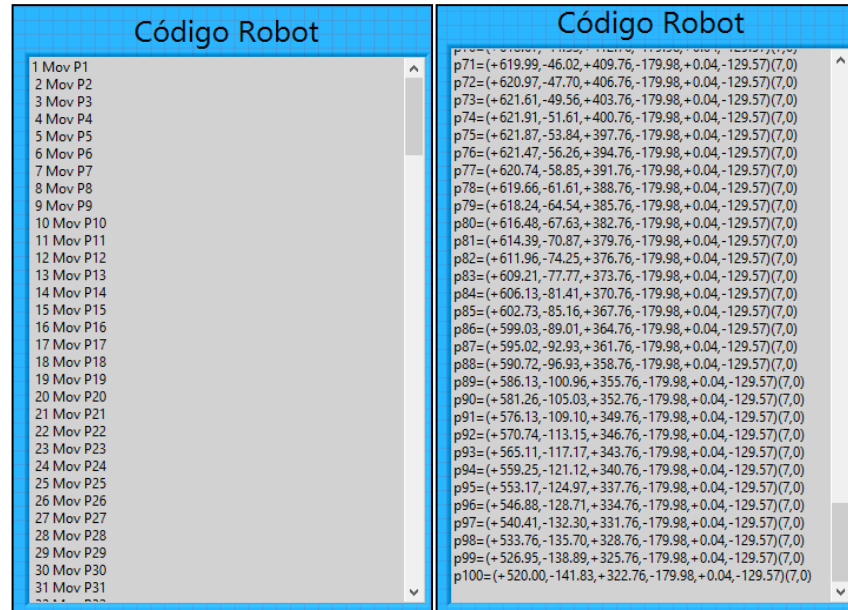


Fuente: Autor.

Finalmente se puede ver como a mayor cantidad de puntos, la trayectoria tiene a comportarse a la función matemática especificada, y tiende a ser mucho más suave o fluida (Figura 28).

Aunque al mismo tiempo el código generado para mover el robot tiene alrededor de 200 líneas en total, tanto de comandos como de posicionamiento (Figura 29); haciendo que el robot tenga un movimiento más fluido pero mucho más lento a causa de la gran cantidad de instrucciones a cumplir.

Figura 29: Código del Robot, para 100 puntos.



Código Robot	Código Robot
1 Mov P1	p71=(+619.99,-46.02,+409.76,-179.98,+0.04,-129.57)(7,0)
2 Mov P2	p72=(+620.97,-47.70,+406.76,-179.98,+0.04,-129.57)(7,0)
3 Mov P3	p73=(+621.61,-49.56,+403.76,-179.98,+0.04,-129.57)(7,0)
4 Mov P4	p74=(+621.91,-51.61,+400.76,-179.98,+0.04,-129.57)(7,0)
5 Mov P5	p75=(+621.87,-53.84,+397.76,-179.98,+0.04,-129.57)(7,0)
6 Mov P6	p76=(+621.47,-56.26,+394.76,-179.98,+0.04,-129.57)(7,0)
7 Mov P7	p77=(+620.74,-58.85,+391.76,-179.98,+0.04,-129.57)(7,0)
8 Mov P8	p78=(+619.66,-61.61,+388.76,-179.98,+0.04,-129.57)(7,0)
9 Mov P9	p79=(+618.24,-64.54,+385.76,-179.98,+0.04,-129.57)(7,0)
10 Mov P10	p80=(+616.48,-67.63,+382.76,-179.98,+0.04,-129.57)(7,0)
11 Mov P11	p81=(+614.39,-70.87,+379.76,-179.98,+0.04,-129.57)(7,0)
12 Mov P12	p82=(+611.96,-74.25,+376.76,-179.98,+0.04,-129.57)(7,0)
13 Mov P13	p83=(+609.21,-77.77,+373.76,-179.98,+0.04,-129.57)(7,0)
14 Mov P14	p84=(+606.13,-81.41,+370.76,-179.98,+0.04,-129.57)(7,0)
15 Mov P15	p85=(+602.73,-85.16,+367.76,-179.98,+0.04,-129.57)(7,0)
16 Mov P16	p86=(+599.03,-89.01,+364.76,-179.98,+0.04,-129.57)(7,0)
17 Mov P17	p87=(+595.02,-92.93,+361.76,-179.98,+0.04,-129.57)(7,0)
18 Mov P18	p88=(+590.72,-96.93,+358.76,-179.98,+0.04,-129.57)(7,0)
19 Mov P19	p89=(+586.13,-100.96,+355.76,-179.98,+0.04,-129.57)(7,0)
20 Mov P20	p90=(+581.26,-105.03,+352.76,-179.98,+0.04,-129.57)(7,0)
21 Mov P21	p91=(+576.13,-109.10,+349.76,-179.98,+0.04,-129.57)(7,0)
22 Mov P22	p92=(+570.74,-113.15,+346.76,-179.98,+0.04,-129.57)(7,0)
23 Mov P23	p93=(+565.11,-117.17,+343.76,-179.98,+0.04,-129.57)(7,0)
24 Mov P24	p94=(+559.25,-121.12,+340.76,-179.98,+0.04,-129.57)(7,0)
25 Mov P25	p95=(+553.17,-124.97,+337.76,-179.98,+0.04,-129.57)(7,0)
26 Mov P26	p96=(+546.88,-128.71,+334.76,-179.98,+0.04,-129.57)(7,0)
27 Mov P27	p97=(+540.41,-132.30,+331.76,-179.98,+0.04,-129.57)(7,0)
28 Mov P28	p98=(+533.76,-135.70,+328.76,-179.98,+0.04,-129.57)(7,0)
29 Mov P29	p99=(+526.95,-138.89,+325.76,-179.98,+0.04,-129.57)(7,0)
30 Mov P30	p100=(+520.00,-141.83,+322.76,-179.98,+0.04,-129.57)(7,0)
31 Mov P31	

Fuente: Autor.

Todo el desarrollo mostrado se diferencia de otros autores y de otros trabajos en varios aspectos:

El dispositivo de visión utilizado no se había usado en integración a un robot manipulador Mitsubishi, lo cual abre una gran cantidad de aplicaciones en el área de la robótica y visión artificial.

