



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**Detección de palets mediante
LIDAR**



Presentado por Mario Flores Espiga
en Universidad de Burgos — 25 de junio
de 2019

Tutor: Jesus Enrique Sierra García



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. Jesus Enrique Sierra García, profesor del departamento de Ingeniería, área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Mario Flores Espiga, con DNI 71306054M, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Detección de palets mediante LIDAR.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 25 de junio de 2019

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. nombre tutor

D. nombre co-tutor

Resumen

Cada vez es más la tecnología que se implementa en la industria de cara a mejorar sus procesos haciendolos más eficientes.

Dentro del transporte de mercancías, ya sea a nivel interno como a nivel externo de una industria, se han visto importantes mejoras en tecnología en los últimos años, como la popularización de los AGV's (Automated Guided Vehicle).

Los AGV's son vehículos automatizados que se usan para transportar cargas, por lo general paletizadas, a través de las zonas de las fábricas o almacenes. Normalmente, estos vehículos tienen rutas preprogramadas para recoger la mercancía, pero ¿qué pasa si la mercancía que iban a recoger no se encuentra exactamente donde tenía que estar?

Una desviación en el ángulo de un palet, una separación de unos metros respecto de donde debía estar, pueden hacer que el AGV no sea capaz de recoger ese palet, con la consecuencia del paro en esa línea de transporte y la necesidad de una intervención humana para volver a retomar el funcionamiento.

Aquí es donde entra en juego la visión artificial, dotar a los AGV de la capacidad de detectar la mercancía y su posición, para así volverlos más precisos y eficientes y no depender de la correcta colocación de la mercancía.

Existen multitud de opciones para el reconocimiento de los pallets en cuanto a sensores se refiere, pero se ha optado por utilizar un sensor de tipo LIDAR por su capacidad de funcionamiento independientemente de las condiciones de luminosidad, funcionando incluso en total oscuridad.

Otros sensores como cámaras RGB no son capaces de proporcionarnos este rango de operación, teniendo el inconveniente de perder mucha precisión en el momento que las condiciones de luminosidad no son las adecuadas y llegando a ser prácticamente inservibles en total oscuridad.

En este proyecto se investiga como utilizar un sensor láser LIDAR para el reconocimiento de pallets en el entorno simulado de un almacén.

Descriptores

LIDAR, detección de pallets, AGV, láser, nube de puntos ...

Abstract

Technology nowadays is more and more implemented in the industry in order to enhance it's process and make them more efficient. Referring to logistics, whether it is at internal level or at external level to the industry itself, there has been mayor improvements in last years technology, e.g. the popularization of AGV's (Automatic Guided Vehicles).

AGV's are automated vehicles used for cargo transportation, tipically palletized, through the different zones of a warehouse or a factory. Tipically, AGV's follow preprogrammed routes to pick cargo up, but, what if the cargo that was going to be picked is not exactly where it is supposed to be?

A little angle deviation, a small separation from the original position of the cargo, can make the AGV not able to pick the cargo up, slowing down or even stopping that transportation route and needing human intervention to restore normal operation.

This is where artificial vision systems come into play, giving AGV's the capability to to detect cargo and its position, to make them more efficient and precise, so they don't depend on the exact colocation of the cargo.

There are plenty of options to recognise pallets if we talk about sensor types, but in this particular case it has been chosen to use a LIDAR sensor due to its capacity to work independently of the lighting conditions,

beeing able to work even at complete darkness. Other sensor types like RGB cameras are not able to provide this vast range of operation, with the disadvantage of loosing a lot of presition when the lighting conditions are not optimal and even beeing almost useless at total darkness.

In this project it is investigated how to use a laser LIDAR sensor to recognise pallets at a simulated warehouse enviroment.

Keywords

LIDAR, pallet detection, AGV, laser, point cloud ...

