



ESCUELA POLITÉCNICA



UNIVERSIDAD DE EXTREMADURA

Escuela Politécnica

Grado en Ingeniería Informática en Ingeniería de
Computadores

Trabajo Fin de Grado

Modelado activo de objeto en robots móviles
mediante visión



ESCUELA POLITÉCNICA



UNIVERSIDAD DE EXTREMADURA

Escuela Politécnica

Grado en Ingeniería Informática en Ingeniería de
Computadores

Trabajo Fin de Grado

Modelado activo de objeto en robots móviles
mediante visión

Autor: José Manuel Parra Martín

Tutor: Pilar Bachiller Burgos

ÍNDICE GENERAL DE CONTENIDOS

Contenido

1	INTRODUCCIÓN.....	10
2	OBJETIVOS.....	13
3	ESTADO DEL ARTE	14
3.1	Modelado de objetos utilizando planos	15
3.1.1	INICIALIZACIÓN DEL MODELO PLANO.....	16
3.1.2	EXTRACCION DE PLANOS DE LA RECONSTRUCCIÓN.....	17
3.1.3	OPTIMIZACIÓN EL MODELO PLANO	18
3.2	OBJETOS BASADOS EN EXTRUSIONES	19
4	MATERIAL Y MÉTODO.....	21
4.1	HERRAMIENTAS	21
4.1.1	KDevelop.....	21
4.1.2	QTDesigner	21
4.1.3	Robocomp.....	21
4.1.4	OpenCV	23
4.2	PROPUESTA DE MODELADO ACTIVO	23
4.2.1	DESCRIPCIÓN GENERAL DE LA PROPUESTA.....	23
4.2.2	PRIMITIVAS VISUALES	24
4.2.3	CALCULO PRELIMINAR DE LOS DATOS.....	29
4.2.4	DETECTOR DE ESQUINAS	29
4.2.5	MÉTODOS DE BÚSQUEDA DE ESQUINA DEL MODELO.....	31
4.2.6	MÁQUINA DE ESTADOS	34
5	RESULTADOS	44
5.1	ESCENARIOS.....	44
5.2	PROBLEMAS, SOLUCIONES Y RESULTADOS.....	48
5.2.1	Generación del modelo.....	48

5.2.2	Reorientación de la cámara durante el modelado	55
5.2.3	Mantenimiento del modelo.....	62
5.2.4	Problema de los ángulos obtusos en Harris	65
6	CONCLUSIONES.....	67

ÍNDICE DE TABLAS

Tabla 1	Tabla de tiempos para los 5 primeros escenarios.	49
Tabla 2	Se añade la búsqueda de esquinas con Hough.....	50
Tabla 3	Se optimiza el análisis de puntos en la búsqueda con dirección.	51
Tabla 4	Tiempos después de corregir el cálculo de direcciones.....	52
Tabla 5	Tiempos después de optimizar el cálculo de direcciones.....	52
Tabla 6	Priorizando esquinas de partida.....	53

ÍNDICE DE FIGURAS

Figura 1 Vistas para reconstruir el modelo 3D de un objeto. [3]	10
Figura 2 Cuatro nuevas vistas generada a partir del modelo 3D de la figura 1. [3]	11
Figura 3 Correspondencias obtenidas de F estimadas directamente a partir de los datos (izquierda) y F estimadas a partir de un plano dominante más el paralaje (derecha). Las correspondencias obtenidas a través del plano más el paralaje son más fiables, particularmente donde hay estructura repetida (Por ejemplo, en las ventanas de la capilla).	16
Figura 4 Reconstrucción obtenida únicamente a partir de puntos triangulados. Los puntos están unidos por caras triangulares mapeadas con textura, obtenidas usando triangulación Delaunay. Esta reconstrucción es bastante ruidosa.	17
Figura 5 Vista aérea de la reconstrucción. Los puntos verdes pertenecen al plano izquierdo; Los puntos azules pertenecen al plano de la derecha. Los valores atípicos se muestran en rojo. Las líneas ajustadas se superponen en la vista.....	18
Figura 6 (a) Proyección del modelo plano antes de ser optimizado, (b) después de ser optimizado y (c) planos de textura mapeados.	19
Figura 7 Tres pasos para optimizar el eje de extrusión.	20
Figura 8 Diferencias entre un borde, una esquina y una zona lisa o sin borde en función de la intensidad.	25
Figura 9 Orientación del borde según la dirección del eje	26
Figura 10 (a) Recta en el plano xy; (b) espacio de parámetros	28
Figura 11(a) Parametrización de las líneas en el plano xy; (b) curvas sinusoidales en el plano ρ	29
Figura 12 Representación del ajuste de una esquina. La esquina azul es la esquina que se desea ajustar, y la esquina verde es una candidata.	31
Figura 13 Máquina de estados del plan perceptivo	35
Figura 14 Máquina de estados para hallar la primera esquina del modelo.....	37
Figura 15 Máquina de estados para hallar la segunda esquina del modelo.....	38
Figura 16 Máquina de estados para hallar la primera esquina del modelo.....	40
Figura 17 Modelo del escenario más básico.....	44
Figura 18 Modelo del escenario con algunos objetos sobre la mesa.....	45
Figura 19 Modelo del escenario con algunos objetos sobre la mesa e inclinación de cámara.....	45

Figura 20 Modelo del escenario con objetos adicionales.....	46
Figura 21 Modelo del escenario con muchos objetos sobre la mesa.....	46
Figura 22 Escenario con muchos objetos, vista de las esquinas sin filtrado de altura. ..	47
Figura 23 Mesa larga, vista inicial.....	48
Figura 24 Mesa larga vista después de realizar el modelado.	48
Figura 25 Se parte de una esquina no valida (esquina azul).....	54
Figura 26 Intersecciones entre rectas de Hough de la situación de la figura 25.....	54
Figura 27 Se parte de otra esquina cercana a una intersección (ver figura 26).	54
Figura 28 Se alcanza la tercera esquina (el punto verde junto al rojo es la dirección de búsqueda).....	55
Figura 29 Ejemplo de detección de vista parcial de la mesa, en el escenario 5.	55
Figura 30 Ejemplo de esquina no deseada priorizada por intersecciones entre rectas de Hough ocasionando la generación de esquinas únicamente a su altura. Mientras que en los demás escenarios se descartan con facilidad, en el escenario 6 tiene un fuerte impacto en el rendimiento	57
Figura 31 Resultado de las pruebas del escenario 6, donde se priorizan las esquinas cercanas a puntos de corte entre rectas de Hough.	58
Figura 32 Resultados de las pruebas en el escenario 6 priorizando las esquinas cercanas a puntos de corte entre rectas de Hough.	59
Figura 33 La máquina de estados detecta un vista parcial de la mesa sin esquinas de partida. Se centra la cámara en el punto indicado por la imagen de la izquierda.....	59
Figura 34 Se escoge una esquina de partida (la esquina negra), y se busca la segunda esquina del modelo. Se detecta una vista parcial de la mesa y se centra en el punto de corte (el punto verde).....	60
Figura 35 Una vez centrada la cámara se retoma la búsqueda de la segunda esquina, pero se vuelve a detectar una vista parcial de la mesa. Volvemos a centrar la cámara (punto verde de la imagen izquierda).	60
Figura 36 Se alcanza la segunda esquina (color rojo).	61
Figura 37 Se alcanza la tercera esquina y se calcula el modelo resultante (imagen derecha).	61
Figura 38 Situación en la que el modelo inicial generado por el sistema presenta incorrecciones.....	62
Figura 39 Datos del modelo erróneo	63
Figura 40 Datos del modelo corregido.	64

Figura 41 Ejemplo de vista que no detecta bien las esquinas mediante Harris.	65
Figura 42 Esquinas de Hough sin supresión de no máximo.....	66
Figura 43 Se aplica la supresión del no máximo y se incorporan las esquinas de Harris.	66

RESUMEN

El principal objetivo de este trabajo es explorar una línea de modelado de objetos que permita que un robot móvil dotado de sensores de visión sea capaz de extraer el modelo de los objetos de manera activa, sin partir de una representación densa de los mismos. Así, a partir de diferentes primitivas visuales (esquinas, bordes, líneas), el robot deberá poner en marcha una serie de “planes perceptivos” que permitan extraer la información necesaria para obtener una descripción cerrada del objeto. Estos planes darán lugar a una actividad en el robot dirigida a describir el objeto y a mantener un modelo del mismo a lo largo del tiempo. Como trabajo experimental, se plantea un escenario en el que el objeto que deberá ser descrito es una mesa situada en una habitación en la que el robot se encuentra localizado. Se realizarán pruebas de distintos niveles de complejidad: mesa sin objetos, vistas parciales, mesa con objetos, oclusiones provocadas por otros objetos.

1 INTRODUCCIÓN

Hoy en día los gráficos 3D se han convertido en un elemento cotidiano en ordenadores domésticos y, en muchos sectores profesionales, existe una creciente demanda para la generación de modelos 3D realistas sin el uso costoso de paquetes CAD y técnicos. La adquisición automática de modelos 3D a partir de secuencias de imágenes es una aproximación al problema que se ha perseguido activamente por los investigadores de visión artificial en los últimos años [1]. En muchas ocasiones, la finalidad de estos modelos 3D puede ser crear un prototipo de un objeto a partir de una secuencia estática y previamente seleccionada de imágenes del objeto a modelar desde distintas perspectivas como se muestra en la figura 1 [3].

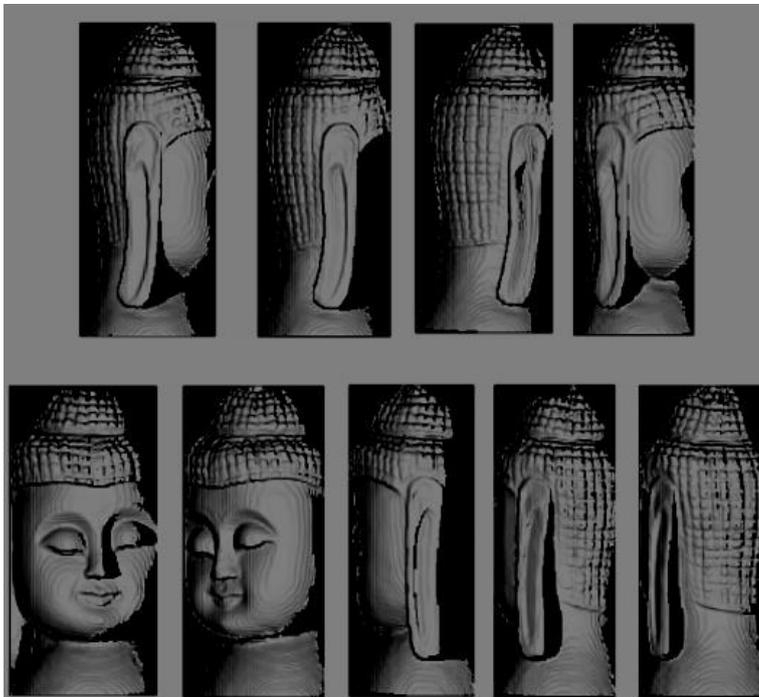


Figura 1 Vistas para reconstruir el modelo 3D de un objeto. [3]

Posteriormente, y utilizando el modelo reconstruido a partir de las vistas de la figura 1 se podría generar nuevas vistas diferentes como se muestra en la figura 2.

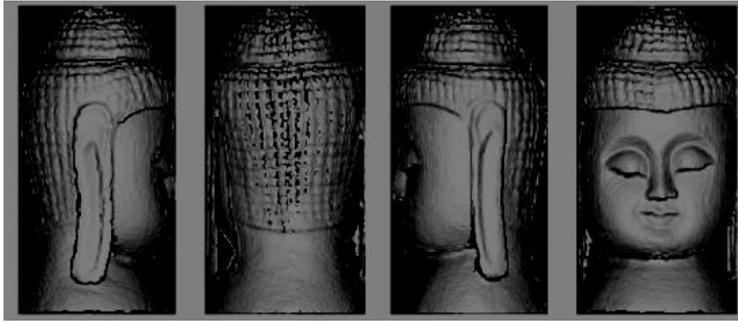


Figura 2 Cuatro nuevas vistas generada a partir del modelo 3D de la figura 1. [3]

En el campo de la robótica, ciertas capacidades, como la habilidad para manipular objetos, son esenciales cuando hablamos de robots que realizan tareas como asistir a los seres humanos en entornos domésticos [2]. Estos robots planifican y ejecutan un agarre de los objetos que le rodean, y deben hacerlo de forma fiable. Sin embargo, para realizar la planificación del agarre es necesario una planificación perceptiva previa, ya que requiere información del objeto, y en entornos no estructurados (no se parte de una representación densa de los objetos) serán necesario algoritmos que nos permitan extraer el modelo de los objetos a partir de la información que obtiene de los sensores de visión.

El modelado automático con secuencia de imágenes por general utiliza características de bajo nivel como puntos y líneas, que pueden ser detectados automáticamente con bastante facilidad. Sin embargo la reconstrucción de cada punto o línea individual implica estimar un gran número de parámetros ($3n$ parámetros para cada n puntos, por ejemplo), lo cual hace que este enfoque este condicionado y dependiente de muchos puntos y líneas característicos detectados con precisión.

Es posible suavizar los requisitos para una completa automatización, los sistemas iterativos pueden utilizar primitivas de nivel superior, tales como planos y bloques, los cuales son extraídos automáticamente de una secuencia de imágenes reduciendo significativamente la dimensión del problema, obteniendo así, soluciones más robustas y modelos más convincentes. Sin embargo estos sistemas únicamente pueden modelar escenas en las cuales las primitivas que usan son aplicables, y no son completamente automáticos.

En robots autónomos que interactúan con objetos del mundo es fundamental que incorporen un sistema perceptivo que proporcione el modelado de dichos objetos. El modelo generado se utiliza en otros planes, como planes de agarre de objetos, pero por

lo general no vuelven a comprobar ni realizar ajustes sobre el propio modelo. No existe una visión activa sobre el objeto modelado.

En este trabajo se propone una estrategia activa para el modelado de objetos por robots móviles utilizando información visual, que surge como alternativa al modelado convencional. Así, la generación del modelo de un objeto se plantea a través de un plan perceptivo que permita dirigir las acciones del robot a obtener, mantener y mejorar un modelo que represente geoméricamente al objeto de manera completa. La principal ventaja de este esquema es que el robot no necesita partir de una representación densa del objeto para poder describirlo, superando así problemas habituales relacionados con oclusiones o vistas parciales del objeto. Además, se plantea una relación entre objeto y plan perceptivo que permite, no sólo obtener un modelo a partir del conocimiento a priori de lo que se está modelando, sino también describir qué relaciones entre percepción y acción deben verificarse para clasificar un objeto cómo perteneciente a una determinada categoría.

El resto de la memoria describe la propuesta desarrollada de manera completa. En la sección 2 se detallan los objetivos concretos que se plantearon desde el inicio del trabajo. La sección 3 presenta otras técnicas de modelado utilizadas tanto en exteriores como para objetos en interiores. En la sección 4 se describe la propuesta concreta y se dan detalles de la implementación desarrollada. La sección 5 incluye los resultados obtenidos durante las pruebas del sistema desarrollado. Por último, la sección 6 presenta las principales conclusiones del trabajo.

2 OBJETIVOS

El principal objetivo de este trabajo es implementar un sistema de modelado activo de objetos para un robot móvil dotado de sensores de visión. Puesto que se trata de una propuesta experimental, el trabajo se centra en un objeto concreto: una mesa rectangular situada en una habitación en la que el robot se encuentra localizado. La elección de este objeto se ha basado en la variedad de situaciones que se pueden plantear con él. Por un lado sus dimensiones pueden ser muy variables. Además pueden plantearse múltiples escenarios en los que la mesa aparece parcialmente ocluida por otros objetos situados encima o en los que la vista no es completa. Estas situaciones sólo pueden resolverse adecuadamente desde una estrategia activa.

Para cumplir este objetivo global se han planteado los siguientes subobjetivos:

- **Definir la representación interna del objeto**, es decir, establecer la parametrización que caracterizará el modelo de la mesa.
- **Definir las primitivas visuales que se utilizarán durante el modelado.** Se analizarán y seleccionarán características visuales de bajo/medio nivel que sean de utilidad para el modelado del objeto mesa.
- **Definir el conjunto de acciones necesarias** que permitirán resolver las diferentes situaciones que puedan darse durante la generación del modelo.
- **Diseñar e implementar la máquina de estados** que conformará el plan perceptivo asociado con el objeto mesa.
- **Definir una batería de pruebas** que permita validar la estrategia de modelado desarrollada y obtener resultados cualitativos y cuantitativos que la avalen..

3 ESTADO DEL ARTE

En el campo de la robótica el modelado de objetos 3D se convierte en un elemento muy importante capaz de dotar a robots autónomos de una percepción del mundo real, permitiendo así, el desarrollo de tareas como la planificación de agarre de objetos [2].

El enfoque más frecuente para dotar a los robots de dicha información es utilizar una base de datos de mallas poligonales que representan los diferentes objetos que el robot puede manipular. Dicha información sobre la geometría del objeto puede ser utilizada por los planificadores de presión para sintetizar la forma del agarre de la mano y la orientación para la iteración física con un objeto. Sin embargo, esto es válido cuando trabajamos en un dominio estructurado donde contamos con un pequeño conjunto de diferentes objetos.

Cuando no trabajamos en un dominio estructurado, es posible que no tengamos información sobre muchos de los objetos, y se necesita de otros métodos para desarrollar la planificación perceptiva.

El modelado automático se convierte en un elemento de suma importancia en estas ocasiones. Existen varias representaciones válidas cuya eficacia está condicionada al tipo de objeto y de la información que conocemos de antemano sobre dicho objeto u objetos.

Podríamos utilizar planos para el modelado de construcciones arquitectónicas y mobiliario, es decir, objetos que se adaptan bien a la estructura de un plano, esto es útil ya que la utilización de características de alto nivel reduce significativamente el tamaño del problema [1].

Existen otras posibilidades más complejas como utilizar extrusiones lineales para caracterizar objetos basados en extrusiones, o utilizar representaciones supercuádricas basadas en la determinación de simetrías en las nubes de puntos.

Estas dos últimas representaciones son muy útiles cuando queremos desarrollar una planificación de agarre de objetos, pues pueden utilizarse para planificar agarres sobre objetos que involucran regiones de la nube que actualmente son invisibles, permitiendo planificar una gama más amplia de agarres que incluyen, por ejemplo, agarres laterales basados en una oposición de los dedos situados en la parte delantera (visible) y la parte trasera (no visible) del objeto.

3.1 Modelado de objetos utilizando planos

A continuación se presenta un ejemplo de sistema que utiliza planos para modelar construcciones arquitectónicas a partir de una secuencia de escenas [1]. Utiliza tres etapas para realizar la tarea: partir de una reconstrucción basada en correspondencias de esquinas, inicializar un modelo basado en planos y optimizar dicho modelo.

Este modelo de sistema de modelado podría ser utilizado por un robot en su plan perceptivo para modelar objetos que pueden ser modelados utilizando primitivas visuales basadas en planos.

Para este caso, cada uno de estos planos p viene especificado por: parámetros espaciales θ_S^p , parámetros de fronteras θ_B^p , parámetros de textura θ_T^p , y parámetros de paralaje θ_Z^p , parámetros que serán necesarios para realizar estimaciones durante el modelo, y en adición permitiendo mapear texturas en el modelo resultante.