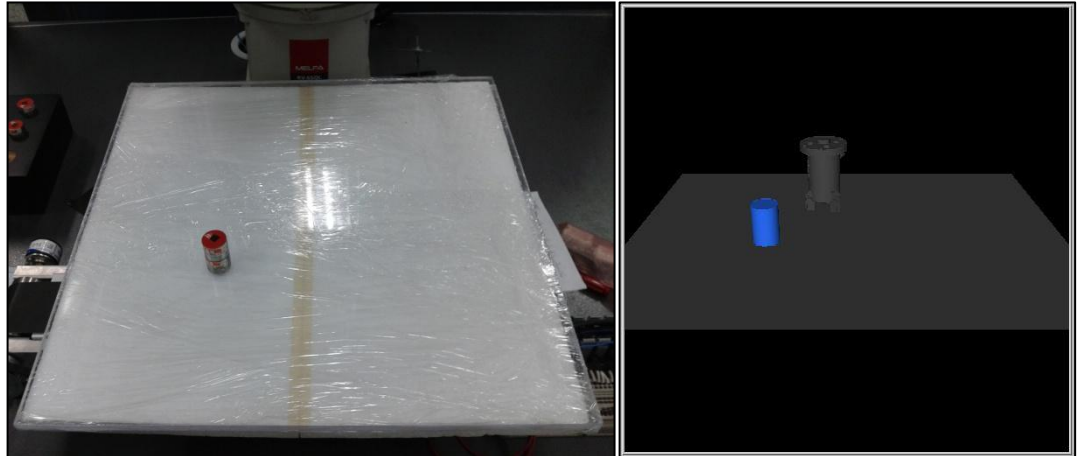
De la misma manera la creación de un planeador de trayectorias para los robots Mitsubishi, es algo novedoso y que puede ser tomado por otros autores para hacerle mejoras o simplemente basarse en este desarrollo.

La reconstrucción de objetos y espacios 3D, ya se había tratado en otros trabajos pero con otros dispositivos como el Kinect y en lenguaje C. Ahora teniendo la posibilidad de generar modelos en formato STL y manipularlos de tal manera que se tenga este mismo formato pero en vectores, permite la aparición de nuevos desarrollos debido a la infinidad de tratamientos matemáticos existentes.

Como análisis final, al ser el trabajo de grado la integración de dos dispositivos novedosos y que no se han trabajado a profundidad, abre la posibilidad a futuros autores de seguir con la investigación y mejorarla o como idea o estado del arte para nuevos desarrollos.

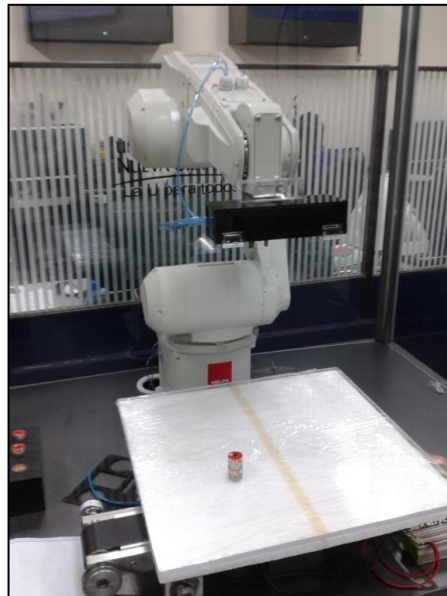
6.1.4 IMÁGENES DEL DESARROLLO DEL SISTEMA INTEGRADO:

Figura 30: Comparación entorno real con reconstrucción.



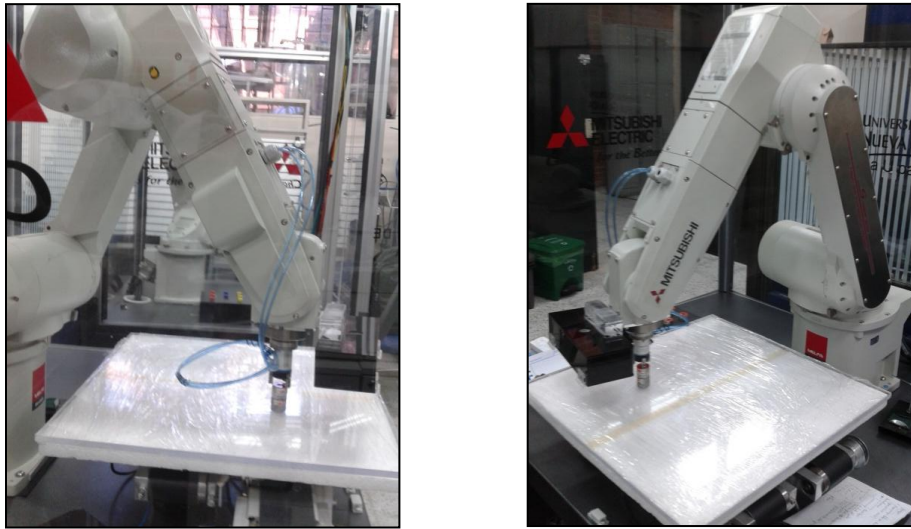
Fuente: Autor.

Figura 31: Posición inicial de sensado 3D.



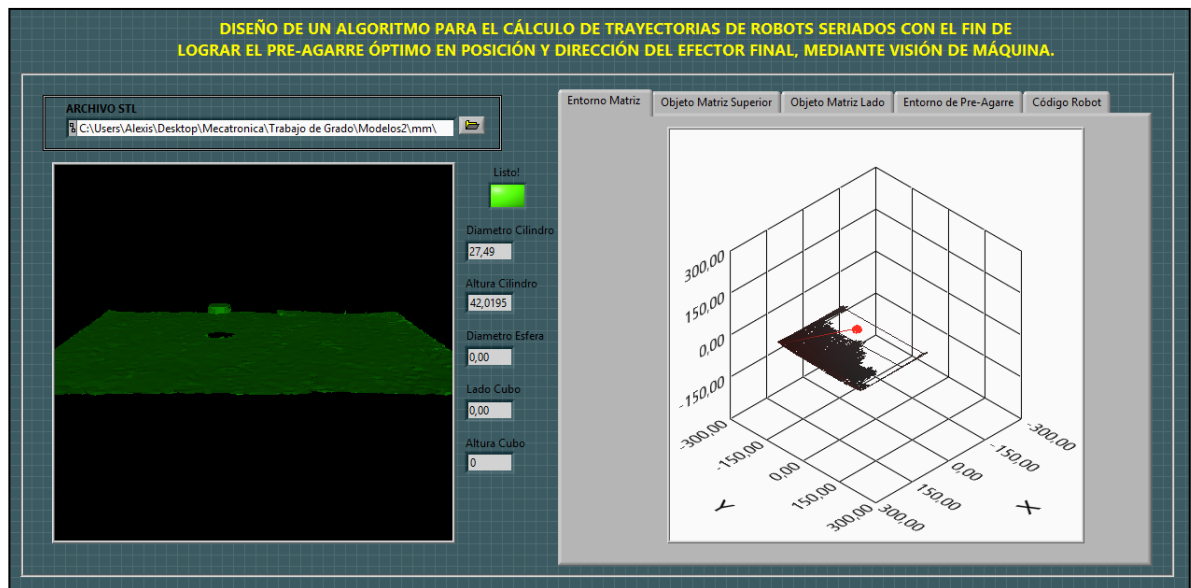
Fuente: Autor.