Índice general

Índice general	III
Índice de figuras	V
Índice de tablas	VI
Introducción	1
1.1. Introducción	1
1.2. Estructura de la memoria	2
Objetivos del proyecto	5
2.1. Objetivos generales	5
2.2. Objetivos técnicos	5
2.3. Objetivos personales	6
Conceptos teóricos	7
3.1. Preprocesado de los datos	8
3.2. Algoritmo de detección	9
3.3. Material necesario	12
3.4. Tratamiento de los datos del láser	14
Técnicas y herramientas	17
4.1. Herramientas para el desarrollo de la memoria	17
4.2. Herramientas para el desarrollo y pruebas del código	17
4.3. Metodología	18
4.4. Patrones de diseño	19
4.5. Gestión del repositorio	19

4.6. Gestión de las tareas	20
Aspectos relevantes del desarrollo del proyecto	21
5.1. Conocimientos aplicados en el proyecto	21
5.2. Desarrollo del algoritmo de detección	24
5.3. Desarrollo del apartado gráfico	26
Trabajos relacionados	27
6.1. Artículos científicos	30
Conclusiones y Líneas de trabajo futuras	35

Índice de figuras

3.1. Traducción de los datos hexadecimal-decimal	9
3.2. Algoritmo K-Means	10
3.3. Regla del coseno	11
3.4. Medidas de un palet de tipo europeo	12
3.5. Sensor láser Hokuyo utilizado en el proyecto	13
4.6. Instantánea del entorno de desarrollo integrado Spyder	18
4.7. Patrón MVC	19
5.8. Ejemplo de búsqueda en Google Scholar	24
5.9. Punto óptimo de instalación del sensor láser en un AGV	25
5.10. Trama de datos en la que se puede identificar visualmente las patas del palet detectado	26

Índice de tablas

Introducción

1.1. Introducción

Cada vez es más la tecnología que se implementa en la industria de cara a mejorar sus procesos haciendolos más eficientes.

Dentro del transporte de mercancías, ya sea a nivel interno como a nivel externo de una industria, se han visto importantes mejoras en tecnología en los últimos años, como la popularización de los AGV's (Automated Guided Vehicle).

Los AGV's son vehículos automatizados que se usan para transportar cargas, por lo general paletizadas, a través de las zonas de las fábricas o almacenes. Normalmente, estos vehículos tienen rutas preprogramadas para recoger la mercancía, pero ¿qué pasa si la mercancía que iban a recoger no se encuentra exactamente donde tenía que estar?

s

Una desviación en el ángulo de un palet, una separación de unos metros respecto de donde debía estar, pueden hacer que el AGV no sea capaz de recoger ese palet, con la consecuencia del paro en esa línea de transporte y la necesidad de una intervención humana para volver a retomar el funcionamiento.

Aquí es donde entra en juego la visión artificial, dotar a los AGV de la capacidad de detectar la mercancía y su posición, para así volverlos más precisos y eficientes y no depender de la correcta colocación de la mercancía.

Existen multitud de opciones para el reconocimiento de los pallets en cuanto a sensores se refiere, pero se ha optado por utilizar un sensor de tipo LIDAR por su capacidad de funcionamiento independientemente de las condiciones

de luminosidad, funcionando incluso en total oscuridad.

Otros sensores como cámaras RGB no son capaces de proporcionarnos este rango de operación, teniendo el inconveniente de perder mucha precisión en el momento que las condiciones de luminosidad no son las adecuadas y llegando a ser prácticamente inservibles en total oscuridad.

En este proyecto se investiga como utilizar un sensor láser LIDAR para el reconocimiento de palets en el entorno simulado de un almacén.

1.2. Estructura de la memoria

La memoria se divide según la siguiente estructura:

- **Introducción:** Descripción resumida del problema a resolver y solución propuesta al mismo. Estructura de la memoria.
- **Objetivos del proyecto:** Objetivos que pretende alcanzar el proyecto
- **Conceptos teóricos:** Explicación de los conceptos teóricos no triviales tratados en el proyecto para la comprensión de la solución.
- **Técnicas y herramientas:** Listado de técnicas metodológicas y herramientas utilizadas para gestión y desarrollo del proyecto.
- **Aspectos relevantes del desarrollo:** Explicación de los aspectos destacables que se han dado durante la realización del proyecto.
- **Trabajos relacionados:** Estado del arte en la investigación de los AGV's autónomos y reconocimiento de materiales paletizados.
- **Conclusiones y líneas de trabajo futuras:** Conclusiones a las que se ha llegado tras la realización del proyecto y potenciales mejoras y líneas de desarrollo futuras.