Los parámetros espaciales determinan la posición y orientación del plano p en las coordenadas $[X Y Z]$ del eje de coordenadas del mundo y están definidas como “ $\theta_S^p * [X Y Z] = I$ ”. Los parámetros de fronteras son una lista de puntos los cuales definen un recorrido en sentido horario en torno al límite del plano. Los parámetros de textura consisten en una serie de valores de brillo desconocidos i (entre 0 y 255) definidos para cada punto X en una malla 2D regular a través del límite del plano. Cada elemento del conjunto de parámetros de paralaje define un parámetro $x(X)$ de paralaje para cada punto X , que define el desplazamiento del punto X perpendicular al plano. El modelo contiene un plano de fondo, con extensión infinita.

Posteriormente, el sistema utiliza estos datos para realizar una estimación del plano, donde dado los datos D para una secuencia de imágenes el problema es encontrar:

$$\theta = \cup p \theta_S^p \theta_B^p \theta_T^p \theta_Z^p$$

tal que la probabilidad posterior

$$p(\theta|DI) \sim p(\theta|DI) p(\theta|I)$$

Es maximizada, donde I representa información anterior. Este es un producto entre la probabilidad y la probabilidad a priori.

3.1.1 INICIALIZACIÓN DEL MODELO PLANO

La estimación inicial de modelo plano se obtiene a partir de una reconstrucción 3D basada en correspondencias de características. A continuación se describe brevemente la adquisición del modelo 3D y cómo se segmenta para derivar una estimación inicial del modelo plano.

La primera estimación del modelo se basa en una reconstrucción basada en correspondencias de esquinas, las cuales se detectan utilizando el detector de esquinas de Harris. Para cada esquina en la posición (x, y) en la primera imagen, las coincidencias candidatas de un vecino de (x, y) en la segunda imagen se clasifican por correlación cruzada. Para reducir el número de falsas correspondencias, la geometría epipolar (es decir, la matriz fundamental F) entre el par de imágenes se estima utilizando RANSAC [4].



Figura 3 Correspondencias obtenidas de F estimadas directamente a partir de los datos (izquierda) y F estimadas a partir de un plano dominante más el paralaje (derecha). Las correspondencias obtenidas a través del plano más el paralaje son más fiables, particularmente donde hay estructura repetida (Por ejemplo, en las ventanas de la capilla).

a partir de las correspondencias 2D-3D, y una nueva estructura se triangula a partir de las coincidencias 2D-2D. En la figura 11 se muestra un modelo obtenido utilizando este método.



Figura 4 Reconstrucción obtenida únicamente a partir de puntos triangulados. Los puntos están unidos por caras triangulares mapeadas con textura, obtenidas usando triangulación Delaunay. Esta reconstrucción es bastante ruidosa.

3.1.2 EXTRACCIÓN DE PLANOS DE LA RECONSTRUCCIÓN

Los planos se extraen de la reconstrucción 3D mediante la aplicación recursiva del algoritmo de estimación de planos RANSAC [1, 4]. Por cada iteración, Los planos se hipotetizan a partir de una muestra aleatoria de 3 puntos, y se selecciona el plano que más se adapta a aquellos puntos que no han sido asignados a planos anteriores.

El principal problema para este enfoque es que no incluye el concepto de extensión espacial para ninguno de los planos, y por lo tanto tiende a extraer planos con límites irregulares y mutuamente solapados. Podemos mitigar este efecto aplicando restricciones espaciales que dependen de las características del objeto, para construcciones arquitectónicas se asume que todos los planos son perpendiculares a un plano de suelo común, pues el plano comúnmente dominante se corresponde con las paredes de las construcciones.

Siempre que se haya extraído al menos dos planos del modelo puntual, se puede obtener una estimación por mínimos cuadrados del plano de suelo normal a partir de las otras normales. Ahora se puede simular una vista desde arriba de la reconstrucción proyectando todos los puntos ortogonalmente sobre el plano de suelo.

Esta vista es útil, ya que le se pueden aplicar muchas restricciones robustas y comunes a los modelos arquitectónicos, por ejemplo, puesto que se espera que las paredes

adyacentes sean susceptibles de intersectar a unos 90 grados, podemos definir una probabilidad para aplicarla al parámetro espacial θ_S^p .

En la Figura 12 se muestran dos líneas que han sido ajustadas a dos conjuntos de "inliers" obtenidos de la estimación mediante RANSAC recursiva. Estas líneas se cortan proporcionando una estimación inicial de la extensión espacial (paralela al plano de suelo) de cada plano. Los límites superior e inferior de cada plano también son inicializados por los puntos "más alto" y "más bajo" de cada grupo, donde la altura se mide perpendicularmente al plano de suelo. De este modo, se obtiene un sencillo modelo plano inicial en el que todos los planos tienen forma rectangular y son perpendiculares a un plano de suelo. Este modelo se utiliza a continuación como semilla de búsqueda de una estimación mejorada.

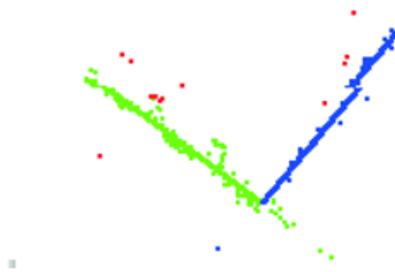


Figura 5 Vista aérea de la reconstrucción. Los puntos verdes pertenecen al plano izquierdo; Los puntos azules pertenecen al plano de la derecha. Los valores atípicos se muestran en rojo. Las líneas ajustadas se superponen en la vista.

3.1.3 OPTIMIZACIÓN EL MODELO PLANO

Teniendo un modelo 3D consistente en un conjunto de rectángulos, y definida una medida de probabilidad para el modelo, se lleva a cabo una búsqueda de los parámetros del modelo [1]. El proceso de extracción de planos tiene tres fuentes principales de error: la estimación por mínimos cuadrados del plano de suelo, la segmentación de los planos por la intersección de sus proyecciones en el plano suelo, y la aproximación de cada plano limitado por un rectángulo. De esta manera, el modelo se optimiza mediante la búsqueda por descenso del gradiente en: (a) los componentes del plano de suelo, (b) los extremos de la proyección de cada plano en el plano de suelo y (c) el límite de cada plano.

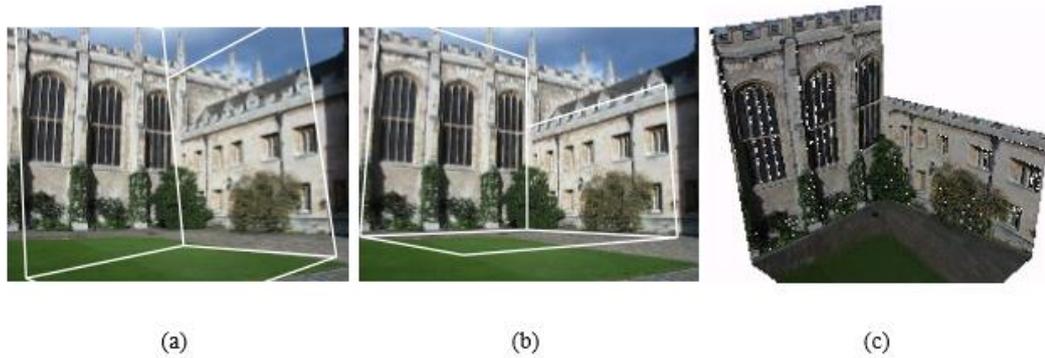


Figura 6 (a) Proyección del modelo plano antes de ser optimizado, (b) después de ser optimizado y (c) planos de textura mapeados.

3.2 OBJETOS BASADOS EN EXTRUSIONES

El diseño de la planificación de agarre en los robots autónomos requiere información sobre la forma completa del objeto a manipular [2]. Muchos objetos domésticos se basan en extrusiones. De hecho, muchos sistemas de modelado y fabricación utilizan extrusiones lineales y rotacionales en una jerarquía para generar los modelos utilizados para la fabricación. Descubrir las extrusiones en nubes de puntos parciales puede ayudar a generar una nube de puntos completa a partir de una observación parcial.

Además, este conocimiento se puede utilizar para crear un gran conjunto de agarres factibles a partir de los cuales un planificador puede escoger candidatos adecuados para la ejecución de tareas. Por ejemplo, la detección del eje de simetría en un objeto rotacionalmente simétrico nos permite girar cualquier agarre factible alrededor de este eje.

Un ejemplo de sistema de este tipo puede realizarse en tres etapas utilizando por ejemplo extrusiones rotacionales. En el primer paso, utilizamos puntos de las nubes de puntos parciales para generar hipótesis para el eje de extrusión. En el caso de las extrusiones rotacionales, muestreamos aleatoriamente pares de puntos y usamos la normal de cada punto para crear una línea. Cada par de líneas se interseca y el punto resultante se utiliza como una hipótesis para el eje de extrusión. La figura 14 muestra un ejemplo de puntos muestreados desde un objeto cilíndrico. Para medir el ruido, usamos el punto medio de la línea que conecta los puntos más cercanos, en caso de que las dos líneas no se intersecten. Los puntos de hipótesis recogidos se utilizan entonces para crear una estimación inicial del eje de extrusión. Con este fin, ponemos una línea en el conjunto de hipótesis utilizando mínimos cuadrados lineales. El

algoritmo RANSAC se utiliza además para reducir la influencia de los valores atípicos. Dada esta estimación inicial, realizamos la optimización para producir un eje de extrusión más preciso.

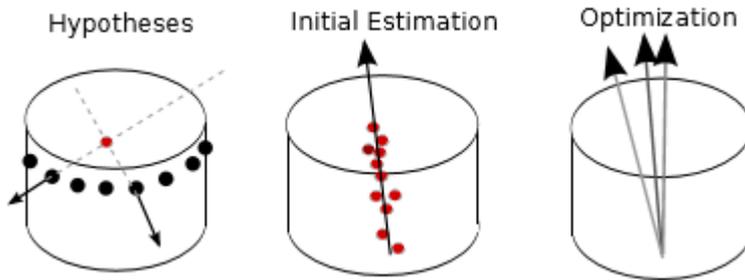


Figura 7 Tres pasos para optimizar el eje de extrusión.

Específicamente, se utiliza el algoritmo hill climbing [14] para buscar un eje de extrusión que reduzca la dispersión de puntos a lo largo del perfil del objeto. En cada iteración, el eje de extrusión se utiliza para rotar todos los puntos de la nube de puntos parcial hacia atrás en un plano. A continuación, se estima la densidad de los puntos utilizando un kernel estimador de densidad. Al maximizar la densidad usando el algoritmo hill climbing, podemos reducir la dispersión de los puntos proyectados, recreando así el perfil del objeto.

4 MATERIAL Y MÉTODO

4.1 HERRAMIENTAS

Este apartado describe las características de las librerías, frameworks y editores utilizados para el desarrollo del proyecto.

4.1.1 KDevelop

Kdevelop es un entorno de desarrollo integrado para sistemas GNU/Linux y otros sistemas Unix, publicado bajo licencia GPL, orientado al uso bajo el entorno gráfico KDE, aunque también funciona con otros entornos como Gnome [11].

Se ha escogido como entorno de trabajo por algunas de sus características que se listan a continuación:

- Editor de código fuente con destacado de sintaxis
- Gestión de proyectos CMake y qmake.
- Navegador entre clases de la aplicación.
- Completado automático del código en C y C++.
- Permite control de versiones.

4.1.2 QTDesigner

QTDesigner es una herramienta para el diseño y construcción de interfaces gráficas de usuario para componentes QT. Los widgets y formularios diseñados se integran perfectamente con el código de programa utilizando el mecanismo de señales y slots de Qt.

QTDesigner se ha utilizado para diseñar una interfaz gráfica que sirvió como apoyo durante la fase de desarrollo.

4.1.3 Robocomp

Dos de los principales problemas encontrados en la creación de software son la escalabilidad y la reutilización. Estos problemas son más graves cuando se trata de software que se utiliza en la robótica debido a que se usa excepcionalmente en este campo [9].

En el campo de la robótica es muy común que los investigadores implementen todos los algoritmos con un diseño rígido orientado a tareas y/o a un robot específico. De

ser así, cuando se completa el implementación, el software desarrollado es a la larga imposible de utilizar. Suele estar tan vinculado a una plataforma o tarea específica que es más práctico comenzar de cero (debido a las dependencias y su rigidez).

La programación orientada a componentes surge como una solución a estos problemas. Es un enfoque que no necesariamente tiene que ver con la concurrencia y la computación distribuida, sino con la forma en que se organiza el software. La programación orientada a objetos representó un gran avance en la programación estructurada, sin embargo, cuando el número de clases y sus interdependencias aumenta, es demasiado difícil de entender el sistema en general. Por lo tanto, resulta beneficioso tener un mayor grado de encapsulación, que combina varias clases relacionadas bajo una única interfaz, y permite comprender el sistema con menos detalles. La programación orientada a componentes muchos la ven como el siguiente paso después de la programación orientada a objetos.

Desde el punto de vista de diseño, un componente, puede verse como una gran clase que ofrece métodos públicos. La única diferencia es, que la complejidad introducida por los componentes dependientes que no son el dominio del problema, desaparece debido a que la interfaz oculta el componente. Un componente puede ser arbitrariamente complejo, pero un paso atrás, todo lo que ves es la interfaz ofrecida. Esto es lo que se define como un componente.

RoboComp es un framework robótico de código abierto, que se compone de diferentes componentes y las herramientas necesarias para su uso.

Los componentes en ejecución forman un grafo de procesos que pueden distribuirse a través de varios núcleos y CPU que utilizan tecnología de componentes. Las comunicaciones son manejadas por el framework Ice, utilizando el enfoque utilizado por los sistemas de robótica Orca2. Los objetivos principales de Robocomp en su diseño del sistema son la eficiencia, la simplicidad y la reutilización. También utiliza en algunos de sus componentes otras herramientas como CMake, Qt4, IPP, OpenSceneGraph y OpenGL. Los lenguajes de programación utilizados son principalmente C++ y Python, aunque el framework Ice permite utilizar componentes codificados en otros lenguajes de forma sencilla. Al igual que ROS, Robocomp se puede llamar sistema operativo del robot, siempre y cuando proporcione una capa de abstracción al hardware del robot.

Si comparamos Robocomp con otros proyectos similares (como Orca o ROS), la principal ventaja es su eficiencia y facilidad de uso.

4.1.4 OpenCV

OpenCV (Open Source Computer Vision Library) es una librería de software de código abierto destinada a la visión artificial y aprendizaje automático, y fue construido para ofrecer una infraestructura común para las aplicaciones de visión artificial y acelerar el uso de la percepción de las máquinas en los productos comerciales [10].

Actualmente la librería cuenta con más de 2500 algoritmos optimizados. Estos algoritmos pueden usarse para detectar y reconocer caras, identificar objetos, extraer modelos 3D, etc.

OpenCV ha sido la librería más importante del proyecto, pues hemos desarrollado nuestro plan perceptivo utilizando las herramientas que nos ofrece. La detección de características se ha llevado a cabo utilizando algoritmos de detección implementados en la librería. Concretamente, para la detección de bordes en la imagen, la detección de esquinas y la detección de líneas en la imagen, se han utilizado los algoritmos de Canny, Harris y Houghlines respectivamente.

4.2 PROPUESTA DE MODELADO ACTIVO

4.2.1 DESCRIPCIÓN GENERAL DE LA PROPUESTA

En este trabajo se ha desarrollado un método de modelado de objetos que permite que un robot móvil dotado de sensores de visión sea capaz de extraer el modelo de los objetos de manera activa, sin partir de una representación completa de los mismos. Así, a partir de diferentes primitivas visuales, el robot pone en marcha una serie de “planes perceptivos” que permiten extraer la información necesaria para obtener una descripción cerrada del objeto. Estos planes dan lugar a una actividad en el robot dirigida a describir el objeto y a mantener un modelo del mismo a lo largo del tiempo.

Se plantea un escenario en el que el objeto que deberá ser descrito es una mesa situada en una habitación en la que el robot se encuentra localizado. La reconstrucción del objeto mesa se basa en el análisis de un flujo constante de imágenes adquiridas a través una cámara de profundidad.

Se han planteado 3 tipos de escenarios sobre los que realizaremos una batería de pruebas para definir nuestro plan perceptivo: una mesa pequeña sin objetos, una mesa pequeña con objetos y una mesa larga sin objetos.

Se ha dotado al robot de dos motores de rotación horizontal y vertical que permiten orientar la cámara durante el modelado, de esta manera, si nuestro plan perceptivo recibe una vista parcial de la mesa, reorienta la cámara para obtener la parte no visible de la mesa cuando sea necesario.

A continuación, en la sección 2.2 se definen las primitivas visuales utilizadas durante las fases de modelado de la mesa y mantenimiento del modelo.

4.2.2 PRIMITIVAS VISUALES

El modelado de los objetos se ha realizado utilizando una serie de primitivas visuales, esquinas, puntos de borde y líneas.

En un primer enfoque, únicamente se utilizaron puntos de borde y esquinas como puntos característicos para realizar el modelado inicial del objeto mesa. Sin embargo, surgió la necesidad de detectar líneas para brindar información extra al modelado y conseguir así una optimización en tiempo de cálculo del modelo.

A continuación se definen las primitivas mencionadas.

4.2.2.1 Esquinas

Las esquinas en imágenes representan información muy útil y son muy importantes para describir objetos para su reconocimiento e identificación. Para nuestro plan perceptivo, la esquina es la primitiva más importante y además es la que se utiliza para definir el objeto mesa. El modelo se considera válido cuando tenemos 4 esquinas que cumplen ciertas restricciones.

Se considera que un punto es una esquina cuando en el entorno local del punto se observa un alto gradiente en 2 direcciones claramente diferentes.

Por el contrario en un borde la intensidad solo varía en una dirección, y en las zonas uniformes la intensidad no varía.

La primera imagen de la figura 1 representa una zona uniforme debido a que no se detecta ningún cambio en la dirección, la segunda imagen representa una zona de borde pues ahora se detecta cambios en una dirección y, por último, la tercera se

correspondería con una esquina, ya que ahora los cambios de dirección se producen en varias direcciones

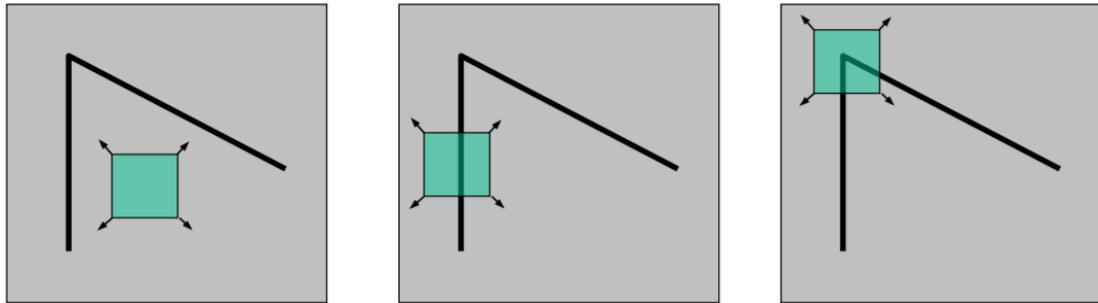


Figura 8 Diferencias entre un borde, una esquina y una zona lisa o sin borde en función de la intensidad.