Figura 32: Posición final de pre-agarre al objeto.



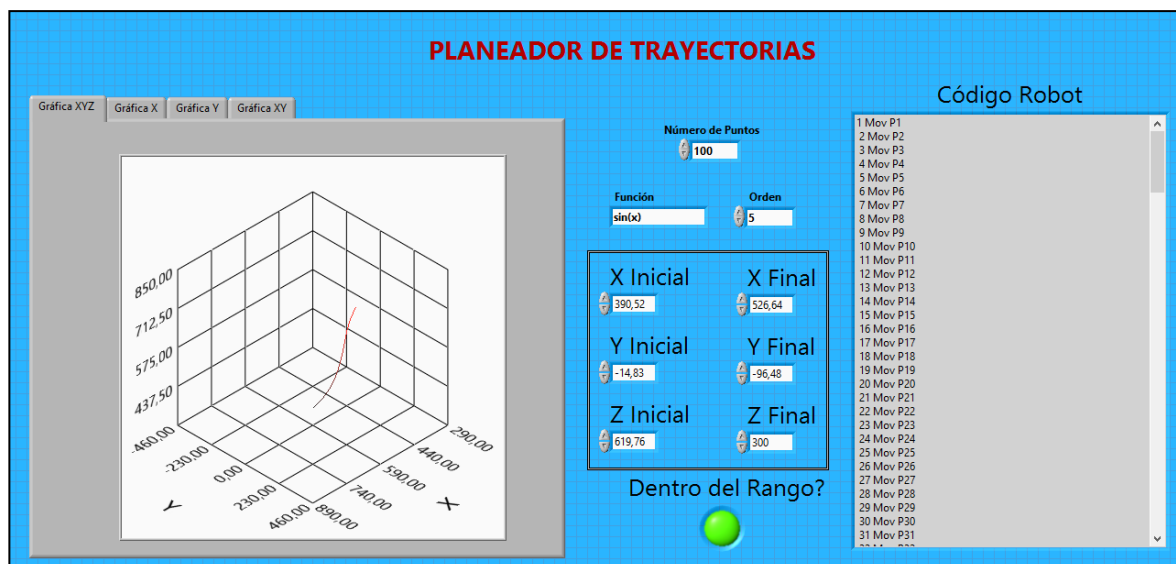
Fuente: Autor.

Figura 33: Interfaz gráfica programa principal.



Fuente: Autor.

Figura 34: Interfaz gráfica programa planeador de trayectorias.



Fuente: Autor.

7. CONCLUSIONES

Después de hacer el respectivo análisis de las posibles soluciones al problema planteado y considerando todos los dispositivos de visión útiles para la aplicación deseada, se concluyó mediante una matriz de decisión, la solución más viable y adecuado para el desarrollo. Un dispositivo de bajo costo, fácil manejo y fácil alimentación, como lo es el Sensor PrimeSense Carmine 1.09, es una buena opción para reemplazar el uso del Kinect, que aunque posee librerías para usarlo, presenta desventajas respecto al primero.

El programa SKANECT facilitó el sensado del entorno real, debido a su fácil acople y compatibilidad con el dispositivo Carmine, además su software de demostración es gratuito, de muy fácil manejo y con muchas opciones de manipulación de datos que para otras aplicaciones o desarrollos, tendrían validez.

Mediante el correcto tratamiento de imágenes, se logró determinar características físicas de objetos reales como diámetros de cilindros, diámetros de esferas, altura de cilindros y lados de cubos, de manera fiable justificado en el error menor al 10% encontrado, comparando el elemento real con el reconstruido. Además de la utilización del árbol de decisión por el método de inferencia que facilitó la escogencia de la opción más viable y fiel.

Los distintos módulos o complementos de LABVIEW, ayudan a la solución de problemas y desarrollo de aplicaciones; para este caso el módulo de tratamiento de imágenes evitó extensos procesos de programación que tienden a reducir el tiempo de ejecución de programas y aumentar la probabilidad de errores.

Se decidió utilizar el robot Mitsubishi RV-6SQL ya que es el manipulador presente en el laboratorio de automatización de la Universidad Militar Nueva Granada, esto facilitó tanto la manipulación de este y como la integración de dispositivos externos, algo que no se podría haber logrado con manipuladores no académicos.

Utilizando la teoría de aproximación Chevyshev como técnica de optimización, se logró construir un planeador de trayectorias para el robot Mitsubishi RV-6SQL. Este planeador posee distintos parámetros, esto con el fin de que el usuario los ajuste dependiendo de la trayectoria que este desee que realice el robot.

Fue necesario conocer el alcance máximo del manipulador en sus 3 ejes, de este modo si algún punto de la trayectoria se encontrara fuera de este espacio de

trabajo, no se generaba el código de movimiento del robot, esto para seguridad del robot con el entorno, y para evitar la aparición de alarmas que indiquen algún punto inalcanzable para el robot.

La cinemática inversa del robot Mitsubishi, fue de gran ayuda para la construcción del planeador de trayectorias y lograr el movimiento del robot manipulador Mitsubishi, debido a que solo era necesario definir puntos X, Y y Z del espacio real, mas no mover cada articulación por separado para llegar al mismo punto.

8. RECOMENDACIONES Y TRABAJO FUTURO

Para el programa se recomienda que el momento de ejecutarlo, se utilicen objetos de dimensiones como mínimo de 25 mm, esto a razón que entre el objeto sea más grande, el programa tiene a detectarlo mejor, si los objetos llegan a ser muy pequeños, en un punto no se podrá diferenciar si el elemento es un círculo o un cuadrado.

Cuando se graben los modelos desde el SKANECT, es importante guardarlos como extensión STL y en la opción de SCALE, seleccionar la opción en milímetros.