Junto a la memoria se proporcionan los siguientes anexos:

- **Plan del proyecto software:** Planificación temporal y estudio de viabilidad del proyecto.
- **Especificación de requisitos del software:** Se describe la fase de análisis; los objetivos generales, el catálogo de requisitos del sistema y la especificación de requisitos funcionales y no funcionales.
- **Especificación de diseño:** Se detalla la fase de diseño; el ámbito del software, el diseño de datos, el diseño procedimental y el diseño arquitectónico.
- **Manual del programador:** Recoge los aspectos más relevantes relacionados con el código fuente (estructura, compilación, instalación, ejecución, pruebas, etc.).
- **Manual de usuario:** Guía de usuario para el correcto manejo de la aplicación.

Objetivos del proyecto

Este apartado explica cuales van a ser los objetivos que persigue el proyecto. Tratando de diferenciar los objetivos generales de partida y los objetivos técnicos que aparecen a lo largo del desarrollo.

2.1. Objetivos generales

Los objetivos generales del proyecto se resumen en:

- Establecer la conexión con el dispositivo láser para ser capaz de recibir tramas de información desde un ordenador.
- Interpretar los datos de lectura recibidos y efectuar el adecuado tratamiento de los mismos.
- Concluir si en los datos de lectura existe o no un palet y su posición en el caso de existir.

2.2. Objetivos técnicos

Los objetivos técnicos del proyecto son los siguientes:

- Desarrollar un algoritmo capaz de tratar los datos recibidos en forma de puntos y procesarlos en tiempo real.
- Desarrollar una aplicación en python para la recepción y tratamiento de los datos.

- Establecer un umbral para poder decidir si en los datos recibidos se visualiza un palet y conocer la situación del laser respecto del palet en términos de distancia.
- Utilizar la arquitectura Modelo-Vista-Controlador MVC (Model-View-Presenter MVP)
- Usar durante el desarrollo del proyecto una herramienta de control de versiones, en este caso GitHub
- Hacer uso de la metodología ágil Scrum para el desarrollo del proyecto.

2.3. Objetivos personales

- Aprender sobre las distintas aproximaciones al problema de la detección de palets.
- Hacer uso de los conocimientos aprendidos a lo largo de la carrera.
- Ser capaz de utilizar herramientas y conocimientos ajenos a la carrera mediante el autoaprendizaje.

Conceptos teóricos

La parte del algoritmo más compleja reside en el propio algoritmo de detección de palets y en el tratamiento de los datos recibidos desde el láser. El flujo del programa se compone de los siguientes pasos:

- Preprocesado de los datos:
 1. Primero se establece una conexión con el LIDAR, mediante TCP.
 2. Posteriormente se tratan los datos para su interpretación en el algoritmo detector.
- Algoritmo detector de palet:
 1. Se selecciona un área de interés donde se conoce previamente que puede estar el palet.
 2. Sobre la nube de puntos ya acotada se realiza un clustering con el algoritmo KMEANS.
 3. Se calculan las distancias entre los diferentes clusters y al emisor, para contrastarlas con las medidas reales del tipo de palet que se esté utilizando y validar así la ocurrencia del palet.

3.1. Preprocesado de los datos

Las tramas que se reciben del láser tienen en su interior información que no nos interesa para el fin del proyecto, como es el estado del mismo, datos de control etc. con lo cual se criban los datos descartando información innecesaria.

Posteriormente, ya que el LIDAR utilizado trabaja en datos hexadecimales, para poder procesarlos, se debe realizar una traducción de los mismos a formato decimal.

La traducción se realiza de la siguiente manera:

Traducción

Una vez extraídos los datos de cada apartado de la cadena devuelta por el láser, estos deben ser traducidos en base a los siguientes pasos:

1. Se extrae uno por uno el dato en hexadecimal de los caracteres.
2. Si este dato está comprendido entre 30_h y 39_h , se le resta 30_h . Mientras que si está entre 41_h y 46_h , se le resta 37_h .
3. Se convierte cada dato hexadecimal a binario.
4. Se agrupan todos los datos en binario y se traduce la agrupación a decimal para expresar el dato en milímetros.