Los métodos de detección de esquina que utilizan la imagen de intensidad se basan en el cálculo de la matriz de autocorrelación del entorno local de un punto y en la aplicación de un operador sobre dicha matriz que cuantifique el grado de esquina del punto en función de las variaciones de intensidad observadas.

Entre los métodos más populares se encuentran el detector de Shi&Tomasi [7], Harris&Stephens [6] y el de Brown&Szeliski&Winder [8]. En este trabajo se ha utilizado el método de Harris&Stephens por su compromiso entre velocidad de cómputo y calidad de los resultados.

Como paso posterior a la detección de esquinas 2D, la altura de los puntos detectados, obtenida a partir de la imagen de profundidad, es utilizada para seleccionar el conjunto de esquinas que presenta una probabilidad alta de pertenecer a la mesa.

4.2.2.2 Bordes

La detección de bordes es clave en la reconstrucción tridimensional porque se puede extraer información importante de la imagen, como pueden ser las formas de los objetos que la componen. Los bordes indican dónde están los límites de los objetos, su forma, su tamaño, y también ofrecen información sobre su textura. El algoritmo de modelado desarrollado emplea los puntos de bordes como referencia para encontrar pares de esquina que formen segmento de borde, una restricción necesaria para que las esquinas realmente pertenezcan al objeto mesa.

Los bordes se definen como puntos donde hay una frontera entre dos regiones de la imagen. En la práctica, los bordes se representan como cadenas de puntos de la imagen concatenados que presentan una alto gradiente (grandes variaciones de intensidad).

Se han desarrollado múltiples de algoritmos para la detección de bordes. Uno de los más utilizados es el método de canny [12].

El método de canny consta de los siguientes pasos:

Paso 1. Filtrar el ruido de la imagen original antes de localizar y detectar cualquier borde. Se suele emplear un filtro gaussiano utilizando métodos de convolución. Hay que tener en cuenta que cuanto mayor es mayor es el tamaño de la máscara utilizada en el filtro y menor es la sensibilidad del detector al ruido.

Paso 2. Tras suavizar la imagen, se determina la intensidad del borde calculando el gradiente de la imagen. Para calcular la derivada se emplean máscaras de convolución (en dirección horizontal y vertical).

$$|G| = |G_x| + |G_y|$$

Ecuación 3. Aproximación de la intensidad del borde.

Paso 3. Calcular la dirección del borde a partir de la orientación del gradiente.

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

Ecuación 4. Dirección del borde

Paso 4. Después de calcular la dirección del borde, hay que relacionar la dirección del borde con la dirección en la imagen. Así si en una ventana de imagen de 5x5 fijando el punto central como origen, solo puede haber 4 posibles direcciones, 0 grados (dirección horizontal), 45 grados (en la dirección de la diagonal positiva), 90 grados (en la dirección vertical), 135 grados(en la dirección de la diagonal negativa). Teniendo esto en cuenta, la orientación final del borde dependerá de a cuál de estas cuatros direcciones se acerca más:

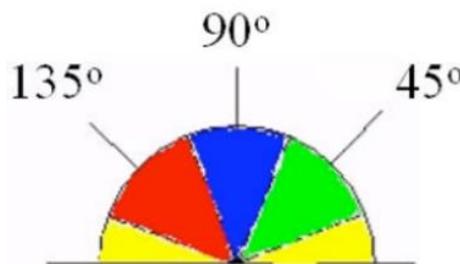


Figura 9 Orientación del borde según la dirección del eje

Si la dirección del eje está en la zona amarilla de la Figura 3 (entre 0- 22.5 o entre 157.5 -180) entonces se pone a 0 grados. Cualquier borde en la zona verde (entre 22.5 - 67.5) a 45 grados. Cualquier borde en la zona azul (67.5- 112.5) se establece a 90 grados. Y si está en la zona roja (entre 112.5 -157.5) a 135 grados.

Paso 5. En este paso se calcula el máximo local en la dirección del gradiente y se elimina el resto. El máximo local se calcula comparando el píxel con los píxeles vecinos en la dirección del gradiente.

Paso 6. En el último paso, en vez de utilizar un umbral único, Canny introduce un umbral con histéresis. Esto se debe a que, si el operador fluctúa por encima y por debajo del umbral, provoca una discontinuidad en el borde. Existen dos niveles de umbral, t_h y t_l . Cualquier píxel que sea mayor que t_h será un borde y si es menor que t_l no será un borde. En el caso de que el píxel se encuentre entre t_h y t_l no será un borde a no ser que esté conectado a otro píxel marcado como borde. La calidad de la detección de bordes en una imagen depende del ajuste de estos dos umbrales.

4.2.2.3 Líneas de Hough

La Transformada de Hough [13] es un algoritmo empleado en reconocimiento de patrones en imágenes que permite encontrar ciertas formas dentro de una imagen, como líneas, círculos, etc. La versión más simple consiste en encontrar líneas rectas. La Transformada de Hough fue propuesta por Paul Hough en 1962 y patentada por IBM. El algoritmo emplea el conjunto total de puntos característicos de la imagen, haciéndolo robusto ante datos incompletos y/o ruidosos. Las líneas de Hough se introdujeron en una fase final del proyecto para producir una mejora de tiempo en la fase de modelado.

Existen dos versiones de la Transformada de Hough para la detección de líneas, dependiendo de la parametrización usada dichas líneas. La primera versión utiliza los parámetros de la pendiente a y el parámetro de intersección con la ordenada b , para lo cual se apoya en la representación de la recta dada por la siguiente ecuación:

$$y_i = ax_i + b$$

Ecuación 1.

Por cada punto del plano pasan infinitas rectas que satisfacen la ecuación anterior para diferentes valores de a y b . Al considerar el plano ab , denominado espacio de parámetros, se obtiene una única recta para cada punto; por ejemplo, si consideramos

dos puntos (x_i, y_i) y (x_j, y_j) , cada punto tendrá asociada una única recta, que al intersectarse darán los parámetros de a y b de la recta que contiene los puntos colineales de (x_i, y_i) y (x_j, y_j) , tal como se muestra en la figura 4.

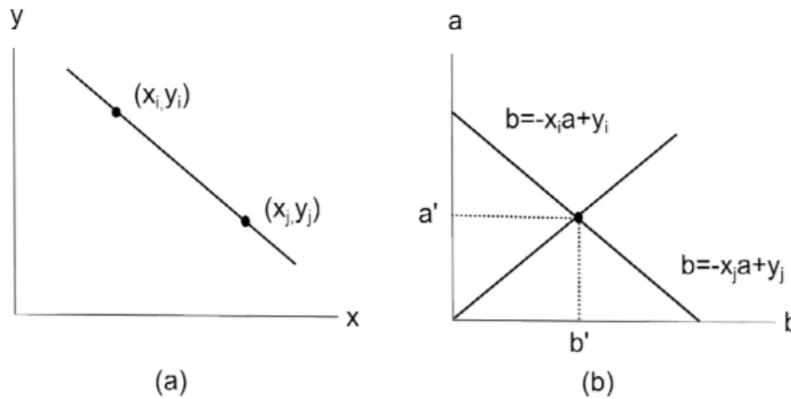


Figura 10 (a) Recta en el plano xy ; (b) espacio de parámetros

En la implementación de la Transformada de Hough se utilizan unas celdas acumuladoras que dividen el espacio de parámetros. Esto permite evaluar todos los posibles valores por cada punto y acumular los frecuentes (fase de votación). El resultado es la creación de una malla; cuanto mayor sea el valor en cada casilla, más probable es la existencia de una línea recta (fase de localización de picos en la malla). El inconveniente de utilizar este espacio de parámetros es la dificultad para detectar líneas verticales debido a que los parámetros pueden llegar a ser infinitos a medida que la línea se hace vertical.

La segunda versión de la Transformada de Hough para detección de líneas utiliza coordenadas polares y está compuesto por los parámetros ρ y θ , con la siguiente ecuación:

$$\rho = x \cos \theta + y \sin \theta$$

Ecuación 2.

donde ρ es la longitud de una normal desde el origen hasta la línea y θ es el ángulo de ρ con respecto al eje x , θ está limitado al rango de $[0 \pi]$. El comportamiento es similar a lo realizado en la primera versión. La diferencia radica en que en vez de rectas, cada punto tiene asociada una senoide en el plano $\rho\theta$, como se muestra en la figura 5. Utilizando las coordenadas polares se soluciona el inconveniente de la primera versión.

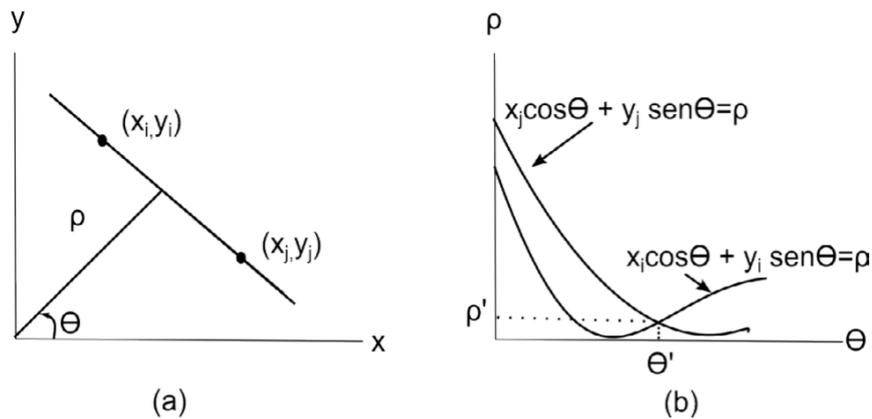


Figura 11(a) Parametrización de las líneas en el plano xy ; (b) curvas sinusoidales en el plano ρ .

4.2.3 CALCULO PRELIMINAR DE LOS DATOS

Este apartado describe brevemente cómo y cuándo se calculan los datos que posteriormente utiliza la máquina de estados en cada iteración. Lo primero que se hace en el hilo principal es obtener las imágenes de cámara, así como la correspondencia 2D-3D para cada píxel de la imagen. Concretamente se obtiene la imagen de grises y la imagen de profundidad.

A continuación, se calcula la imagen de borde utilizando el algoritmo de canny proporcionado por OpenCV. El resultado es una imagen de binaria, donde cada píxel toma valor 0 si no es borde y 255 si es borde.

A través del detector de esquinas, que configuramos a conveniencia según el estado en el que se encuentra la máquina de estados principal, obtenemos las esquinas utilizando el método de Harris, y también obtenemos un vector con los puntos de cortes entre rectas calculadas con la Transformada de Hough que aportará información extra sobre las esquinas de la mesa. La configuración del detector determinará qué esquinas se seleccionan y en qué orden en que las devuelve.

4.2.4 DETECTOR DE ESQUINAS

Para el cálculo de esquinas se ha desarrollado un detector específico, que utiliza esquinas de Harris y se puede apoyar con los puntos de corte que generan las rectas de Hough cuando se cortan entre sí. La configuración del detector determinará la calidad y formato de las esquinas. La función del detector es arreglar los problemas

que se originan con las correspondencias 2D-3D y ofrecer las mejores soluciones para cada configuración.

Uno de los problemas de un punto de esquina puede ser, por ejemplo, que sea frontera entre un punto de la mesa y de la pared. Sea $Q = (q_1, q_2)$ una esquina próxima a una esquina real de la mesa y sea C el conjunto de puntos del entorno en un radio r de Q , pueden existir n puntos cuyas coordenadas en el sistema de referencia 3D no pertenecen al tablero mesa.

El problema podemos reducirlo a encontrar una primera esquina de la mesa. Para resolverlo, durante la fase de supresión del no máximo, tomando cada esquina $Q_i = (q_{i_1}, q_{i_2})$ del conjunto de esquinas C obtenidas mediante la detección de esquinas, y siendo $Q_i' = (q'_{i_1}, q'_{i_2}, q'_{i_3})$ las coordenadas 3D con respecto al mundo, $B = (b_1, b_2, b_3)$ las coordenadas del robot con respecto al mundo, $d = D(Q_i, B)$ la distancia entre Q_i' y B , se priorizan los máximos locales que minimizan la distancia d .

De esta manera evitamos esquinas pertenecientes al plano de suelo o de la pared. Únicamente puede darse el caso de que objetos situados en la mesa u objetos que están delante de la propia mesa puedan generar esquinas cuyas coordenadas 3D no pertenezcan a la mesa pero tiene menos impacto y en la mayoría de los casos no impiden llegar a una solución.

Para el resto de los casos de los estados correspondientes al modelado, sea $Q_i = (q_{i_1}, q_{i_2})$ una esquina del conjunto de esquinas C calculadas con Harris con coordenadas 3D $Q_i' = (q'_{i_1}, q'_{i_2}, q'_{i_3})$, $E_1 = (e_1, e_2, e_3)$ las coordenadas de la primera esquina del modelo y $dif = |q'_{i_2} - e_2|$ la diferencia de sus coordenadas en el eje Y del mundo, se priorizan los máximos locales que minimizan la diferencia dif . De esta manera el detector proporciona puntos de esquina cuyas posiciones en el mundo se encuentran a la misma altura.

El detector de esquinas realiza gran parte del trabajo de la fase de mantenimiento del modelo. Así, tomando cada esquina del modelo $A' = (a'_1, a'_2, a'_3)$ y su proyección sobre el plano 2D $A = (a_1, a_2)$, devuelve una lista L de todas las esquinas que detecta en el entorno de A ordenadas de mayor a menor calidad.

La función del mantenimiento del modelo consiste en expandir y estabilizar el modelo, seleccionando una de las esquinas pertenecientes a L .

La calidad de las esquinas de la lista viene por uno de los ángulos α o β (ver figura 6) utilizando el que más se aleja del valor $\pi/2$.

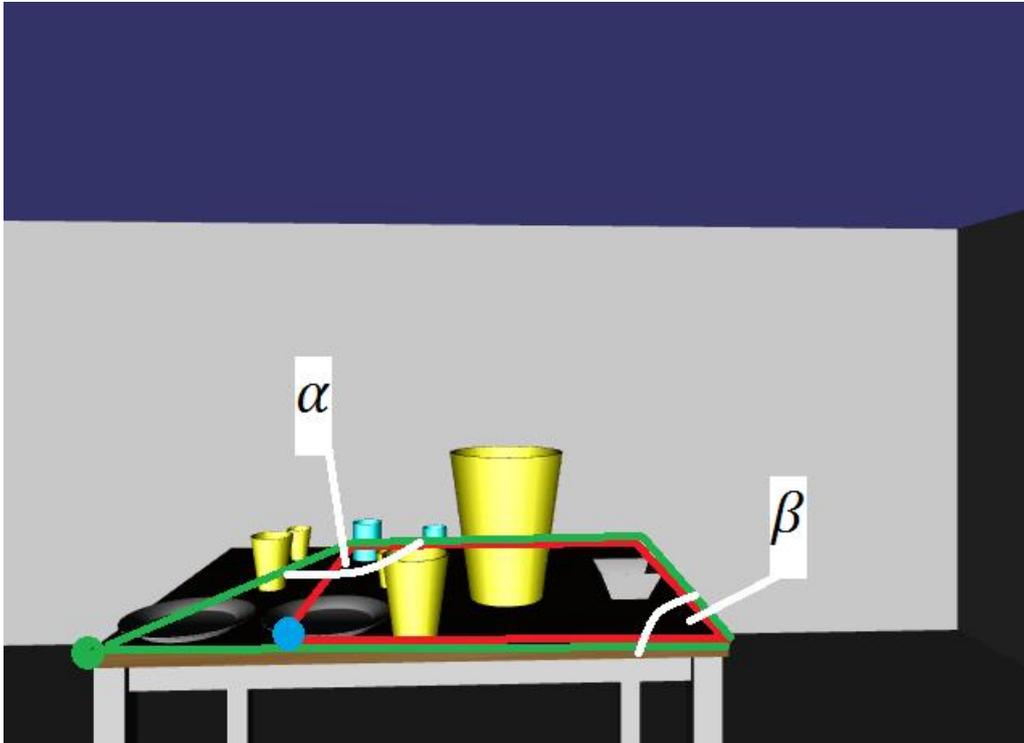


Figura 12 Representación del ajuste de una esquina. La esquina azul es la esquina que se desea ajustar, y la esquina verde es una candidata.

Para una esquina cualquiera, la primera comprobación es que al menos uno de los dos ángulos es cercano a $\pi/2$, y si no se cumple la esquina se descarta de ser añadida a L . La calidad de la esquina vendrá dada por el ángulo que más dista de $\pi/2$, α en el caso de la figura 6. Priorizaremos los ángulos que más distan de $\pi/2$, para favorecer la expansión del modelo.

4.2.5 MÉTODOS DE BÚSQUEDA DE ESQUINA DEL MODELO

Como se ha comentado en puntos anteriores definimos el modelo de la mesa utilizando esquinas. Los siguientes apartados describen los métodos utilizados para la búsqueda de esquinas.

4.2.5.1 BÚSQUEDAS CON DIRECCION

Dada una esquina de partida A con coordenadas $A(a1, a2)$ y un punto B con coordenadas $B(b1, b2)$, donde $\alpha = \text{atan2}(-(a1 - b1), (a2 - b2))$, se realiza una búsqueda en la dirección definida por \overrightarrow{AB} y para todo punto C con coordenadas $C(c1, c2)$ donde $\theta = \text{atan2}(-(a1 - c1), (a2 - c2))$ se analiza y se comprueba si es esquina si se cumple que $|\alpha - \theta|$ es menor que 0.08. La búsqueda termina cuando se encuentra una esquina que cumpla una serie de condiciones que se detallan a continuación, o bien si se han analizado todos los puntos que cumplen la condición anterior y no se ha encontrado una esquina.

Requisitos de esquina válida:

- Es esquina.
- No es una esquina visitada.
- Sea A la esquina de partida y C la esquina candidata donde $A' = (a'_1, a'_2, a'_3)$ y $C' = (c'_1, c'_2, c'_3)$ sus coordenadas en el sistema de referencia del mundo respectivamente y sea $d = \sqrt{(a'_1 - c'_1)^2 + (a'_2 - c'_2)^2 + (a'_3 - c'_3)^2}$ la distancia real en milímetros que las separa, es una esquina válida si d pertenece al rango de valores definido por [distancia mínima, distancia máxima].

No es necesario comprobar la altura real de la esquina candidata, ya que una vez tenemos una altura determinada por la primera esquina el detector de Harris únicamente nos da las esquinas que están a una altura aproximada.

Este método de búsqueda puede ser utilizado a conveniencia en varios estados de la máquina de estados que determina el plan perceptivo: SEGUNDA_ESQUINA, TERCERA_ESQUINA y TERCERA_ESQUINA_DESDE_PRIMERA.