En el desarrollo del proyecto de grado, se utilizó la versión demo del software SKANECT, el cual posee una limitante, después de haber obtenido el entorno 3D, al momento de guardarlo, solo permite exportarlo reduciendo este modelo a 5000 caras.

Sería de gran utilidad que la Universidad Militar Nueva Granada pudiera obtener la versión completa de este software, que aunque posee un costo, se podrían desarrollar grandes aplicaciones, además teniendo en las instalaciones una máquina de prototipado rápido.

Esta versión completa deja guardar el modelo con una cantidad de caras sin limitantes y permite grabaciones consecutivas de un mismo entorno mediante varias caras; construyendo así una reconstrucción 3d de 360 grados de un entorno real.

A manera de trabajo futuro y continuación del trabajo de grado, se tiene:

Para completar el desarrollo de este trabajo de grado, se podría pensar en manipular las variables A, B y C del código del robot, que hacen referencia a la orientación del gripper.

Uniendo lo mencionado anteriormente, podrían hacerse reconstrucciones de no solo objetos de geometría básica, sino de elementos amorfos y que el robot tenga la posibilidad de decidir como sujetarlo de la mejor manera.

Ya que se logró el pre-agarre al objeto, se tiene un proyecto de investigación liderado por el Ing. Mauricio Mauledoux Ph.D. y el Ing. Vladimir Prada que consiste en la obtención de puntos óptimos de agarre para un manipulador con efector final de tres dedos, lo cual complementaria el desarrollo mostrado en este trabajo de

grado. El objetivo de este proyecto de investigación es desarrollar una estrategia de búsqueda de puntos óptimos de agarre para un manipulador con un efector final de tres dedos, capaz de obtener una solución en un tiempo coherente con su aplicación.

9. BIBLIOGRAFÍA

1. DESARROLLO DE APLICACIONES INTEGRANDO ROBÓTICA Y VISIÓN EN UN ROBOT INDUSTRIAL KUKA PARA DEMOSTRAR SUS CAPACIDADES. Proyecto de final de carrera de Pablo Muñoz Rodríguez de la Universidad Politécnica de Valencia presentado en Septiembre del 2011.
2. DISEÑO DE UN AMBIENTE VIRTUAL MEDIANTE VISIÓN ARTIFICIAL PARA APLICACIONES EN TELEROBÓTICA. Paper desarrollado por Jaime Masiá, Juan Ramón Rufino, Francisco Javier Tormo, José Ferri y Borja Mansilla en la Universidad Politécnica de Valencia, Campus de Alcoi.
3. DISEÑO Y CONSTRUCCIÓN DE UN ROBOT MANIPULADOR TIPO SCARA CON SISTEMA DE VISIÓN DE MÁQUINA, QUE PUEDA SER UTILIZADO COMO HERRAMIENTA DIDÁCTICA EN ASIGNATURAS DE ROBÓTICA Y CONTROL. Trabajo de grado del Ingeniero César Eduardo Jiménez Robayo de la Universidad Militar Nueva Granada presentado en el año 2009.
4. GENERACION DE TRAYECTORIAS PARA UN ROBOT MANIPULADOR UTILIZANDO PROCESAMIENTO DE IMAGENES Y SPLINES. Paper desarrollado por J. A. Soto Cajiga, estudiante de posgrado Centro de ingeniería y desarrollo industrial, J. E. Vargas Soto, Director facultad de ingeniería Universidad Anahuac México sur y J. C. Pedraza Ortega, Investigador Centro de ingeniería y desarrollo industrial.
5. RECONSTRUCCIÓN 3D DE MODELOS UTILIZANDO TÉCNICAS DE VISIÓN ARTIFICIAL. Por Karen Vilá Ubieto y sus directores Álvaro Arranz Domingo , Manuel Alvar Miró y Álvaro Sánchez Miralles. Universidad Pontificia Comillas Madrid, 2009.
6. RECONSTRUCCIÓN TRIDIMENSIONAL A PARTIR DE VARIAS IMÁGENES. Por Dr. Luis Gerardo de la Fraga. Sección de Computación Departamento de Ingeniería Eléctrica CINVESTAV. Noviembre, 2005.
7. HERRAMIENTA SOFTWARE PARA RECONSTRUCCIÓN DIGITAL 3D Y VISUALIZACIÓN DE SUPERFICIES. Por Johan Alberto Arzuza Narváez. Centro de investigaciones y desarrollo – Facultad de ingeniería Universidad de Manizales. 2005
8. RECONSTRUCCIÓN 3D: DE LA ADQUISICIÓN A LA VISUALIZACIÓN. Por Federico Cristina, Sebastián H. Dapoto, Javier Vegas, Verónica Artola, Claudia Russo, María José Abásolo. Instituto de Investigación en Informática LIDI Facultad de Informática. Universidad Nacional de La Plata.
9. ALGORITMO DE RECONSTRUCCIÓN DE SUPERFICIES 3D PARA MODELAJE ANTROPOMÉTRICO. O. Ruiz Salguero; R. Schrader Gil; J. Acosta Duarte. Centro Interdisciplinario de Investigación (CII) en CAD / CAM / CG. Universidad EAFIT en Medellín.
10. KINECTFUSION: REAL-TIME 3D RECONSTRUCTION AND INTERACTION USING A MOVING DEPTH CAMERA. Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, Andrew Fitzgibbon. Desarrollo por Microsoft Research Cambridge UK, Imperial College London UK, Newcastle University UK, Lancaster University UK y University of Toronto Canada.