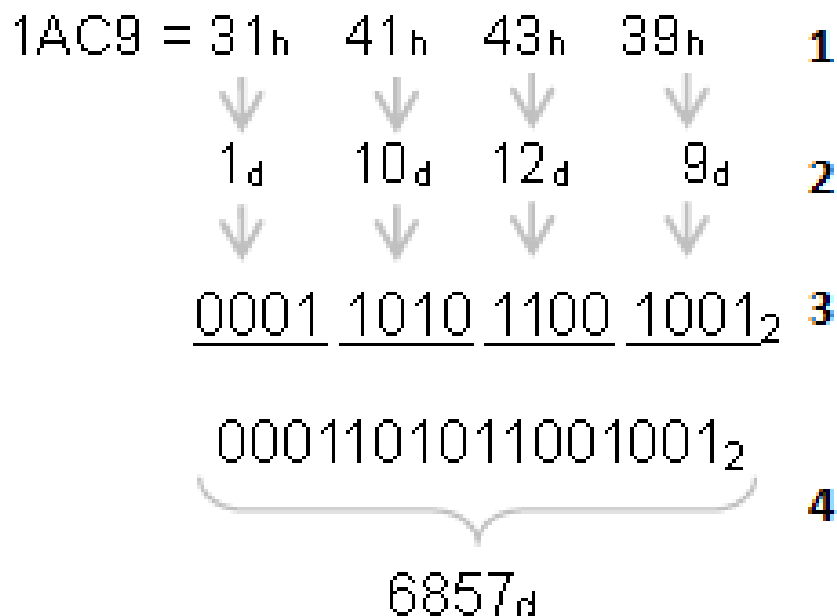


Figura 3.1: Traducción de los datos hexadecimal-decimal

Después, por cada dato traducido, se asigna a cada uno un ángulo correspondiente para expresar así la lectura en coordenadas polares. Posteriormente, se traducen los datos de coordenadas polares a coordenadas cartesianas mediante la siguiente formula:

$$x = r * \cos(\theta)$$

$$y = r * \sin(\theta)$$

Ahora ya se dispone de una lista de distancia y angulo, adecuada para poder ser representada en una gráfica y tratada en el algoritmo detector.

3.2. Algoritmo de detección

Clustering

El *clustering* o *algoritmo de agrupamiento* es un proceso por el cual se agrupan una serie de vectores en función de algun criterio Los criterios generalmente corresponden a cercanía o similitud entre los datos de los

vectores.

En este caso particular se utiliza la función de distancia euclídea para el agrupamiento de los puntos de la nube, formando así grupos de puntos en función de su cercanía en el espacio.

K-Means

K-Means o *K-Medias* es un algoritmo de clustering que particiona n datos en k grupos.

Cada punto es agrupado en el conjunto cuyo valor medio es más próximo al del punto.