4.2.5.2 BÚSQUEDAS UTILIZANDO RECTAS

Sea VR el vector de rectas calculadas utilizando los datos obtenidos con la transformada de Hough $A = (a_1, a_2)$ y la esquina de partida y $R_i = r_1x + r_2y + r_3$ una recta almacenada en la posición i del vector VR , y $D(A, R_i) =$

$\frac{|a_1 * r_1 + a_2 * r_2|}{\sqrt{a_1^2 + a_2^2}}$ la distancia entre el punto A y R_i . Para todas las rectas i donde $D(A, R_i) < d$ para $d = 4$, se busca una esquina, $C = (c_1, c_2)$

perteneciente al conjunto de esquinas calculadas previamente por el detector, que cumpla las siguientes condiciones:

- Es esquina.
- $D(C, R_i) < d$.
- No es una esquina visitada.
- Sea A la esquina de partida y C la esquina candidata donde $A' = (a'_1, a'_2, a'_3)$ y $C' = (c'_1, c'_2, c'_3)$ son sus coordenadas en el sistema de referencia del mundo respectivamente y sea $d = \sqrt{(a'_1 - c'_1)^2 + (a'_2 + c'_2)^2 + (a'_3 + c'_3)^2}$ la distancia real en milímetros que las separa, C es una esquina válida si d pertenece al rango de valores definido por $[distancia\ mínima, distancia\ máxima]$.

La ventaja de este método es que es más rápido que el anterior, ya que la eficacia de houghlines depende de la posición de la mesa en la imagen que obtenemos de la cámara para ofrecer buenas soluciones.

4.2.5.3 CÁLCULO DE LA CUARTA ESQUINA

Cuando tenemos las 3 primeras esquinas, la cuarta esquina se calcula como se describe a continuación:

Sean $A = (a_1, a_2, a_3)$, $B = (b_1, b_2, b_3)$ y $C = (c_1, c_2, c_3)$ las coordenadas 3D de las esquinas 1, 2 y 3 respectivamente del modelo donde $V_{12} = A - B$, con coordenadas $V_{12} = (v_1, v_2, v_3)$, las coordenadas de la cuarta esquina E vienen definidas por $E = (v_1 + c_1, v_2 + c_2, v_3 + c_3)$.

La esquina se anota en el modelo si se cumplen las siguientes condiciones:

- Sea $E' = (e'_1, e'_2)$ la proyección sobre la imagen del punto E calculado, existe una esquina $Q = (q_1, q_2)$ perteneciente al conjunto de esquinas calculadas con el detector donde $d = \sqrt{(e'_1 - q_1)^2 + (e'_2 + q_2)^2}$ tiene un valor inferior a 10.
- Sea $d = \sqrt{(e_1 - c_1)^2 + (e_2 + c_2)^2 + (e_3 + c_3)^2}$ la distancia real en milímetros entre C y E , se considera esquina válida si d pertenece al rango de valores definido por $[distancia\ mínima, distancia\ máxima]$.

Si la esquina calculada está dentro del rango de la cámara y no se satisface alguna de las restricciones esta se desecha y la creación del modelo vuelve a un punto anterior desde el que realizar nuevas búsquedas.

Si la esquina calculada no es visible, se anota la esquina calculada en el modelo y será durante el mantenimiento cuando se compruebe su validez.

4.2.6 MÁQUINA DE ESTADOS

El objetivo principal del plan perceptivo es encontrar el modelo del tablero superior de la mesa que caracterizaremos como un tablero delimitado por 4 esquinas, detectadas previamente por un detector de esquinas, que servirá como modelo a mantener en el tiempo. Por lo tanto, cada vez que hablemos del modelo haremos referencia a la secuencia de esquinas que lo describen.

Este apartado describe la máquina de estados definida para el funcionamiento del plan perceptivo. Se describen las estructuras utilizadas para almacenar información de los estados y se detalla la función de cada uno de estos y cómo están estructurados.

El modelo calculado tiene que cumplir una serie de requisitos:

- La distancia entre esquinas consecutivas tiene que pertenecer a un rango de valores preestablecidos, distancia máxima y distancia mínima.
- El ángulo que forman tres esquinas consecutivas A , B y C respectivamente, donde el vértice del ángulo es la esquina B , tienen que formar aproximadamente un ángulo con valor $\pi/2$ radianes, es decir un ángulo recto.

El funcionamiento de la máquina de estados (figura 6) es el siguiente:

- Se establece como INIT estado de partida. Cada uno de los estados posteriores se encargará de extraer un modelo inicial, excepto el estado MANTENIMIENTO. Este último se encargara de mantener el modelo en el tiempo, realizando los ajustes necesarios para intentar estabilizarlo/mejorarlo.
- El primer estado encargado del modelar el objeto mesa es PRIMERA_ESQUINA. Si este estado tiene éxito realiza una búsqueda de la segunda esquina desde el estado SEGUNDA_ESQUINA. Partiendo de la segunda esquina se realiza una búsqueda de la tercera en el método TERCERA_ESQUINA y la última de las esquinas se calcula en el estado

CUARTA_ESQUINA. Si uno de los estados no tiene éxito se descarta la última esquina añadida al modelo y volvemos al estado en que se calculó para buscar otra esquina alternativa, excepto el estado TERCERA_ESQUINA, que pasa a un estado alternativo TERCERA_ESQUINA_DESDE_PRIMERA que realizará la búsqueda de la tercera esquina del modelo partiendo de la primera esquina calculada. Es entonces cuando vuelve al estado SEGUNDA_ESQUINA en caso de no encontrar una solución que satisfaga las restricciones establecidas.

- Obtenido con éxito un modelo del tablero del objeto mesa, la máquina de estados avanza al estado MANTENIMIENTO que se encarga de mantener el modelo en el tiempo.

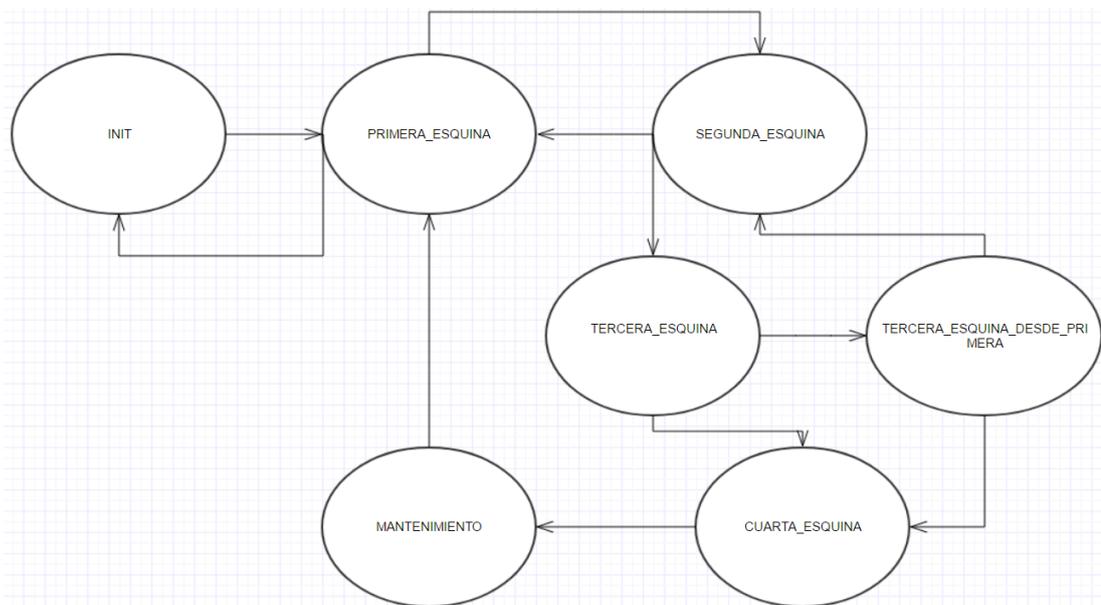


Figura 13 Máquina de estados del plan perceptivo

El primer estado y punto de partida es el estado INIT, este estado realiza una inicialización de todas las estructuras que utilizaremos durante el funcionamiento. Es posible volver a este estado si tenemos la necesidad de reiniciar la máquina de estados, en el caso de que los estados de modelado no sean capaces de extraer un modelo para el tablero de la mesa.

4.2.6.1 ESTRUCTURAS DE DATOS

A continuación se describen las estructuras de relevancia que se han utilizado

- Un vector para el almacenamiento de cada esquina visitadas para cada estado. Se ha definido un vector de esquinas para uno de los estados que generan el modelo inicial: PRIMERA_ESQUINA, SEGUNDA_ESQUINA, TERCERA_ESQUINA, TERCERA_ESQUINA_DESDE_PRIMERA. Cada vez que el método de modelado, partiendo de una esquina a , encuentra una esquina b que considera válida, se transforman sus coordenadas al sistema de referencia 3D del mundo, y se guarda en el vector de esquinas correspondiente al estado activo. Este vector de esquinas 3D es válido mientras que se considere que la esquina de partida a es válida. Si en algún momento se invalida, el vector de esquinas visitadas para este estado se vacía.
- Una imagen binaria para representar esquinas visitadas. Cada vez que entramos a uno de los estados de modelado antes de comenzar con la búsqueda de una nueva esquina, proyectamos todas las esquinas visitadas actualmente en el sistema de referencia 3D al sistema de referencia 2D. Entonces, para cada esquina con coordenadas (x, y) estableceremos valor 1 en el elemento de la imagen binaria cuya posición en la imagen se corresponde con las coordenadas (x, y) de la esquina.

4.2.6.2 PRIMERA_ESQUINA

Es el primer estado de la fase de modelado del objeto mesa. Este estado se considera que tiene éxito cuando encuentra una esquina perteneciente al vector de esquinas 2D que no haya sido visitada anteriormente. Si no tuviera éxito pasaríamos al estado INIT reiniciando la máquina de estados y se vacía su vector de esquinas 3D visitadas.

Existen varias situaciones que se pueden dar:

1. Tenemos una vista completa de la mesa por lo que la solución es escoger una de las esquinas siempre y cuando no sea visitada.
2. Se puede dar el caso de que, por la posición del robot, la orientación de la cámara, o porque la mesa es muy larga, únicamente tengamos una vista parcial de la mesa y no tengamos ninguna esquina de partida. La solución para este problema es cambiar la orientación de la cámara donde haya probabilidades de encontrar esquinas pertenecientes a la mesa.

Para manejar estas dos situaciones se ha desarrollado una máquina de estados que resuelve el problema.

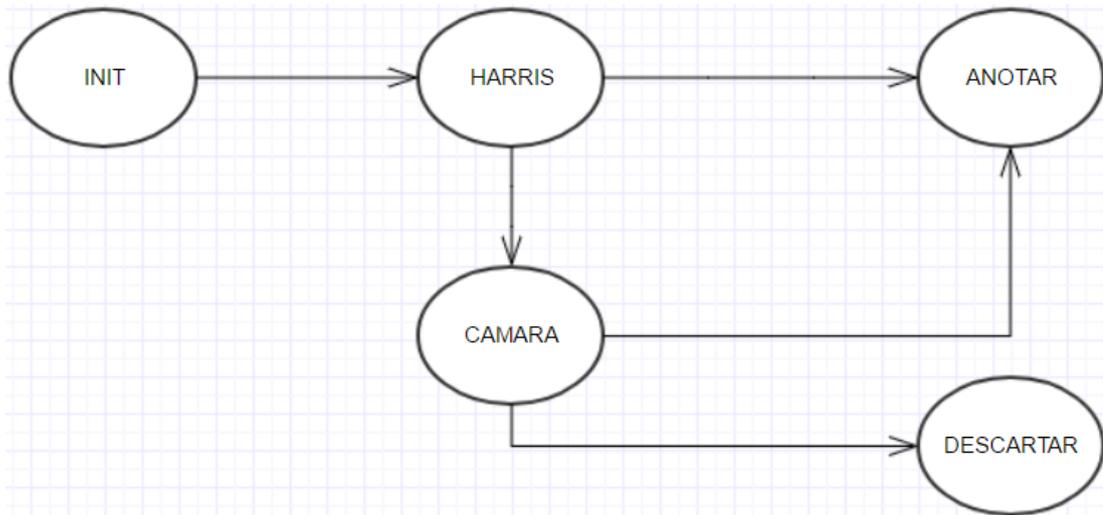


Figura 14 Máquina de estados para hallar la primera esquina del modelo.

El estado de partida es INIT, donde se realizan inicializaciones y cambia el estado a HARRIS.

El estado HARRIS maneja la primera situación, y si encuentra una esquina que no sea visitada pasa al estado ANOTAR. Si el vector de esquinas está vacío o todas las esquinas están visitadas la máquina de estados pasa al estado CAMARA.

El estado CAMARA, reorienta la cámara del robot en alguna dirección en la cual haya probabilidad de encontrar puntos de esquina de la mesa. Se ha desarrollado un método que, calculando las líneas de Hough para el frame actual y las correspondientes ecuaciones de recta, para cada recta calcula los puntos de corte con los límites de la imagen, y comprueba que cada uno de esos puntos son cercanos a puntos de borde y existen puntos vecinos que tienen una altura real perteneciente a un rango (min, max) predefinido. Si existe punto de corte, se centra la cámara en el punto de corte y sale de la máquina de estados sin cambiar el estado de la máquina principal.

Cuando vuelva a entrar en el estado PRIMERA_ESQUINA, se volverá a repetir el proceso descrito.

Cuando hay éxito, el estado pasa a ANOTAR, que establece la esquina encontrada como parte del modelo y avanza al estado SEGUNDA_ESQUINA en la máquina principal.

Si no hay éxito el estado DESCARTAR reinicia la máquina de estados inicial retrocediendo al estado INIT.

4.2.6.3 SEGUNDA_ESQUINA

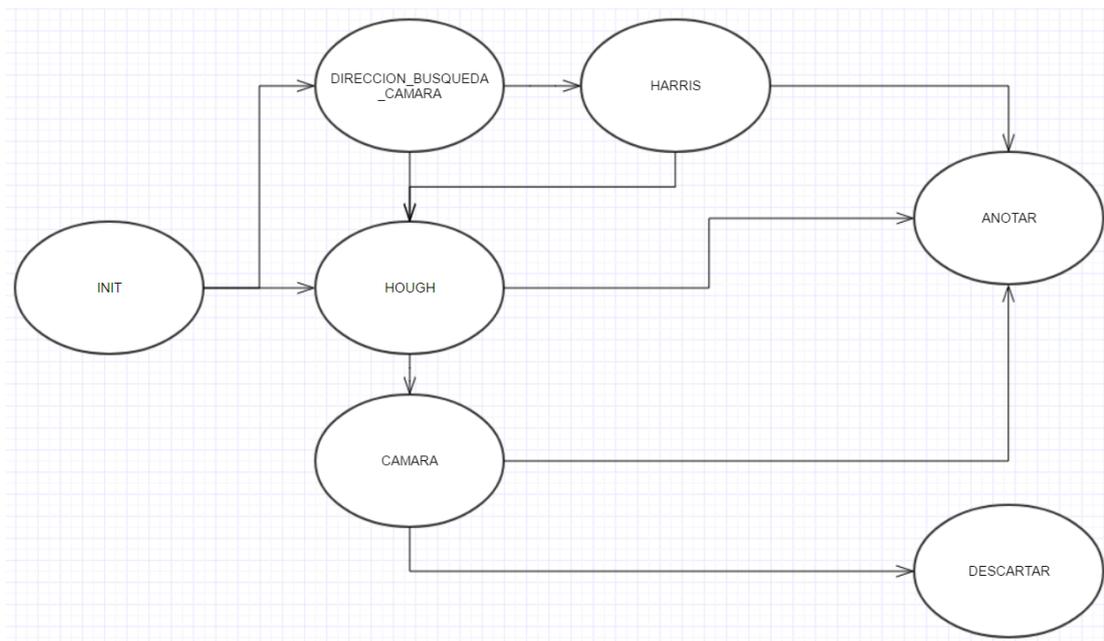


Figura 15 Máquina de estados para hallar la segunda esquina del modelo.

A continuación se definen las condiciones necesarias para que una esquina $C = (c_1, c_2)$ perteneciente al conjunto de esquinas detectadas por el detector de esquina se considere válida en el estado SEGUNDA_ESQUINA.

Sean $A' = (a'_1, a'_2, a'_3)$ la primera esquina real del modelo, y $A = (a_1, a_2)$ su proyección sobre el plano 2D, se procede a realizar la búsqueda de la segunda de las esquinas del modelo con coordenadas 2D y 3D $B = (b_1, b_2)$ y $B' = (b'_1, b'_2, b'_3)$ respectivamente, que cumpla las siguientes condiciones:

- B no pertenece al conjunto de esquinas visitadas
- Sea d con valor $D(A, B) = \sqrt{(b'_1 - a'_1)^2 + (b'_2 - a'_2)^2 + (b'_3 - a'_3)^2}$ el valor de la distancia entre los puntos A y B , se considera que B es una esquina

válida del modelo si d pertenece a un rango de valores determinado por el rango [distancia mínima, distancia máxima].

A continuación se detalla el funcionamiento de la máquina de estados. El punto de partida depende de si se movió la cámara o no.

Si la cámara no se ha movido, el punto de partida es el estado HOUGH, en el que se realiza una búsqueda de esquina utilizando rectas de Hough partiendo del punto $B = (b_1, b_2)$. En caso de encontrar una esquina que cumpla las condiciones propuestas, se avanza al estado ANOTAR donde se añade la esquina al modelo y marcamos la esquina como esquina visitada.

Si por el contrario no encontramos una esquina válida avanzamos al estado CAMARA, pues tenemos que descartar que estamos ante una vista parcial de la mesa, o reorientar la cámara si fuera necesario. Si existe un punto de corte $H_i = (h_1, h_2)$ con coordenadas 3D $H_i = (h'_{i1}, h'_{i2}, h'_{i3})$ perteneciente al conjunto de puntos de corte H de la recta r que pasa por el punto A , y cuyas coordenadas se corresponden con las coordenadas de un punto de borde en la imagen de bordes, se guardan las coordenadas 3D de $H_i = (h'_{i1}, h'_{i2}, h'_{i3})$, y se centra la cámara en H_i . Posteriormente sale de la máquina de estados sin alterar el estado de la máquina principal.

Después de salir al hilo principal para recuperar la nueva imagen de la cámara se vuelve a iniciar la máquina de estados del estado SEGUNDA_ESQUINA, sin embargo, el estado de partida es DIRECCION_BUSQUEDA_CAMARA. Este nuevo estado DIRECCION_BUSQUEDA_CAMARA proyecta el punto $H_i = (h'_{i1}, h'_{i2}, h'_{i3})$ para obtener $H_i = (h_1, h_2)$ y comprobamos si es una dirección de búsqueda válida, pues nos interesa buscar en la dirección que marca el punto donde se centró la cámara. Si es una esquina válida avanzamos al estado HARRIS donde se realiza una búsqueda con dirección desde la esquina de partida $A = (a_1, a_2)$ utilizando como dirección de búsqueda $H = (h_1, h_2)$. En caso de encontrar una esquina que cumpla las condiciones propuestas, se avanza al estado ANOTAR y añadimos la esquina al modelo y marcamos la esquina como esquina visitada. Si no se encuentra una esquina o si no se encontró una dirección de búsqueda válida, avanzamos al estado HOUGH y se repite lo anterior.

Si hay éxito, el estado avanza el estado de la máquina principal al estado TERCERA_ESQUINA.