11. REAL-TIME CAMERA TRACKING AND 3D RECONSTRUCTION USING SIGNED DISTANCE FUNCTIONS. Erik Bylow, Jurgen Sturm, Christian Kerl, Fredrik Kahl and Daniel Cremers. Center for Mathematical Sciences, Lund University, Lund, Sweden and Department of Computer Science, Technical University of Munich, Garching, Germany.
12. REAL-TIME 3D RECONSTRUCTION IN DYNAMIC SCENES USING POINT-BASED FUSION. Por Maik Keller, Damien Lefloch, Martin Lambers, Shahram Izadi, Tim Weyrich y Andreas Kolb. PMDtechnologies, University of Siegen University of Siegen, Microsoft Research, University College London, University of Siegen.
13. METHODS FOR 3D RECONSTRUCTION FROM MULTIPLE IMAGES. Por Sylvain Paris. MIT CSAIL.
14. 3D RECONSTRUCTION USING LABELED IMAGE REGIONS. Por Remo Ziegler, Wojciech Matusik, Hanspeter Pfister and Leonard McMillan. Mitsubishi Electric Research Labs, Cambridge, Massachusetts USA, Laboratory for Computer Science, MIT, Cambridge, Massachusetts USA, University of North Carolina at Chapel Hill, Chapel Hill, North Carolina USA. 2003.
15. SELF-CALIBRATION AND METRIC 3D RECONSTRUCTION FROM UNCALIBRATED IMAGE SEQUENCES. Por Marc Pollefev. Katholieke Universiteit Leuven. Faculteit Toegepaste Wetenschappen. Departement Esat. 1999.
16. VISUAL-GEOMETRIC 3-D SCENE RECONSTRUCTION FROM UNCALIBRATED IMAGE SEQUENCES. Por Reinhard Koch and Jan-Michael Frahm. Multimedia Information Processing Group Christian-Albrechts-University of Kiel, Germany. 2001.
17. 3D FACE RECONSTRUCTION FROM A SINGLE IMAGE USING A SINGLE REFERENCE FACE SHAPE. Por Ira Kemelmacher-Shlizerman, Ronen Basri, Member, IEEE.
18. 3D FACE RECONSTRUCTION USING A SINGLE OR MULTIPLE VIEWS. Por Jongmoo Choi, Gerard Medioni, Yuping Lin. University of Southern California - USA. 2010.
19. STEREO RECONSTRUCTION FROM MULTIPERSPECTIVE PANORAMAS. Por Yin Li, Heung-Yeung Shum, Senior Member, IEEE, Chi-Keung Tang, Member, IEEE Computer Society, and Richard Szeliski, Senior Member, IEEE. Enero 2004.
20. EFFICIENT VISUAL HULL COMPUTATION FOR REAL-TIME 3D RECONSTRUCTION USING CUDA. Por Alexander Ladikos Selim Benhimane Nassir Navab. Department of Computer Science, Technische Universitat Munchen Boltzmannstr. Germany. 2008.
21. 3D SIMULTANEOUS LOCALIZATION ANDMODELING FROM STEREO VISION. Por Miguel Angel Garcia and Agusti Solanas. Intelligent Robotics and Computer Vision Group / Dept. of Computer Science and Mathematics Rovira i Virgili University. 2004.
22. PASSIVE AND ACTIVE STEREO VISION FOR SMOOTH SURFACE DETECTION OF DEFORMED PLATES. Por Chichyang Chen and Yuan F. Zheng, Senior Member, IEEE. Junio 1995.
23. SOFTWARE DE SIMULACIÓN DE UN ROBOT MANIPULADOR MARCA MITSUBISHI RV-2AJ. Por Moreno Trejo Roberto, Cabrera Ríos Juan Salvador, Hernández Ramírez Fernando, José Emilio Vargas Soto y Jesús Carlos Pedraza Ortega. Universidad Tecnológica de San Juan del Río, UTSJR, Universidad Anáhuac – México Sur y Centro de Ingeniería y Desarrollo Industrial. Noviembre 2005.

24. COMPUTATIONAL STEREO VISION USING COLOR. Por John R. Jordan III and Alan C. Bovik. IEEE Control Systems Magazine. Junio 1988.
25. APPLICATION OF DISTANCE MEASURING WITH MATLAB/SIMULINK. Por Mircea Coman, Sergiu-Dan Stan, Milos Manic, Radu Balan. Dept. of Mechatronics, Technical University of Cluj-Napoca, Cluj-Napoca, Romania, University of Idaho, Department of Computer Science. Mayo 2010.
26. GENERALIZING THE HOUGH TRANSFORM TO DETECT ARBITRARY SHAPES. Por D. H. BALLARD. Computer Science Department, University of Rochester, Rochester, NY 14627, U.S.A. Septiembre 1980.
27. GEOMETRIC SHAPE DETECTION WITH SOUNDVIEW. Por K. van den Doel, D. Smilek, A. Bodnar, C. Chita, R. Corbett, D. Nekrasovski, J. McGrenere. Department of Computer Science, Department of Psychology, University of British Columbia, Vancouver, Canada. Julio 2004.
28. NEAR-OPTIMAL DETECTION OF GEOMETRIC OBJECTS BY FAST MULTISCALE METHODS. Por Ery Arias-Castro, David L. Donoho, Member, IEEE, and Xiaoming Huo, Senior Member, IEEE. Julio 2005.
29. DISTANCE SETS FOR SHAPE FILTERS AND SHAPE RECOGNITION. Por Cosmin Grigorescu, Student Member, IEEE, and Nicolai Petkov. Octubre 2003.
30. ROBOTICA. MANIPULADORES Y ROBOTS MOVILES. Autor: Aníbal Ollero Baturone
31. ROBOTICA. Autor: John J. Craig. Tercera Edición. Editorial: Pearson Prentice Hall.
32. AVANCES EN ROBOTICA Y VISION POR COMPUTADOR. Autor: José Andrés Somolinos Sánchez. Ediciones de la Universidad de Castilla – La Mancha.
33. MACHINE VISION. Autor: Wesley E. Snyder Hairong Qi
34. MACHINE VISION: THEORY, ALGORITHMS, PRACTICALITIES. Escrito por E. R. Davies. Tercera Edición. Editorial: Elsevier.
35. HANDBOOK OF MACHINE VISION. Editado por Alexander Hornberg.
36. DISEÑO EN INGENIERIA MECANICA. Por Joseph Edward Shigley, Larry D. Mitchell.
37. MECÁNICA DE MATERIALES. Por R.C. Hibbeler. Sexta Edición. Editorial: Pearson Prentice Hall.
38. VISION POR COMPUTADOR. Por José Francisco Vélez Serrano, Ana Belén Moreno Díaz, Ángel Sánchez Calle, José Luis Esteban Sánchez-Marín. Segunda Edición.
39. VISION ARTIFICIAL: RASGOS DESCRIPTIVOS PARA EL RECONOCIMIENTO DE OBJETOS . Por Juan Humberto Sossa Azuela. Editorial: Ra-Ma.
40. VISION POR COMPUTADOR. Por Arturo de la Escalera Hueso. Editorial Pearson Alhambra. Primera Edición.