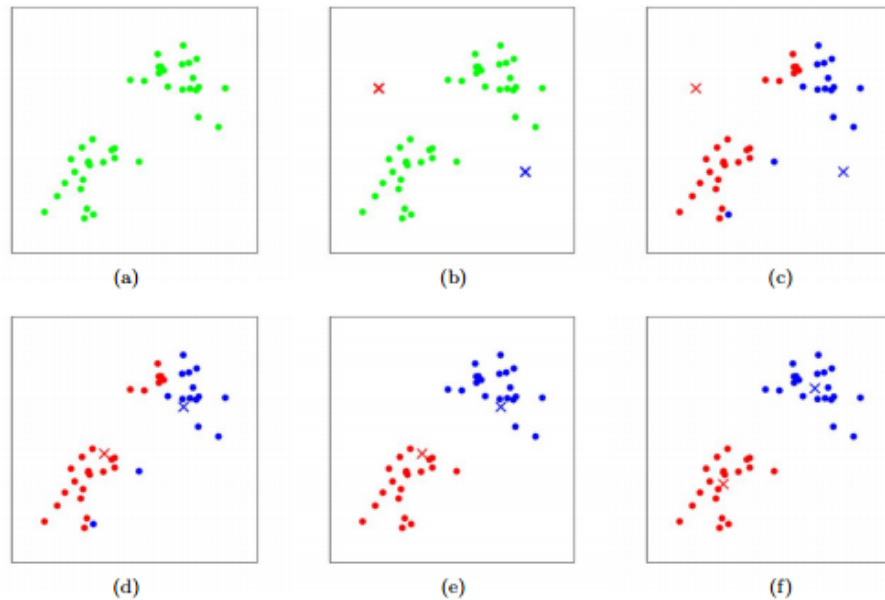


Figura 3.2: Algoritmo K-Means

Para comenzar, el algoritmo asigna k centros aleatorios y realiza las agrupaciones en función de la distancia de los puntos a esos centros. Después se actualizan los nuevos centros como el centro de cada cluster que se ha calculado.

El algoritmo es de carácter iterativo, ejecutándose varias veces hasta que la solución converge (Las soluciones de cada iteración no cambian)

oo

Se ejecuta el algoritmo K-means con $k=3$, puesto que la característica por la que se está intentando identificar el palet es por sus tres patas frontales, las cuales el láser va a observar de manera frontal, al ir este montado en la parte de las palas del AGV.

Una vez agrupados los puntos que corresponden a cada pata, se comprueba que el número de cada cluster tiene un tamaño similar al resto (por motivos de ruido en las lecturas del láser, se aplica un margen de error especificado en el código). Concretamente se comparan el número de puntos que componen a cada cluster.

Posteriormente, se calcula la distancia media a todos los puntos de cada cluster desde el emisor (en este caso el láser) así como el ángulo medio de los puntos de cada cluster.

Con esto conseguimos representar cada cluster por un punto (ángulo y distancia).

Ahora, mediante el teorema del coseno, y con los datos de distancia y ángulo de cada punto representando de los cluster, se calcula la distancia que separa a estos tres puntos.

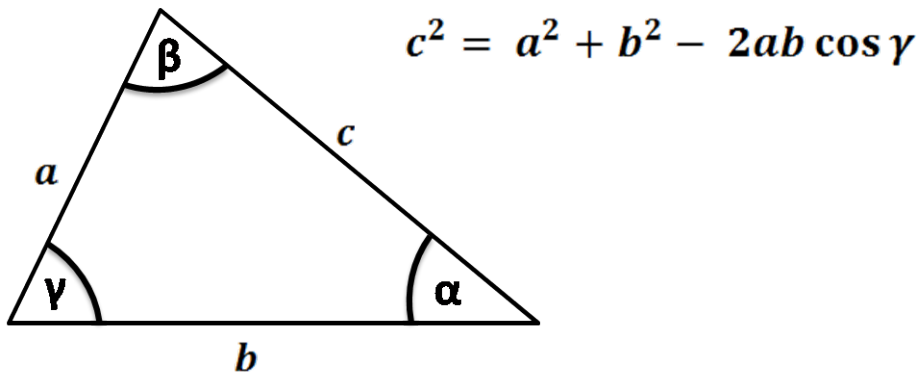


Figura 3.3: Regla del coseno

Estos datos se contrastan con las medidas reales del tipo de palet que se esté utilizando. También se contrastan entre ellos para comprobar que ambas distancias son iguales, de nuevo con una tolerancia por los posibles errores de lectura.



Figura 3.4: Medidas de un palet de tipo europeo

Una vez realizadas todas estas comprobaciones, se puede concluir la existencia de un palet.

3.3. Material necesario

Se ha usado el siguiente hardware para la realización del proyecto:

- Sensor Láser: Encargado de sondear el entorno y devolver los datos obtenidos. El láser que se va a utilizar es de tipo LIDAR (Laser Imaging Detector And Ranging). Este tipo de láser emite ondas infrarrojas, para después recoger las ondas de vuelta, y en función del tiempo entre la emisión y la recepción calcula la distancia. En este proyecto, se ha utilizado el láser Hokuyo Safety Laser Scanner (UAM-05LP-T301), capaz de distinguir tres zonas independientes de detección en función de la distancia a la que se encuentren los objetos en el entorno.