4.2.6.4 TERCERA_ESQUINA y TERCERA_ESQUINA_DESDE_PRIMERA

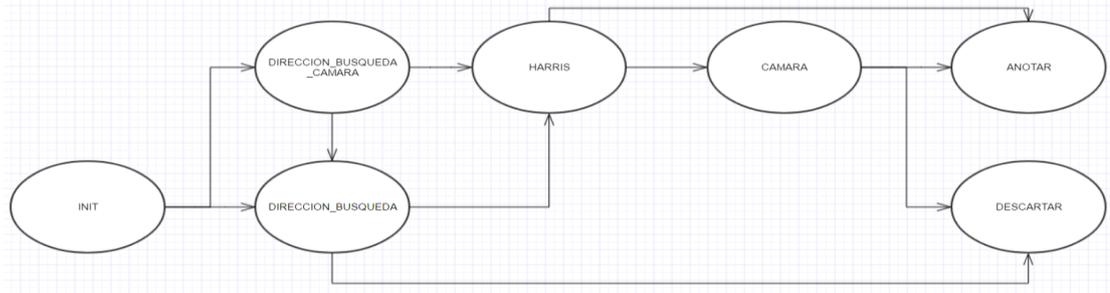


Figura 16 Máquina de estados para hallar la primera esquina del modelo.

A continuación se definen las condiciones necesarias para que una esquina $C = (c_1, c_2)$ perteneciente al conjunto de esquinas detectadas por el detector de esquina se considere válida en los estados TERCERA_ESQUINA y TERCERA_ESQUINA_DESDE_PRIMERA.

Sean $A' = (a'_1, a'_2, a'_3)$, $B' = (b'_1, b'_2, b'_3)$ las coordenadas reales para las esquinas 1 y 2 del modelo, respectivamente, y $A = (a_1, a_2)$ y $B = (b_1, b_2)$ sus proyecciones sobre el plano 2D, se procede a realizar la búsqueda de la tercera esquina del modelo con coordenadas 2D y 3D $C = (c_1, c_2)$ y $C' = (c'_1, c'_2, c'_3)$ respectivamente, que cumpla las siguientes condiciones:

- C no pertenece al conjunto de esquinas visitadas
- Sea $\overline{BA} = (a'_1 - b'_1, a'_2 - b'_2, a'_3 - b'_3)$ y $\overline{BC} = (c'_1 - b'_1, c'_2 - b'_2, c'_3 - b'_3)$ donde el ángulo α que forman los 2 vectores viene dado como:

$$\alpha = \text{acos} \left(\frac{|\overline{BA} \times \overline{BC}|}{|\overline{BA}| * |\overline{BC}|} \right)$$

Se tiene que cumplir que el ángulo α tiene un valor aproximado de $\pi/2$.

- Sea d con valor $D(B, C) = \sqrt{(c'_1 - b'_1)^2 + (c'_2 - b'_2)^2 + (c'_3 - b'_3)^2}$ el valor de la distancia entre los puntos B y C , se considera que C es una esquina

válida del modelo si d pertenece a un rango de valores determinado por el rango [distancia mínima, distancia máxima].

A continuación se detalla el funcionamiento de la máquina del estado para el estado TERCERA_ESQUINA.

El punto de partida depende de si se movió la cámara o no.

Si no se movió la cámara se realizará una “búsqueda con dirección” partiendo del estado DIRECCION_BUSQUEDA. El estado tiene éxito para un punto en la imagen $P = (p_1, p_2)$ donde G es el conjunto de puntos del entorno de P , si existe un punto $G_i = (G_{i1}, G_{i2})$ cuyas coordenadas se corresponden con las coordenadas de un punto de borde en la imagen de Bordes. Si no tiene éxito avanzamos al estado DESCARTAR y cambia el estado de la máquina principal a TERCERA_ESQUINA_DESDE_PRIMERA.

Si tiene éxito, la máquina de estados avanza al estado HARRIS donde se realiza una búsqueda con dirección desde la esquina de partida $B = (b_1, b_2)$ en la dirección y sentido que marca el vector \overrightarrow{BP} . En el caso de encontrar una esquina que cumpla las condiciones propuestas, se avanza al estado ANOTAR y añadimos la esquina al modelo y marcamos la esquina como esquina visitada.

Si por el contrario no encontramos una esquina válida, avanzamos al estado CAMARA, pues tenemos que descartar que estamos ante una vista parcial de la mesa, y, si fuera el caso, reorientar la cámara. Si existe un punto de corte $H_i = (h_1, h_2)$ perteneciente al conjunto de puntos de corte H de la recta r que pasa por los puntos B y P cuyas coordenadas se corresponden con las coordenadas de un punto de borde en la imagen de bordes, se guardan las coordenadas 3D $H'_i = (h'_{i1}, h'_{i2}, h'_{i3})$ de $H_i = (h_1, h_2)$, centramos la cámara en H_i y sale de la máquina de estados sin alterar el estado de la máquina principal.

Después de salir al hilo principal para recuperar la nueva imagen de la cámara, se vuelve a iniciar la máquina de estados del estado TERCERA_ESQUINA, sin embargo, el estado de partida es DIRECCION_BUSQUEDA_CAMARA (se intenta continuar el trazado que se detectó con la supuesta vista parcial de la mesa), pues hemos reorientado la cámara y nos interesa reanudar la búsqueda de la tercera esquina. Este nuevo estado DIRECCION_BUSQUEDA_CAMARA proyecta el punto $H_i = (h'_{i1}, h'_{i2}, h'_{i3})$ para obtener $H_i = (h_1, h_2)$ y comprobamos si es una

dirección de búsqueda válida, si no lo fuera avanzamos al estado DIRECCION_BUSQUEDA y se repite lo anterior.

El estado DESCARTAR del estado TERCERA_ESQUINA de la máquina principal, avanza el estado de la máquina principal a TERCERA_ESQUINA_DESDE_PRIMERA, donde el proceso se repite con el mismo esquema pero partiendo de la primera esquina. Es aquí donde si hay éxito en la búsqueda de esquina retrocedemos al estado SEGUNDA_ESQUINA de la máquina principal.

Si hay éxito, el estado ANOTAR dentro de los estados TERCERA_ESQUINA y TERCERA_ESQUINA_DESDE_PRIMERA avanza el estado de la máquina principal al estado CUARTA_ESQUINA.

Para aumentar el rendimiento del modelado de la mesa, para determinar las coordenadas de la cuarta esquina se realiza un cálculo entre las 3 esquinas ya obtenidas como se explica en apartados anteriores.

4.2.6.5 MANTENIMIENTO

Sea M el conjunto de esquinas que conforman el modelo de la mesa, para cada elemento i del conjunto M se configura el detector de esquinas para obtener $C1_i$ y $C2_i$ que son conjuntos de esquinas detectadas con Harris ordenadas por calidad de altura y por calidad de ángulo respectivamente.

Se procede a realizar el ajuste de altura, para ello. Para cada esquina i de M se busca una esquina k perteneciente a $C1_i$ que produzca una mejora de altura. Si no existe una esquina que produzca una mejora se mantiene la actual esquina en el modelo.

Una vez se ha realizado el ajuste de altura, el siguiente ajuste pretende maximizar las dimensiones de la mesa, pues partimos de la suposición que durante el modelado existe la probabilidad de que una de las esquinas del modelo las genera un objeto sobre la mesa.

Basándonos en esta suposición, sea $A' = M'_i = (a'_1, a'_2, a'_3)$ una esquina perteneciente al conjunto M de esquinas que conforman el modelo de la mesa y $B' = M'_{i-1} = (b'_1, b'_2, b'_3)$ y $C' = M'_{i+1} = (c'_1, c'_2, c'_3)$ sus esquinas adyacentes, se busca maximizar la longitud de \overrightarrow{AB} y \overrightarrow{AC} .

Para realizar la optimización, se busca una esquina perteneciente a $C2_i$ que mejore el modelo. Como vienen ordenadas por la esquina que produce la mayor variación de uno de los ángulos asociados a las esquinas adyacentes, lo más probable es que siempre se escoja el primer elemento de $C2_i$ como esquina que mejora el modelo.

Se puede dar el caso de que $C2_i$ sea un conjunto vacío, lo que significa que no se puede mejorar ni mantener con esquinas de Harris, entonces para validar A se tiene que cumplir que para $A' = (a'_1, a'_2, a'_3)$ cuya proyección sobre el plano 2D es $A = (a_1, a_2)$, donde CB es el conjunto de puntos de borde y CE es el conjunto de puntos del entorno cercano de A, existe un elemento $CB = (cb_1, cb_2)$ perteneciente al conjunto CE y un punto $CE = (ce_1, ce_2)$ perteneciente al conjunto CB donde $cb_1 == ce_1$ y $cb_2 == ce_2$. Si no se verifica, se anota un fallo.

Si existe un fallo, el mantenimiento del modelo recalcula la esquina que no se pudo verificar a partir de las otras 3.

Si existe más de un fallo, se descarta el modelo y cambia el estado de la máquina principal a INIT.

5 RESULTADOS

5.1 ESCENARIOS

Durante la fase de desarrollo se han definido 6 escenarios donde el plan perceptivo desarrollado se encargaba de extraer y mantener, a lo largo del tiempo, un modelo de una mesa.

Este apartado detalla los diferentes escenarios utilizados para poder hacer referencia a ellos cuando sea necesario. La imagen de la izquierda de cada par se corresponde con las esquinas que se detectan en la misma altura que las esquinas del modelo extraído, y la imagen de la derecha es la vista de la mesa sobre la que proyectamos el modelo extraído (la malla de color rojo).

1.1.1.1 Escenario 1

La Figura 17 representa el escenario más básico, del partimos de una vista completa de una mesa sin objetos.

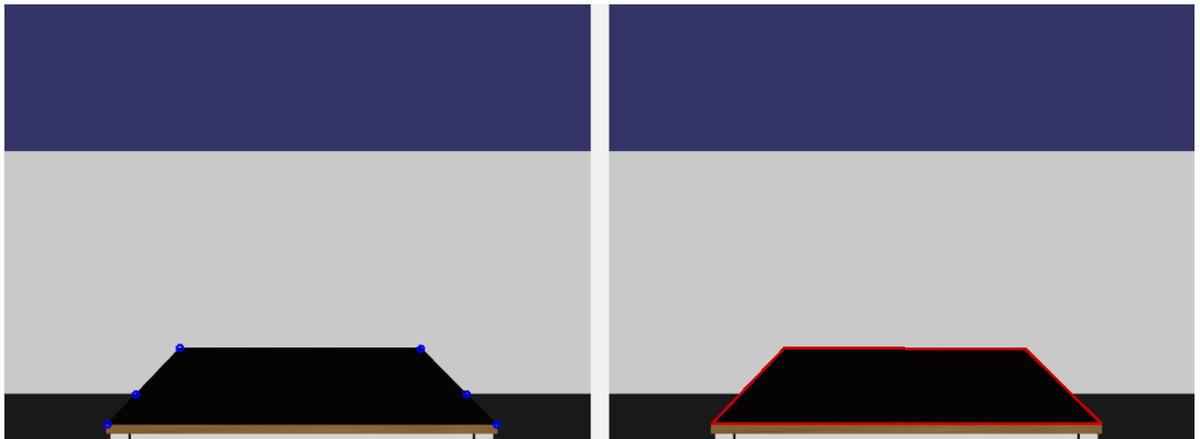


Figura 17 Modelo del escenario más básico.

Escenario 2

La figura 18 muestra el escenario 2, el primer escenario con objetos, el objetivo de este escenario es analizar el comportamiento ante las primeras esquinas falsas producidas por los objetos de la mesa.

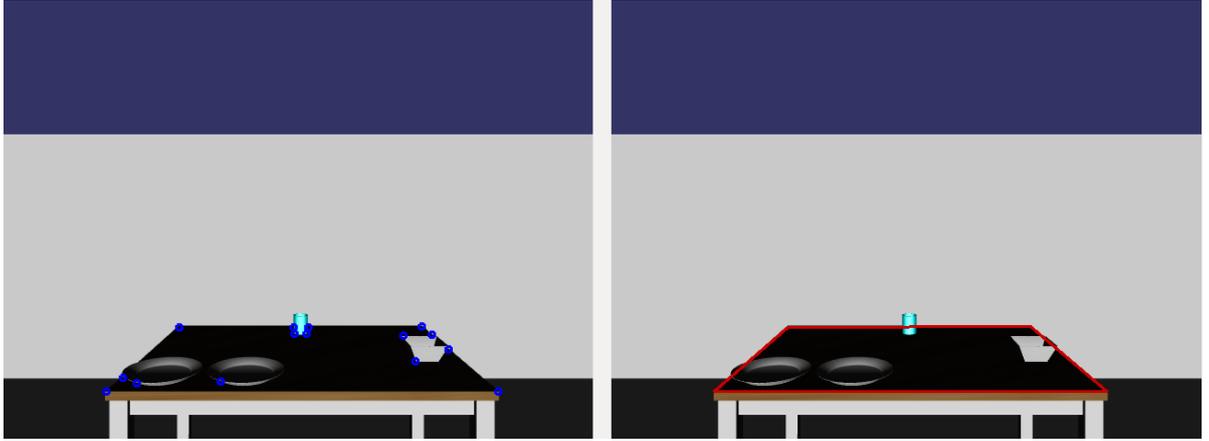


Figura 18 Modelo del escenario con algunos objetos sobre la mesa.

1.1.1.2 Escenario 3

El siguiente escenario (Figura 19) añade más objetos a la mesa y además comienza con la cámara inclinada 0.4 radianes.

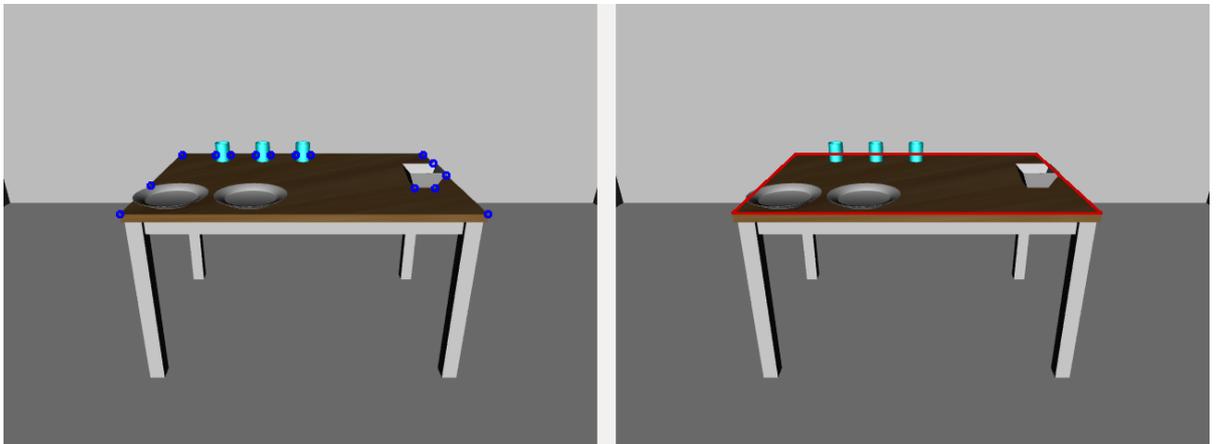


Figura 19 Modelo del escenario con algunos objetos sobre la mesa e inclinación de cámara.

1.1.1.3 Escenario 4

En este escenario (Figura 20) se aumenta aún más el número de objetos en relación al escenario anterior.

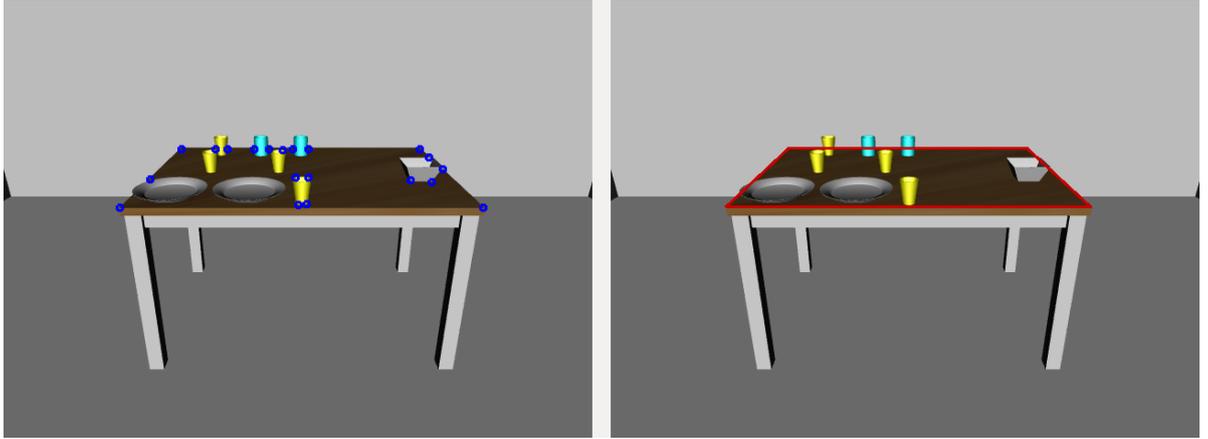


Figura 20 Modelo del escenario con objetos adicionales.

1.1.1.4 Escenario 5

Este escenario es el que más objetos tiene encima de la mesa. Ha sido el escenario referente para la mayoría de pruebas finales, pues es el que mayor complejidad ofrece y el que, junto al siguiente y último escenario, más carga computacional presenta a la hora de estimar el modelo inicial.

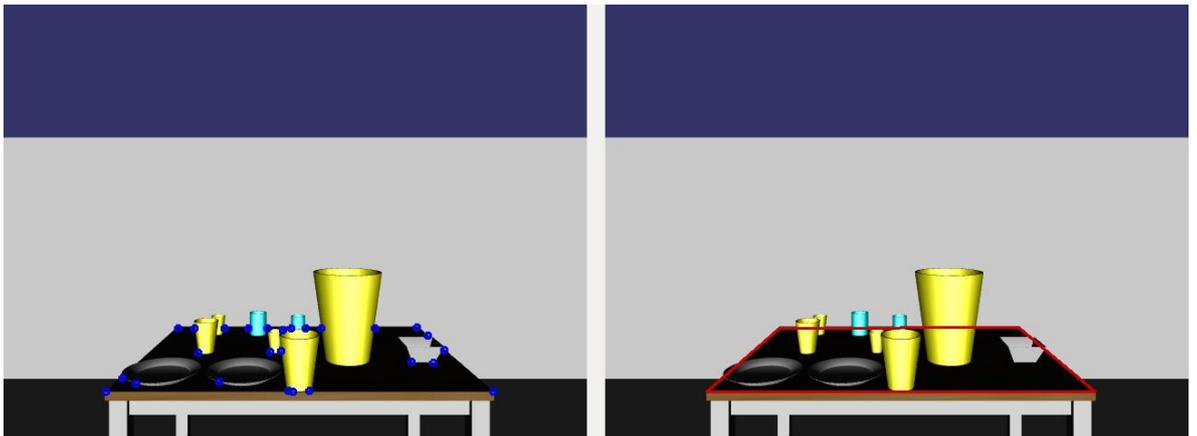


Figura 21 Modelo del escenario con muchos objetos sobre la mesa.

Aunque la primera imagen de la figura 21 no parezca ofrecer mucha complejidad en el cálculo de esquinas, la realidad es la que se muestra en la Figura 22.