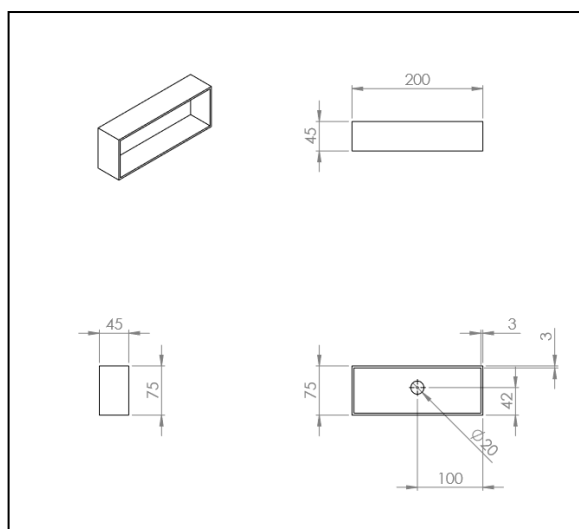
41. ROBÓTICA: Control, Detección, Visión e Inteligencia. K.S. Fu, R.C. Gonzalez, C.S.G. Lee. McGraw-Hill.
42. UNA INTRODUCCIÓN A LA VISIÓN DE MÁQUINA. Restrepo Arias Luis Guillermo. 1999-2007.
43. ROBÓTICA Y VISIÓN ARTIFICIAL. Quira Nelson. Rivera Carlos. Corporación Universitaria Autónoma del Cauca.
44. INTRODUCCIÓN A LA ROBÓTICA. Cristina Urdiales García. Universidad de Málaga – Escuela Técnica Superior de Ingenieros de Telecomunicaciones.
45. LA TELEROBOTICA Y SUS APLICACIONES EN LA MEDICINA Y LA EXPLORACION ESPACIAL. Por Luis G. De Sousa P.

10. ANEXOS

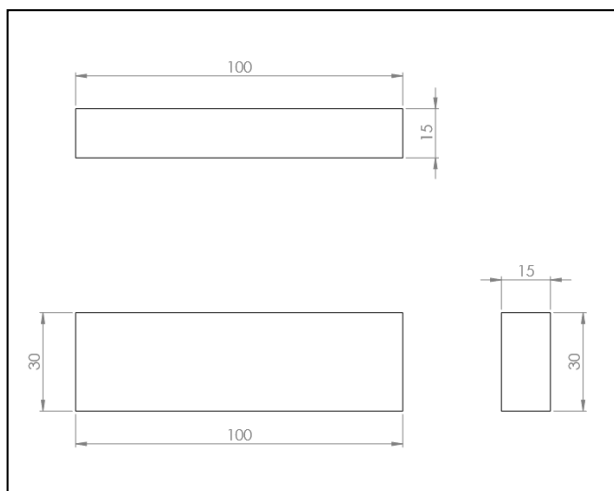
10.1 PLANOS

Planos caja del sensor PrimeSense Carmine 1.09:

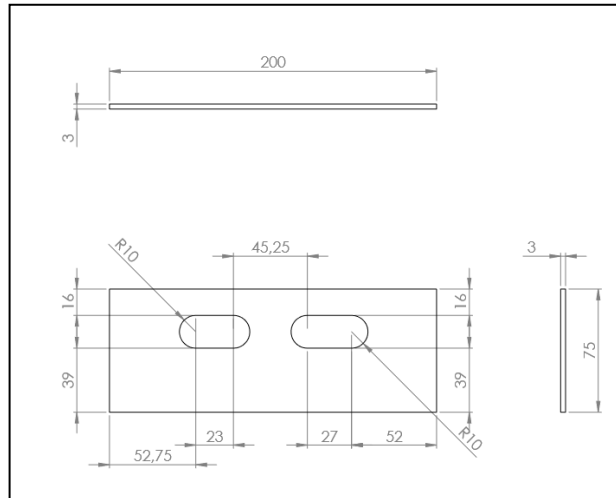
Caja:



Soporte:

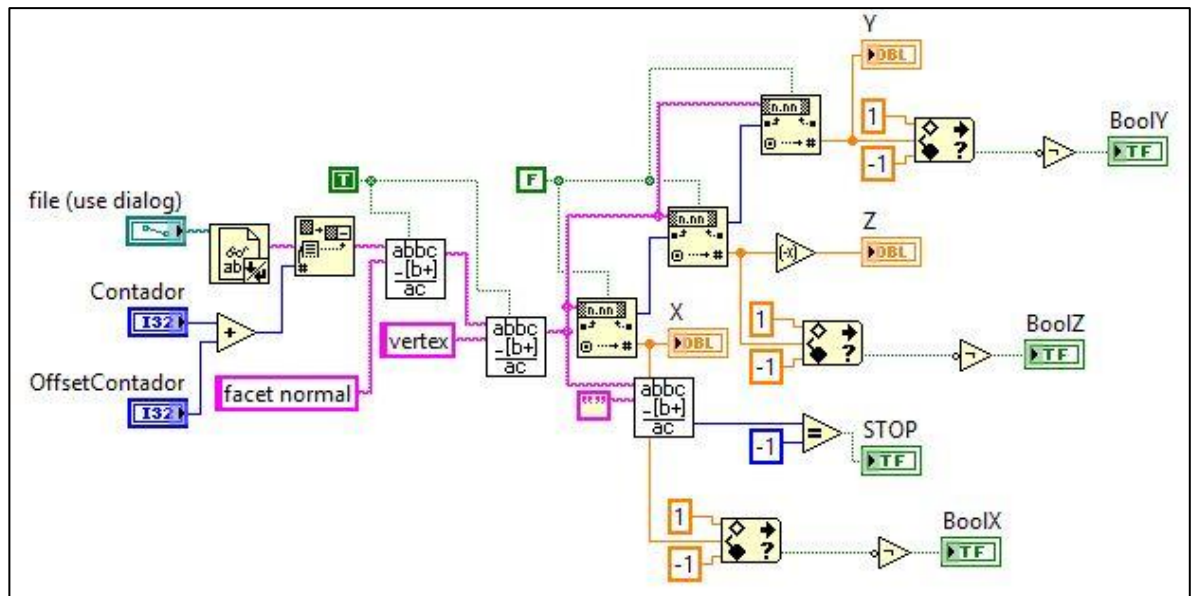


Tapa:



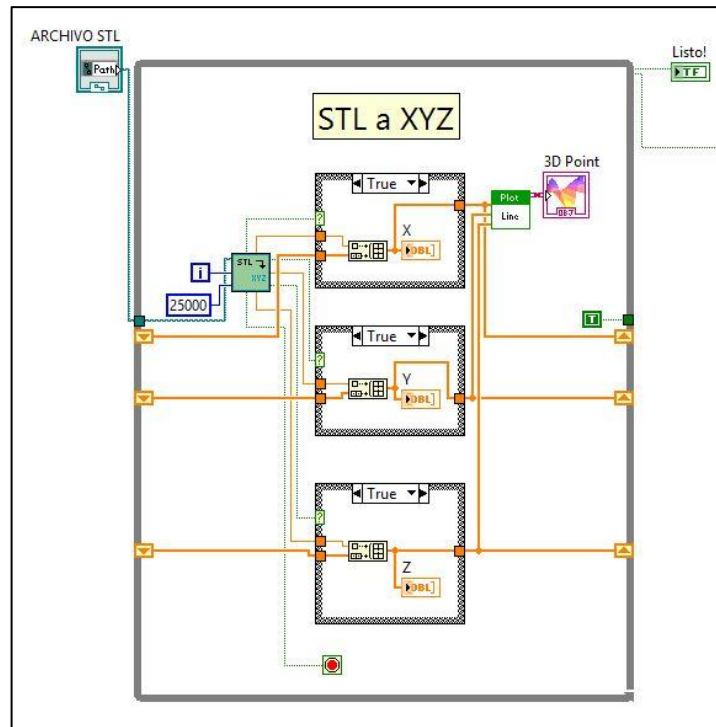
10.2 DIAGRAMAS DE BLOQUES:

Programación del bloque STLtoXYZ



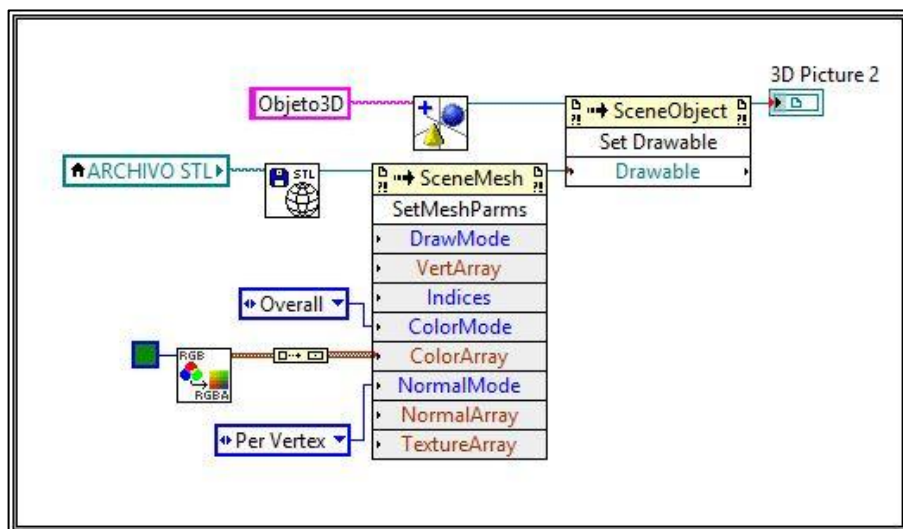
Fuente: Autor.

Programa principal - Vectores XYZ.



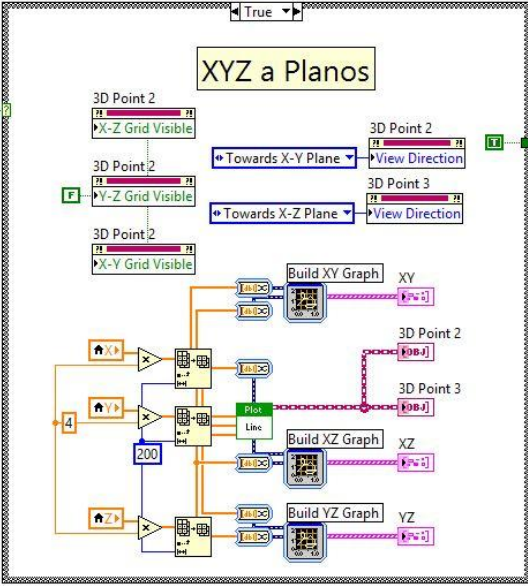
Fuente: Autor.

Programa visualizador archivo STL.



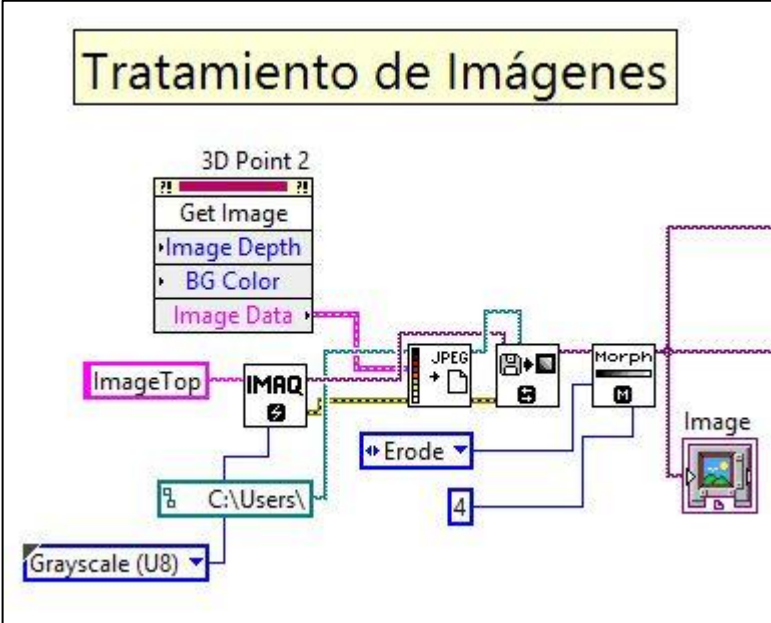
Fuente: Autor.