Figura 3.5: Sensor láser Hokuyo utilizado en el proyecto

- Cable ethernet: A través de este cable se envían y reciben datos del láser al ordenador. Se usará un UAM-NET, un cable Ethernet de 3 metros de longitud desarrollado por Hokuyo, misma empresa desarrolladora del láser empleado, lo que hace que resulte idóneo para evitar problemas de incompatibilidad y asegurar así el correcto funcionamiento del sistema.
- Ordenador: Es la parte central del proyecto. Con él, se ejecuta el sistema software encargado de procesar los datos que recibe de la

unidad láser, y mediante una serie de algoritmos, concluir la detección o no de un palet en el área visionada.

3.4. Tratamiento de los datos del láser

El software desarrollado en este proyecto es el encargado de recibir los datos del láser, para su posterior procesado. Los datos se reciben en forma de tramas y se les debe aplicar un proceso de tratamiento para separar datos no relevantes de las tramas. Posteriormente se deben traducir estos datos de coordenadas polares a coordenadas cartesianas, y finalmente, ejecutar los algoritmos que determinan la posible detección de un palet.

Uso del láser

Para poder establecer la comunicación con el láser y ordenarle la captura de datos, se utilizan comandos. Aunque existen multitud de comandos disponibles, para nuestro objetivo nos bastara con utilizar los comandos de tipo AR, los cuales ordenan al laser devolver los datos de lectura. Existen 6 comandos de este tipo.

1. AR00: Medición única en la que devuelve las distancias.
2. AR01: Medición única en la que devuelve las distancias e intensidades.
3. AR02: Medición continua en la que devuelve las distancias.
4. AR03: Detiene la continuidad del comando AR02.
5. AR04: Medición continua en la que devuelve las distancias en intensidades.
6. AR05: Detiene la continuidad del comando AR04.

Funcionamiento a nivel interno del láser

Este láser en concreto, escanea un ángulo de 270° mediante un cabezal rotativo que gira 2000rpm emisor de ondas infrarrojas. El láser lanza 1081 haces a lo largo de los 270° de visión de los que posee, mientras la unidad receptora recoge la reflexión de cada uno de esos haces en los objetos del entorno. Así, calcula el tiempo que ha tardado cada emisión en retornar y, sabiendo la velocidad a la que se propagan los haces, computa la distancia a la que se encuentra cada punto.

En este apartado cabe destacar los problemas que surgieron derivados del amplio rango de captura del láser, puesto que su funcionalidad principal es la de escaner de seguridad y no de medición. Al tener un rango tan grande, en las capturas efectuadas existía mucho ruido del entorno que dificultaba la correcta identificación del palet, con lo cual se optó por limitar el rango con el que se trabajaba a nivel de código para así focalizar la atención del algoritmo en la parte frontal del láser.

Tránsito de mensajes

Una vez determinado el comando que se quiere utilizar, se debe crear una estructura de mensaje para enviárselo al láser con el siguiente formato:

- STX: Tipo caracter, en bytes, que marca el comienzo del mensaje. Normalmente es un '2'.
- Command size: Tamaño del mensaje que se va a enviar. Formato hexadecimal. En este proyecto, los comandos van a ocupar siempre el mismo espacio, 14 caracteres, con lo que esta parte del mensaje siempre será '000E'.
- Header: Tipo de comando que se le manda al laser, como se ha mencionado antes, en este proyecto solo se usa el tipo AR.
- Subheader: Especificación del comando que se va a utilizar dentro de la familia de comandos escogida en la cabecera.
- CRC: Comprobacion de redundancia Cíclica. Código que se añade para asegurar que el mensaje no se ha corrompido en el envío del mismo.
- EXT: Tipo caracter, en bytes, que marca el final del mensaje. Normalmente es un '3'.

En los mensajes devueltos por el láser al pedirle información de una lectura mediante un comando, se encuentra información como el tiempo empleado para la lectura, estado de los puertos etc que no tiene utilidad en este proyecto. Es por esto que debe ser llevado a cabo un proceso de traducción para extraer la información que nos interesa de las tramas que nos devuelve el láser de las lecturas.