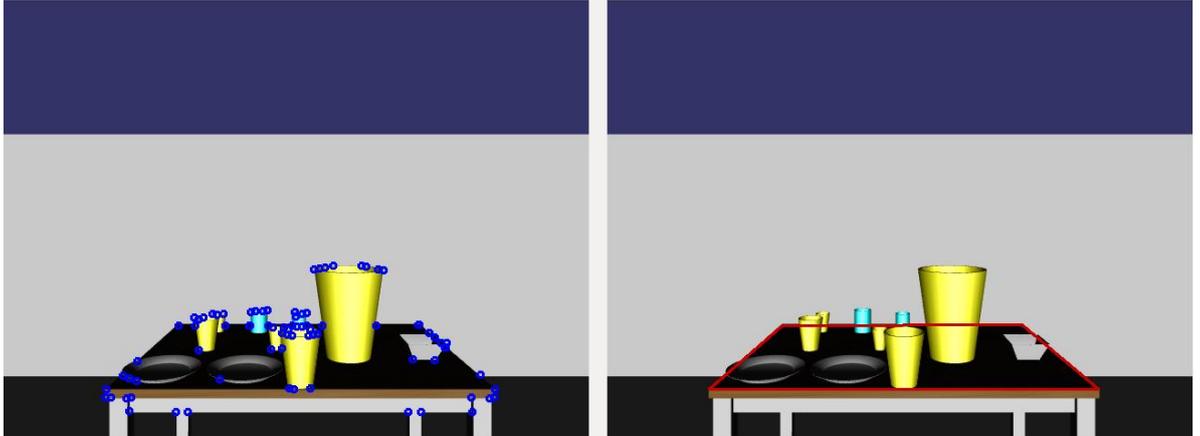


Figura 22 Escenario con muchos objetos, vista de las esquinas sin filtrado de altura.

La Figura 21, nos ofrece una vista de las esquinas que podrían haber generado una hipotética mesa. Son las esquinas que están a la misma altura que las esquinas del modelo. Sin embargo, la realidad a la que se enfrenta el sistema perceptivo es la que se muestra en la figura 22. Como la mesa larga ofrecía problemas de detección de esquinas, fue necesario bajar el umbral de Harris en la detección de esquinas, produciendo más esquinas de textura.

1.1.1.5 Escenario 6

Este escenario comienza con una vista parcial de la mesa, sin tener ninguna esquina real de la mesa en su campo de visión. Este escenario está obligado a manejar la situación en la que movemos la cámara sin partir de ninguna esquina ayudándose de las intersecciones de las rectas de Hough [13] con los límites de la imagen: lateral izquierdo, inferior y lateral derecho. Como el robot es más alto que la mesa, no se comprueba el límite superior de la imagen.

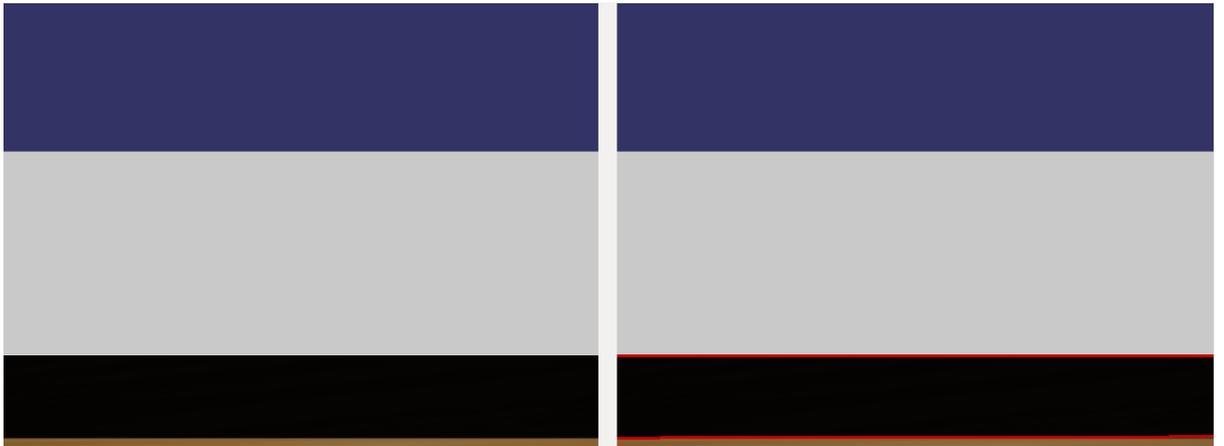


Figura 23 Mesa larga, vista inicial.

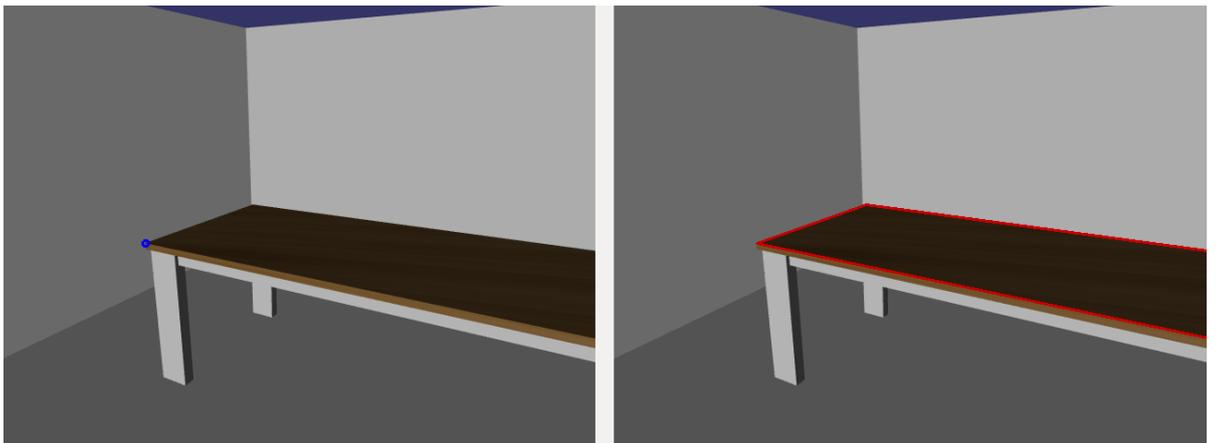


Figura 24 Mesa larga vista después de realizar el modelado.

5.2 PROBLEMAS, SOLUCIONES Y RESULTADOS

5.2.1 Generación del modelo

La aproximación inicial del modelado de la mesa, partía de cualquiera de las esquinas de la mesa para realizar una búsqueda utilizando el método por direcciones descrito en el apartado de metodología.

El desarrollo inicial se realizó sobre el escenario 1, una mesa pequeña sin objetos y con una vista completa de la misma. Surgieron problemas con los siguientes escenarios. El sistema en un principio no soportaba la carga computacional debido al número de esquinas que se generaba. A continuación se muestran unas tablas en la

que se recogieron los tiempos en momentos específicos del desarrollo. Mediante la introducción de optimizaciones en el funcionamiento de la máquina se obtuvieron las mejoras de tiempo durante el modelado que se perseguían. La primera optimización propuesta fue introducir una nueva primitiva de alto nivel que nos brindara información extra, las líneas de Hough. Fue en este punto donde se incluyó la búsqueda de esquinas utilizando líneas de Hough, que se detalla en el apartado de metodologías, sustituyendo la búsqueda con direcciones en el estado SEGUNDA_ESQUINA de la máquina de estados inicial. Se puede observar como lo que era una optimización empeoraba los tiempos en todos los casos (tabla 2).

Tabla 1 Tabla de tiempos para los 5 primeros escenarios.

	Obtener imagen	Cálculo de datos	Tiempo medio por frame	Nº vueltas	Tiempo total
Escenario 1	~95	57	58,7	17	4419
Escenario 2	~95	58	26,68	19	4883
Escenario 3	~95	77	33,57	21	5135
Escenario 4	~95	79	26,78	19	4815
Escenario 5	~95	211	80,73	52	20520

Tabla 2 Se añade la búsqueda de esquinas con Hough.

	Obtener imagen	Cálculo de datos	Tiempo medio por frame	Nº vueltas	Tiempo total
Escenario 1	~95	50	69,54	35	8462
Escenario 2	~95	76	58,26	64	17471
Escenario 3	~95	75	57,17	64	16681
Escenario 4	~95	100	60,25	64	17333
Escenario 5	~95	196	87,25	51	26655

En la tabla 2 se puede observar como aumenta el número de vueltas, o de iteraciones necesarias con los estados del modelado para extraer el modelo con la versión que utiliza Hough para el cálculo de la segunda esquina.

Como los tiempos que ofrecía el cálculo de esquinas mediante Hough son insignificantes, del orden de 1ms, se hizo la suposición de que el problema residía en la búsqueda de la tercera esquina.

La primera solución fue optimizar el algoritmo de análisis de puntos que se realiza en la búsqueda con dirección, haciendo más restrictiva la búsqueda, sin perder eficacia. El resultado de la optimización fue que se analizaban menos puntos de la imagen para llegar a una misma esquina, produciendo una notoria mejora en el tiempo de cálculo del modelo como se puede observar en la tabla 3.

Tabla 3 Se optimiza el análisis de puntos en la búsqueda con dirección.

	Obtener imagen	Cálculo de datos	Tiempo medio por frame	Nº vueltas	Tiempo total
Escenario 1	~95	53	17.11	35	7571
Escenario 2	~95	76	20.98	64	14891
Escenario 3	~95	75	21.53	64	15101
Escenario 4	~95	98	21.14	64	16449
Escenario 5	~95	196	13.37	51	18111

Se puede observar sin embargo que el número de vueltas y los tiempos de cálculo siguen siendo muy altos. Mediante pruebas, nos dimos cuenta de que había ocasiones en que no se calculaba la dirección de búsqueda para hallar la tercera esquina, por lo que se hizo una corrección y una posterior optimización como se muestra en las tablas 4 y 5.

Tabla 4 Tiempos después de corregir el cálculo de direcciones.

	Obtener imagen	Cálculo de datos	Tiempo medio por frame	Nº vueltas	Tiempo total
Escenario 1	~95	58	14,31	35	6703
Escenario 2	~95	81	18,67	64	13155
Escenario 3	~95	82	20,2969	64	13496
Escenario 4	~95	73	18,0781	64	14271
Escenario 5	~95	57	10,74	51	15268

Tabla 5 Tiempos después de optimizar el cálculo de direcciones.

	Obtener imagen	Cálculo de datos	Tiempo medio por frame	Nº vueltas	Tiempo total
Escenario 1	~95	71	16,34	23	4651
Escenario 2	~95	78	19,85	34	7535
Escenario 3	~95	79	20,17	34	7906
Escenario 4	~95	67	19,02	34	8014
Escenario 5	~95	73	11,02	39	12429

La optimización ofrece una mejora del tiempo de cálculo y en el número de vueltas, pero seguía siendo demasiado ineficiente. Esto es debido a un problema que ya existía, no se priorizaban las esquinas de partida con ningún criterio.

El problema se redujo entonces a buscar una variable extra, que permitiera priorizar que esquinas escoger para emprender la búsqueda de la segunda esquina. La solución fue utilizar otra vez información que nos ofrece las líneas de Hough, priorizando aquellas esquinas cuya posición en la imagen es cercana a un punto de intersección entre rectas de Hough (tabla 6).

Tabla 6 Priorizando esquinas de partida.

	Obtener imagen	Cálculo de datos	Tiempo medio por frame	Nº vueltas	Tiempo total
Escenario 1	~95	71	16,34	23	4651
Escenario 2	~95	78	19,85	34	7535
Escenario 3	~95	79	20,17	34	7906
Escenario 4	~95	67	19,02	34	8014
Escenario 5	~95	73	11,02	39	12429

A continuación se muestra una secuencia del cálculo del modelo para el escenario n° 5, después de realizar las optimizaciones y arreglos anteriores.

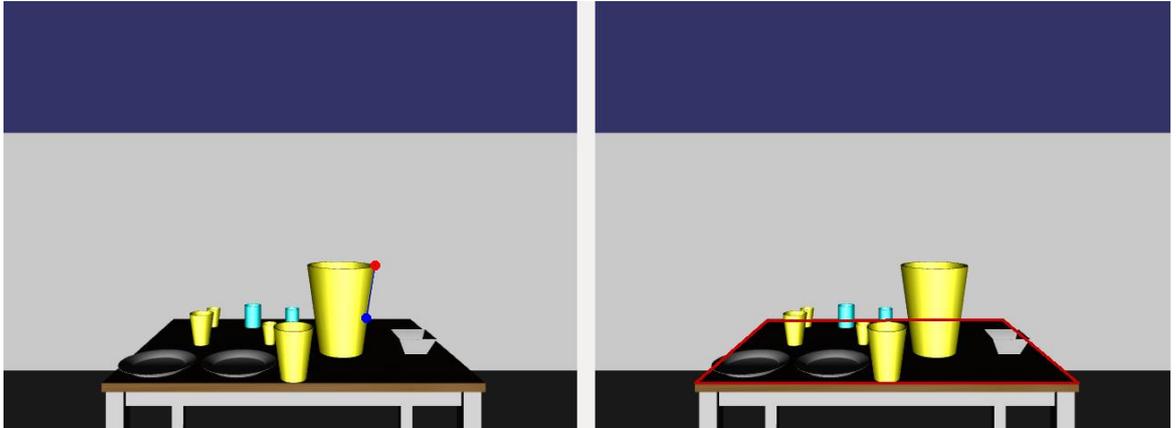


Figura 25 Se parte de una esquina no valida (esquina azul).

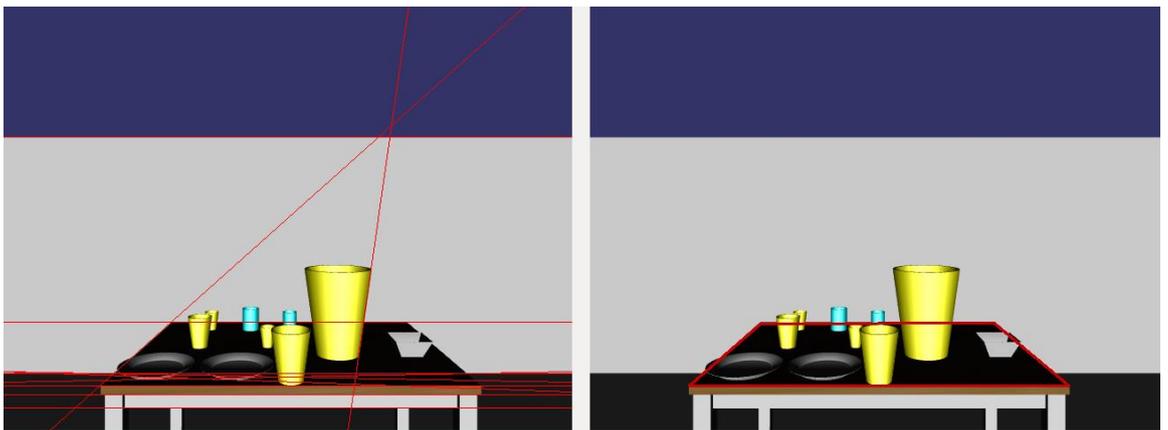


Figura 26 Intersecciones entre rectas de Hough de la situación de la figura 25.

La selección de la falsa esquina se lleva a cabo por su cercanía a una de las intersecciones.

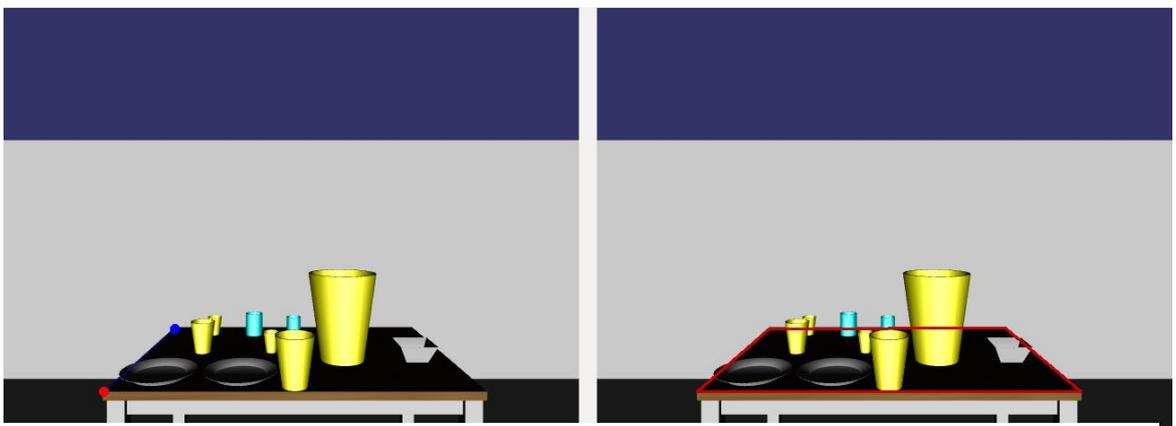


Figura 27 Se parte de otra esquina cercana a una intersección (ver figura 26).

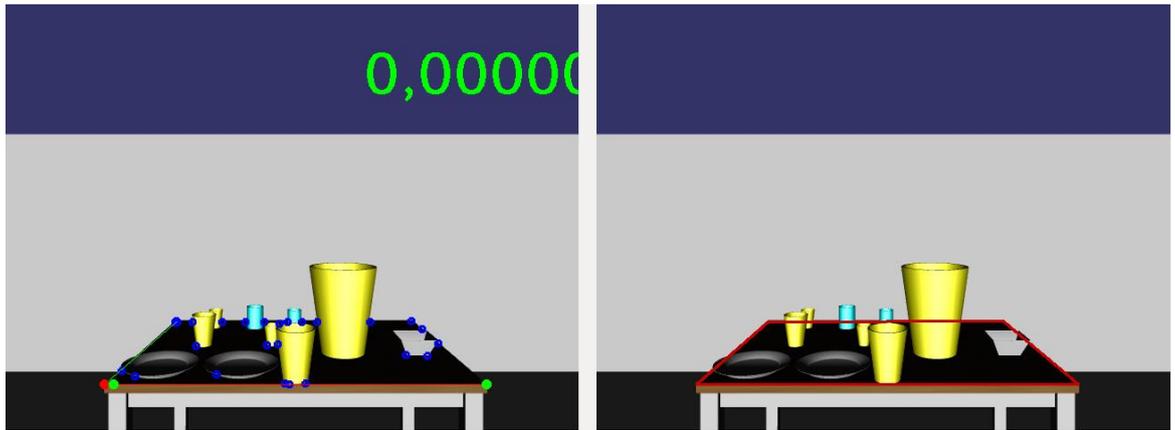


Figura 28 Se alcanza la tercera esquina (el punto verde junto al rojo es la dirección de búsqueda)

5.2.2 Reorientación de la cámara durante el modelado

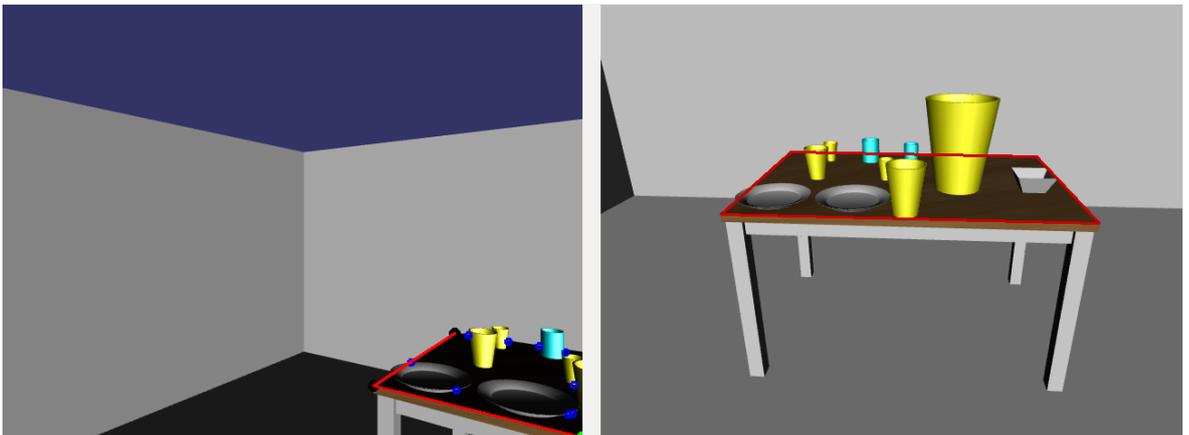


Figura 29 Ejemplo de detección de vista parcial de la mesa, en el escenario 5.