Programa Vectores XYZ a Planos



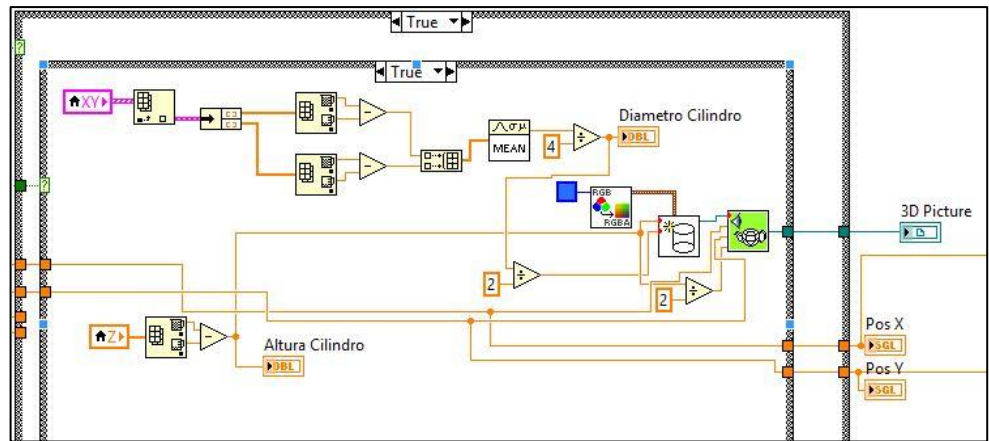
Fuente: Autor

Conversión de Plano a Imagen



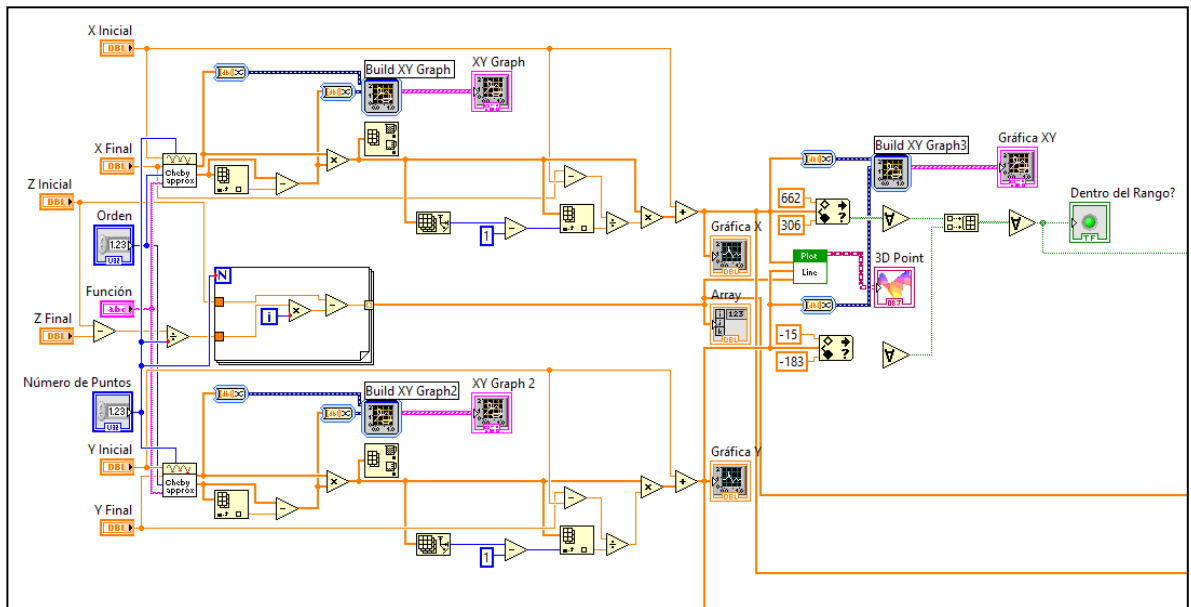
Fuente: Autor

Programa entorno 3D.



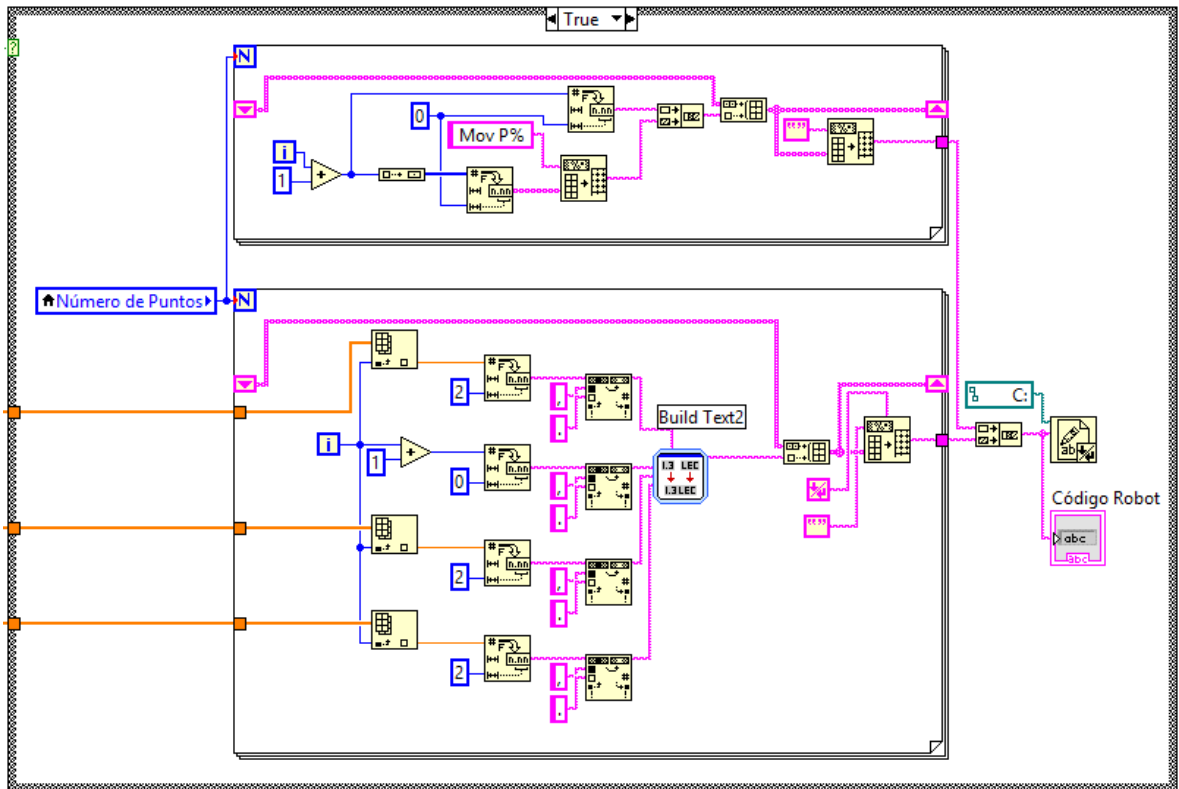
Fuente: Autor.

Planeador de trayectorias.



Fuente: Autor.

Creación código del robot.



Fuente: Autor.