Uno de los casos más importantes que resuelve el sistema desarrollado es la detección del modelo cuando existen vistas parciales de la mesa. Esto nos permite resolver un mayor rango de situaciones sin la necesidad de reorientar manualmente la base del robot.

La figura 29 representa un caso para el cual, el robot solo tiene la vista de dos esquinas de la mesa. La solución propuesta para este caso fue apoyarnos de nuevo en las líneas de Hough.

Aprovechamos las intersecciones de todas las líneas calculadas a partir de la transformada de Hough con los límites de la imagen, para determinar si fuera necesario si estamos ante una vista parcial y necesitamos reorientar la cámara en algún punto del recorrido.

La imagen izquierda de la figura 29, partiendo de una esquina inicial encuentra satisfactoriamente la segunda de las esquinas de la mesa. Sin embargo, durante la búsqueda de la tercera esquina no tiene éxito pero detecta que existe punto de borde a la altura de las demás esquinas que es cercano a una de las intersecciones (la marcada en verde en la primera imagen).

Se procede a reorientar la cámara centrándola en el punto de corte. El resultado es el de la imagen derecha de la figura 40, donde se extrae el modelo de la mesa.

El poder orientar la cámara durante el modelado permite resolver el escenario 6, cuya principal característica, es que el robot parte sin ninguna esquina real en su campo de visión.

El escenario 6 se presenta en las primeras etapas de desarrollo con muchos problemas de rendimiento, debido a que una mala decisión provocaba que se moviera la cámara del robot.

Durante las pruebas se pudo observar que, la selección de la primera esquina, afectaba mucho al rendimiento, pues si partimos de esquinas no deseadas, la cámara se reorientará muchas veces haciendo ineficiente la extracción del modelo.

La mejor alternativa es ordenar la lista de candidatas a primera esquina por su altura real, evitando así, seleccionar las esquinas inferiores al tablero. Sin embargo, en los escenarios anteriores se priorizaban las esquinas cercanas a un punto de corte entre rectas de Hough, para evitar en la medida de lo posible escoger esquinas producidas por objetos. Como se puede observar en la figura 30 este criterio afecta negativamente en el escenario 6, pues escogemos esquinas que no pertenecen al modelo.

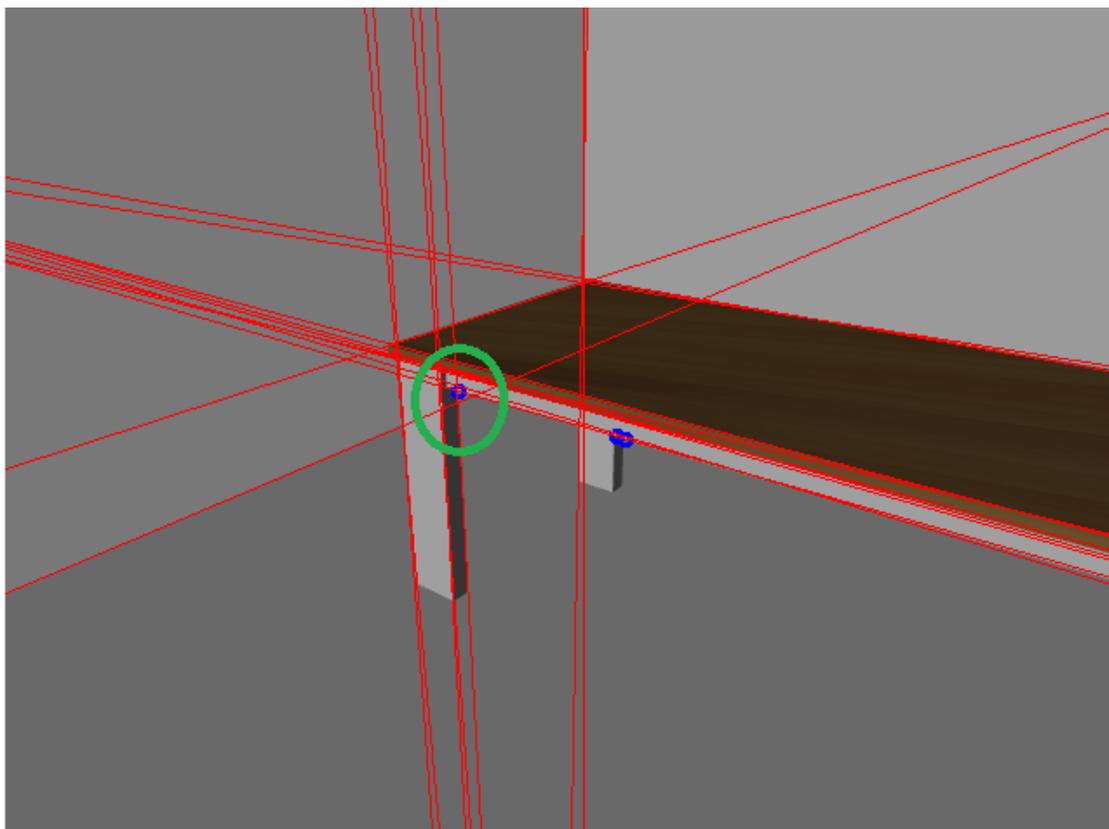


Figura 30 Ejemplo de esquina no deseada priorizada por intersecciones entre rectas de Hough ocasionando la generación de esquinas únicamente a su altura. Mientras que en los demás escenarios se descartan con facilidad, en el escenario 6 tiene un fuerte impacto en el rendimiento

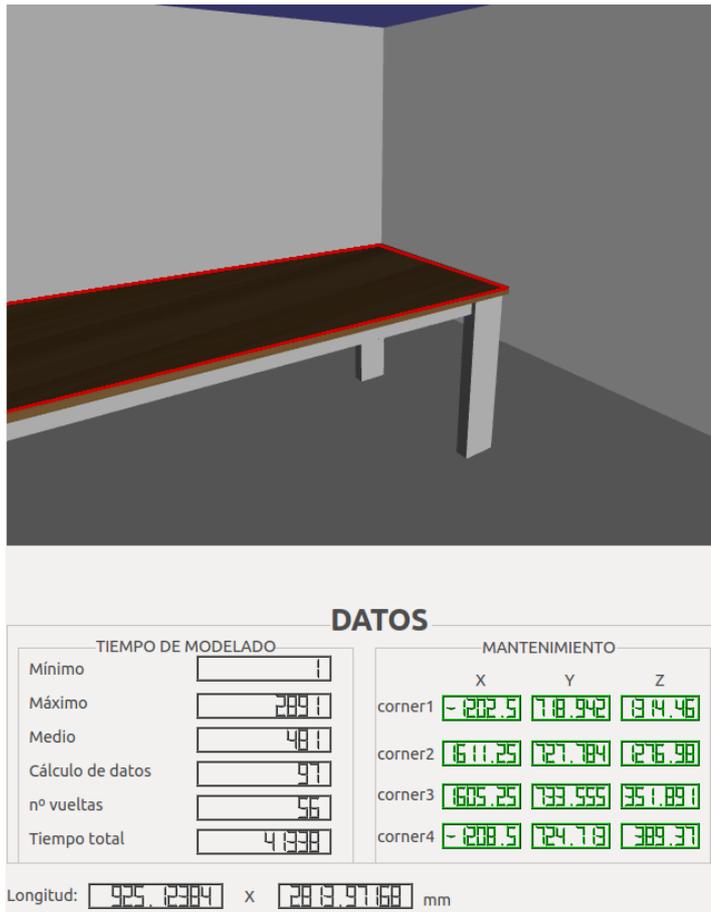


Figura 31 Resultado de las pruebas del escenario 6, donde se priorizan las esquinas cercanas a puntos de corte entre rectas de Hough.

Como se muestra en la figura 31, el tiempo total del cálculo del modelo es de 41 segundos. Para optimizar este tiempo, solo se priorizarán los puntos de corte como punto de partida cuando se detecten muchas esquinas (procedentes de otros objetos), priorizando en caso contrario la altura de las esquinas.

Aplicando la optimización la extracción del modelo se realiza en 7.6 segundos (figura 32).

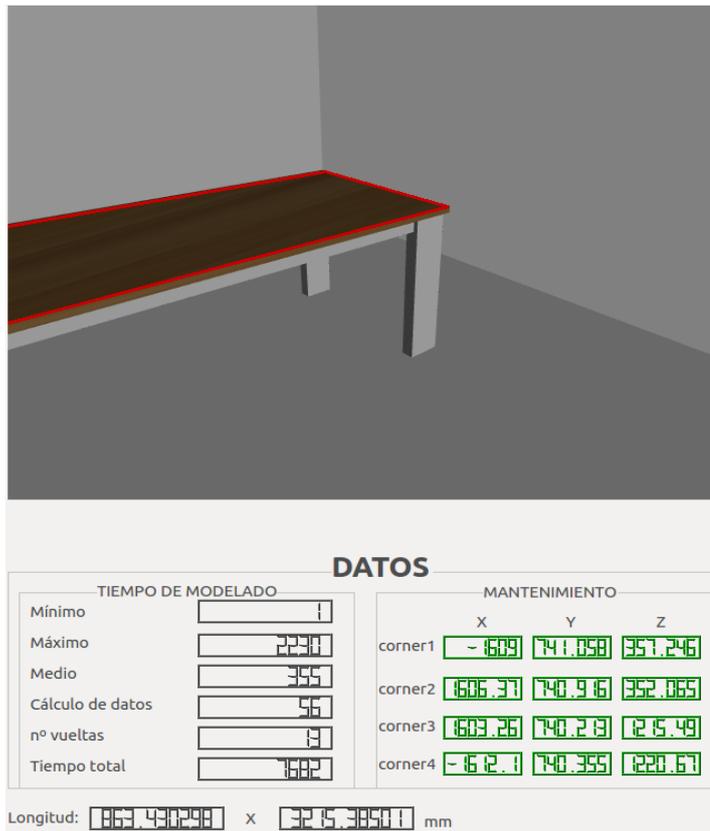


Figura 32 Resultados de las pruebas en el escenario 6 priorizando las esquinas cercanas a puntos de corte entre rectas de Hough.

A continuación se muestra una secuencia de la extracción del modelo para el escenario nº 6, después de realizar las optimizaciones y arreglos anteriores.

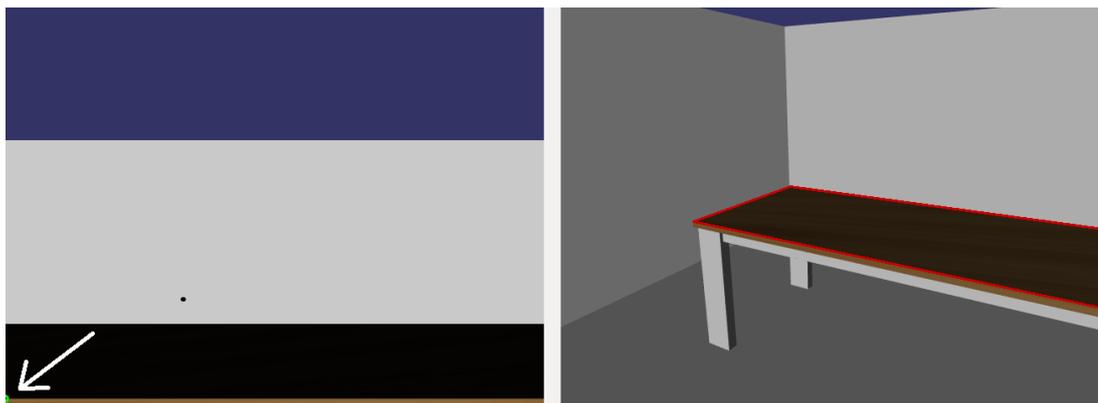


Figura 33 La máquina de estados detecta una vista parcial de la mesa sin esquinas de partida. Se centra la cámara en el punto indicado por la imagen de la izquierda.

Cuando tenemos la situación de la figura 33, se orienta la cámara utilizando los puntos de corte calculados a partir de la intersección de rectas de Hough con los límites de la imagen. Escogiendo en este caso el punto de corte señalado en la imagen izquierda.

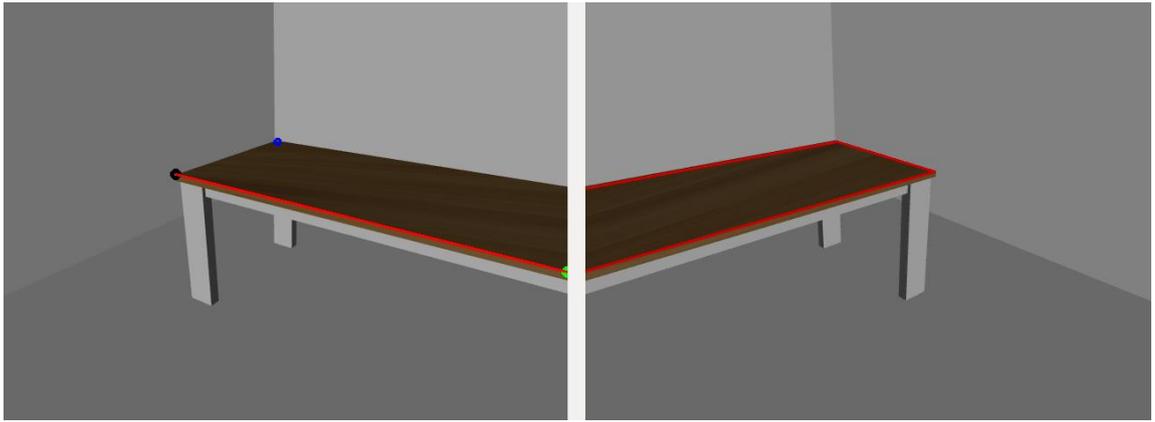


Figura 34 Se escoge una esquina de partida (la esquina negra), y se busca la segunda esquina del modelo. Se detecta una vista parcial de la mesa y se centra en el punto de corte (el punto verde).

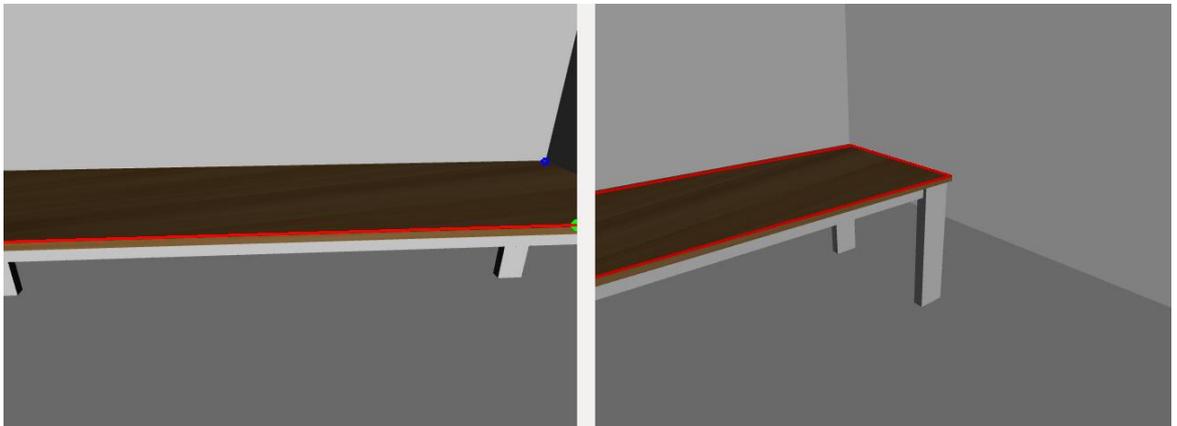


Figura 35 Una vez centrada la cámara se retoma la búsqueda de la segunda esquina, pero se vuelve a detectar una vista parcial de la mesa. Volvemos a centrar la cámara (punto verde de la imagen izquierda).

Como se ha explicado, al tener dimensiones tan grandes la cámara requiere orientarse varias veces para llegar de una esquina a otra cuando están situadas en el extremo del largo de la mesa (Figura 35). Se prioriza que el nuevo punto donde se centrará la cámara continúe el recorrido.

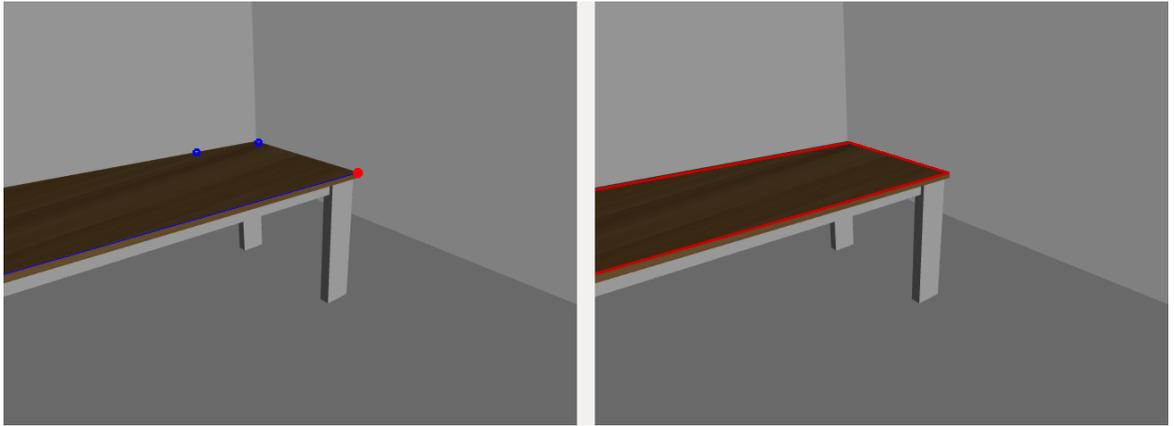


Figura 36 Se alcanza la segunda esquina (color rojo).

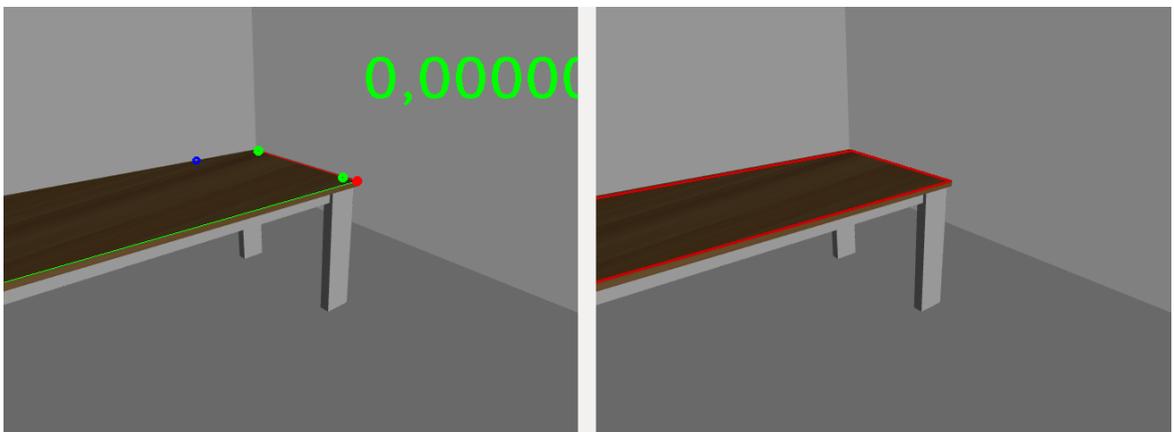


Figura 37 Se alcanza la tercera esquina y se calcula el modelo resultante (imagen derecha).

Como se puede apreciar, la búsqueda de las esquinas 1 y 2, que se encontraban adyacentes en el largo de la mesa, es la que más pasos ha consumido con un tiempo total de extracción del modelo de 7.6 segundos (figura 32).

Si se hace una mala elección de las esquinas de partida la extracción del modelo será muy ineficiente (figura 31).

5.2.3 Mantenimiento del modelo

Para mostrar la importancia de la fase de mantenimiento, a continuación se plantea una situación en la que el modelo inicial generado para la mesa no es correcto y éste es mejorado durante el mantenimiento. Como se puede apreciar en la figura 37, una de las esquinas está tapada por un objeto escogiendo así una esquina producida por el mismo objeto.

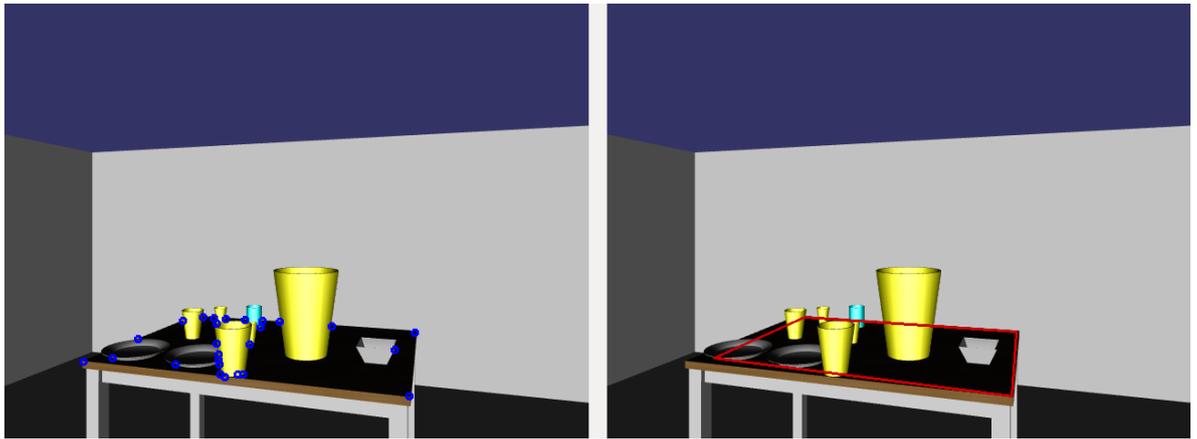


Figura 38 Situación en la que el modelo inicial generado por el sistema presenta incorrecciones.

El robot necesita moverse para ver las esquinas reales de la mesa y mejorar modelo. A continuación se ilustran dos imágenes con datos sobre las dimensiones del modelo extraído y el modelo mejorado.

En el apartado metodologías se explica que el modelo se ha codificado de tal manera que tiende a expandirse si puede, y eso es lo que hará en este caso cuando lo movamos.

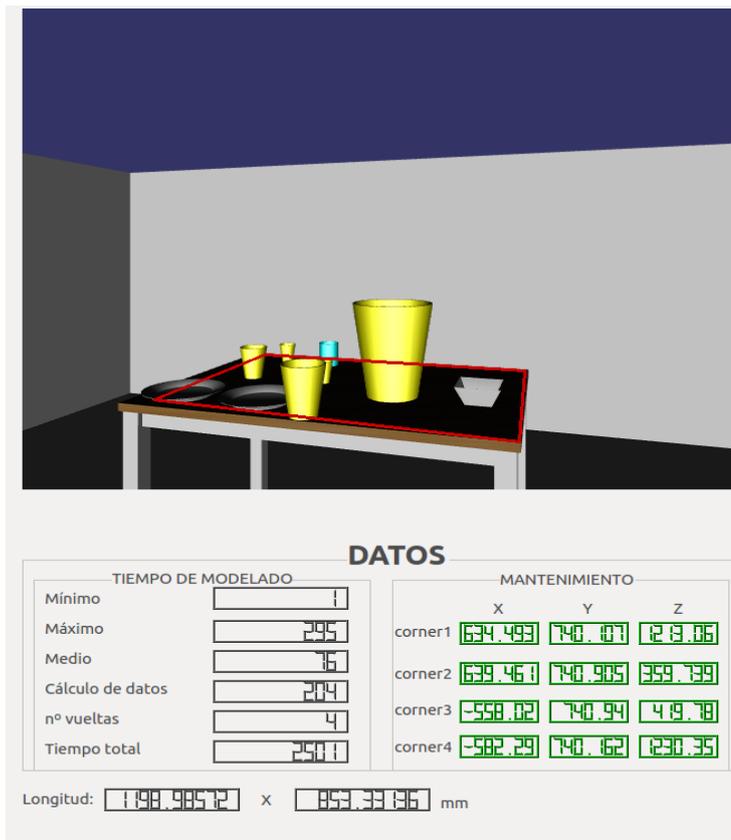


Figura 39 Datos del modelo erróneo .

En la figura 39 se muestran los datos sobre las dimensiones del modelo, así como las coordenadas de las esquinas que conforman la malla del modelo. Y en la figura 40 se muestra la imagen del mismo caso después de mover el robot.

La imagen 39 representa un ejemplo de un modelo inicial donde la malla que representa el modelo no se corresponde con las dimensiones reales del tablero.

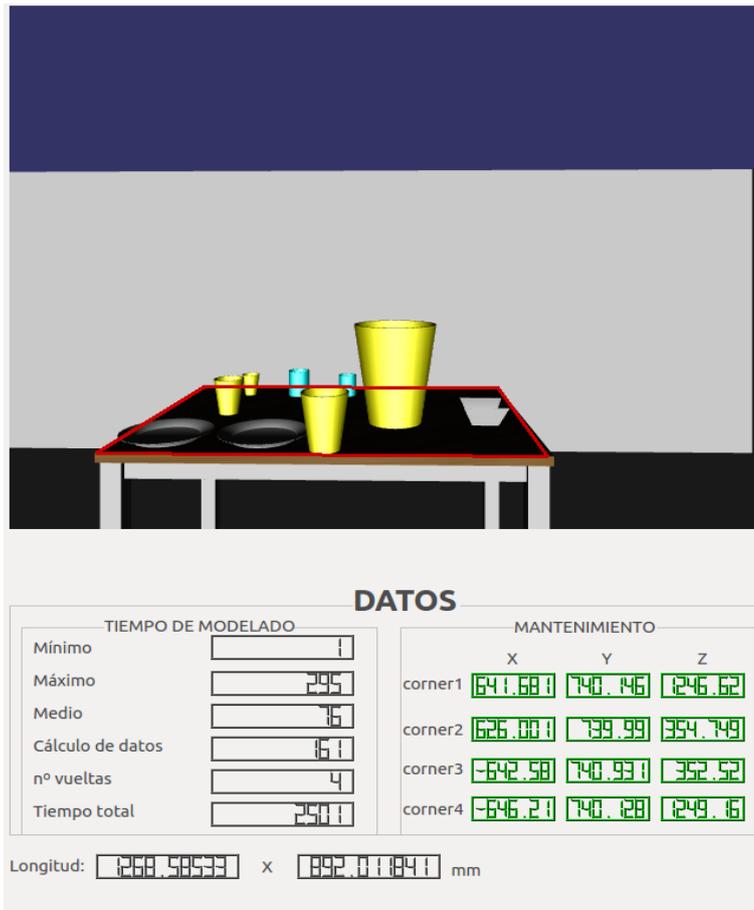


Figura 40 Datos del modelo corregido.

La figura 40 muestra las nuevas características del modelo después de haber sido mejorado por el estado de mantenimiento, tras de mover la base y detectar una mejor configuración que satisface las restricciones establecidas y descritas en el apartado de metodologías.

Se puede apreciar cómo las dimensiones de la mesa han aumentado y ahora la malla abarca todo el tablero de la mesa.

5.2.4 Problema de los ángulos obtusos en Harris

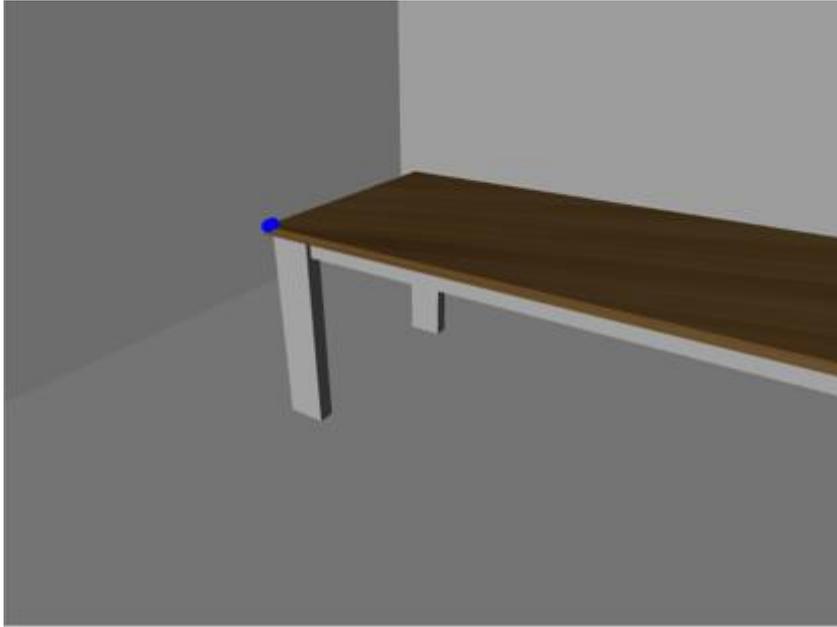


Figura 41 Ejemplo de vista que no detecta bien las esquinas mediante Harris.

Uno de los problemas del método de detección de esquinas de Harris son los ángulos obtusos. En su entorno local, las esquinas con ángulos obtusos sólo presentan cambios de intensidad significativos en una sola dirección. Esta limitación llevaba a los resultados de detección que se muestran en la figura 41. La solución planteada ha sido crear otro conjunto de esquinas de apoyo durante la fase de modelado, utilizando los puntos de intersección de rectas de Hough.

A continuación se detallan los pasos seguidos utilizando el escenario 6, que era el escenario que más sufría este problema debido a las dimensiones de la mesa.

El sentido de la siguiente prueba es encontrar una configuración óptima para incorporar las esquinas de Hough al modelado junto a las esquinas de Harris.

Para la primera prueba se eliminan las esquinas de Harris y hallamos las intersecciones entre rectas que forman un ángulo $\alpha > 0.2$, como se ilustra en la figura 42 y posteriormente se realiza la supresión del no máximo, incorporándose previamente las esquinas de Harris.

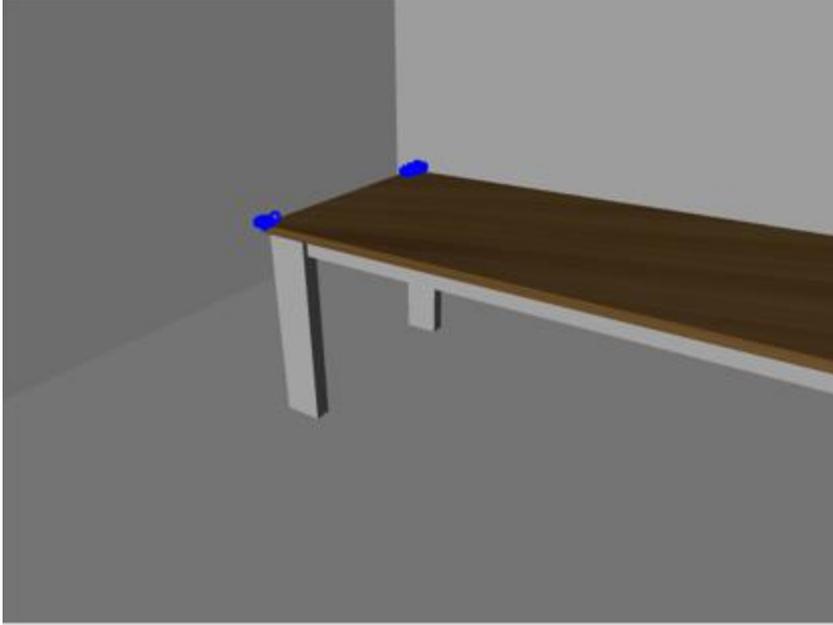


Figura 42 Esquinas de Hough sin supresión de no máximo.

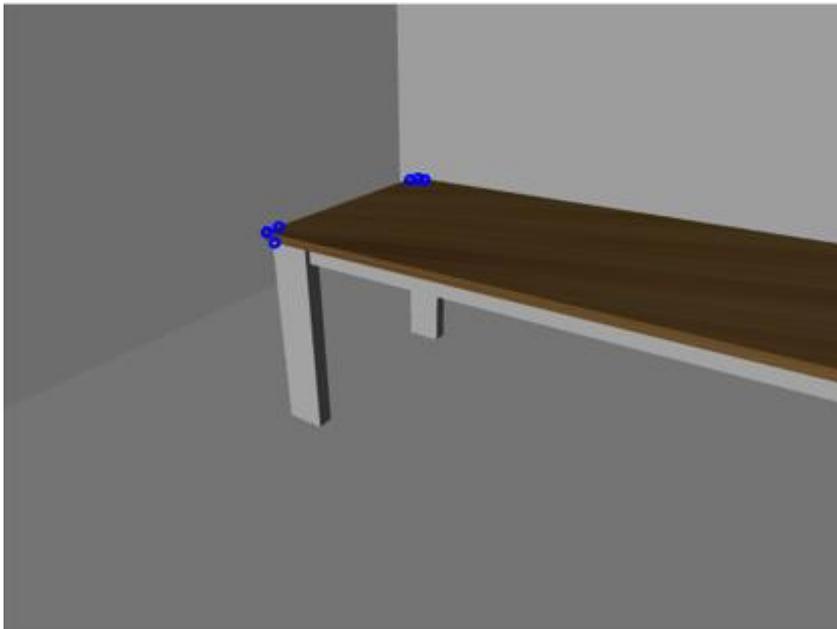


Figura 43 Se aplica la supresión del no máximo y se incorporan las esquinas de Harris.

Este ha sido el mayor problema para el modelado de la mesa larga, pues tardaba mucho tiempo en calcular el modelo inicial hasta que la cámara centraba en un punto en el que Harris detectaba las esquinas necesarias para realizar el modelado.

6 CONCLUSIONES

En este trabajo se ha desarrollado un método de modelado activo que permite a un robot móvil obtener una representación completa de un objeto a través de un plan perceptivo en el que intervienen primitivas visuales y acciones destinadas a obtener, mantener y mejorar el modelo. La implementación se ha llevado a cabo para un tipo de objeto concreto: mesas rectangulares.

Observando los resultados de las pruebas realizadas en los experimentos prácticos puede afirmarse que, al menos trabajando con un modelo sencillo, el plan perceptivo desarrollado proporciona robustez y fiabilidad al modelo extraído. Uno de los mayores problemas, sin embargo, tiene que ver con el tiempo necesario para generar el modelo, aunque, durante el mantenimiento, el modelo mejora y se estabiliza de manera rápida. Esto hace suponer que es posible simplificar la parte del plan dedicada a la generación del modelo inicial, para obtener un primer modelo de manera más eficiente, y destinar más peso a la mejora y mantenimiento.

Como investigación adicional se plantea el desarrollo de escenarios más complejos, en los que se modelen varios objetos, y realizar un estudio posterior que mida la escalabilidad de la propuesta cuando el problema se complica y el sistema se incorpore a otras planificaciones que utilicen los modelos extraídos.

REFERENCIAS BIBLIOGRÁFICAS

- [1] DICK, A, TORR, P and CIPOLLA, R, 2000, Automatic 3D Modelling of Architecture. *Proceedings of the British Machine Vision Conference 2000*. 2000. DOI 10.5244/c.14.39.
- [2] QUISPE, Ana Huaman, MILVILLE, Benoit, GUTIERREZ, Marco A., ERDOGAN, Can, STILMAN, Mike, CHRISTENSEN, Henrik and AMOR, Heni Ben, 2015, Exploiting symmetries and extrusions for grasping household objects. 2015 IEEE International Conference on Robotics and Automation (ICRA). 2015. DOI 10.1109/icra.2015.7139713.
- [3] DORAI, C., WANG, G., JAIN, A. and MERCER, C., 1996. From images to models: automatic 3D object model construction from multiple views. *Proceedings of 13th International Conference on Pattern Recognition*, DOI 10.1109/icpr.1996.546128.
- [4] FISCHLER, Martin A. and BOLLES, Robert C., 1987, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Readings in Computer Vision*. 1987. P. 726–740. DOI 10.1016/b978-0-08-051581-6.50070-2.
- [5] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981.
- [6] HARRIS, C. and STEPHENS, M., 1988, A Combined Corner and Edge Detector. *Proceedings of the Alvey Vision Conference 1988*. 1988. DOI 10.5244/c.2.23.
- [7] SHI, Jianbo and Tomasi, 1994, Good features to track. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94*. 1994. DOI 10.1109/cvpr.1994.323794.
- [8] BROWN, M., SZELISKI, R. and WINDER, S. Multi-Image Matching Using Multi-Scale Oriented Patches. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05)*. DOI 10.1109/cvpr.2005.235.
- [9] RoboComp. *RoboComp* [online]. Available from: <https://robolab.unex.es/index.php/robocomp/>

[10] OpenCV. *OpenCV* [online]. Available from: <http://opencv.org/about.html>.

[11] Kdevelop. *Kdevelop - EcuRed*[online], [Accessed 5 July 2017]. Available from: <https://www.ecured.cu/Kdevelop>.

[12] CANNY, John, 1987, A Computational Approach to Edge Detection. Readings in Computer Vision. 1987. P. 184–203. DOI 10.1016/b978-0-08-051581-6.50024-6.

[13] DUDA, Richard O. and HART, Peter E., 1972, Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*. January 1972. Vol. 15, no. 1p. 11–15. DOI 10.1145/361237.361242.

[14] YURET, D, and de la MAZA, M., 1993, Dynamic hill climbing: Overcoming the limitations of optimization techniques. In *Second Turkish Symposium on Artificial Intelligence and Neural Networks*. 1993.