Técnicas y herramientas

4.1. Herramientas para el desarrollo de la memoria

Para el desarrollo de la memoria del proyecto se ha usado LaTeX como procesador de textos, principalmente por su libertad y simplicidad a la hora de manipular los distintos parametros del documento. Además, desde LaTeX se puede exportar directamente el archivo a PDF de una manera muy sencilla. Como desventaja de este programa, cabe destacar el aprendizaje necesario al no haberlo utilizado antes a lo largo del grado en contraposición a la facilidad de haber utilizado un procesador de textos en el que tuviera más experiencia como por ejemplo Microsoft Word u Open Office.

4.2. Herramientas para el desarrollo y pruebas del código

Realterm

Este programa se ha utilizado para la conexión con el dispositivo láser. Su funcionamiento se resume en un programa para transmitir mensajes TCP a través del puerto serie. Así, a través de esta conexión, establecemos una via de comunicación con el láser de tipo Cliente-Servidor para realizar labores de pruebas.

IDE - Spyder

Para el desarrollo de código en el lenguaje Python se ha escogido el entorno de desarrollo integrado (IDE) Spyder.

Se ha escogido este programa por ser un entorno de desarrollo potente e interactivo.

Entre sus funcionalidades están la depuración, Ipython (Intérprete interactivo de python) y algunas bibliotecas integradas como numPy o matplotlib que han sido de utilidad en el desarrollo de este proyecto.

Para terminar, es un programa muy visual pero sin ser perder por ello funcionalidad y al ser similar a IDEs de otros lenguajes como Eclipse en Java el tiempo de aprendizaje es reducido.

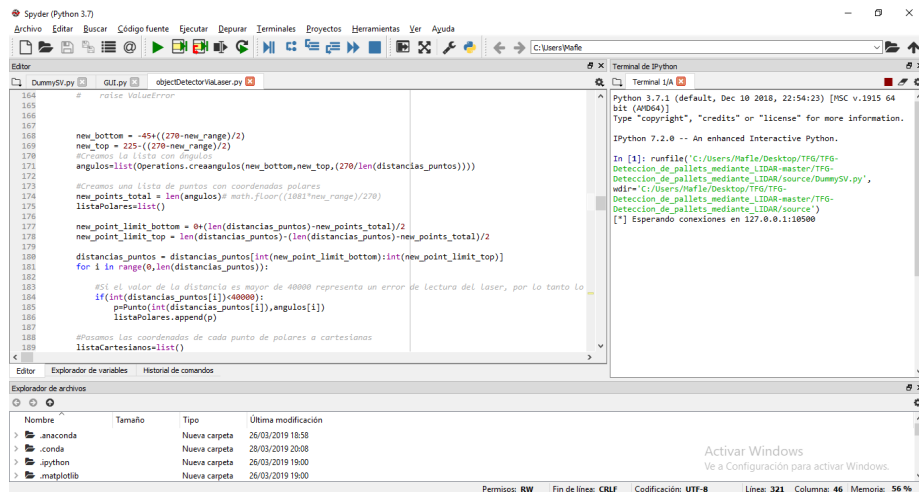


Figura 4.6: Instantánea del entorno de desarrollo integrado Spyder

4.3. Metodología

Scrum

Scrum es una metodología denominada *ágil*. Se realiza un desarrollo de manera incremental, dividiendo el proyecto en subproblemas más pequeños. Cada subtarea pertenece a un *sprint* tras el cual se realiza una revisión de las tareas realizadas. Se ha elegido esta metodología para llevar a cabo el proyecto dado a su facilidad para ser flexible ante cambios y nuevos requisitos del desarrollo.

4.4. Patrones de diseño

Model View Controller (MVC)

MVC consta de tres componentes a nivel de software:

- **Modelo:** Representación de los datos. En este caso los datos proceden directamente de la conexión con el láser.
- **Vista:** Parte que 'muestra' la información al usuario y se comunica con él. Comunmente una interfaz de usuario. Se corresponde con la clase GUI.
- **Controlador:** Es la parte encargada de realizar las peticiones al modelo y gestionar los eventos que el usuario genera. Está integrado en la clase `objectDetectorViaLaser`.

Es un patrón de diseño adecuado puesto que separa en capas claramente distinguibles y funcionalmente similares el software desarrollado, facilitando el mantenimiento y desarrollo de código.

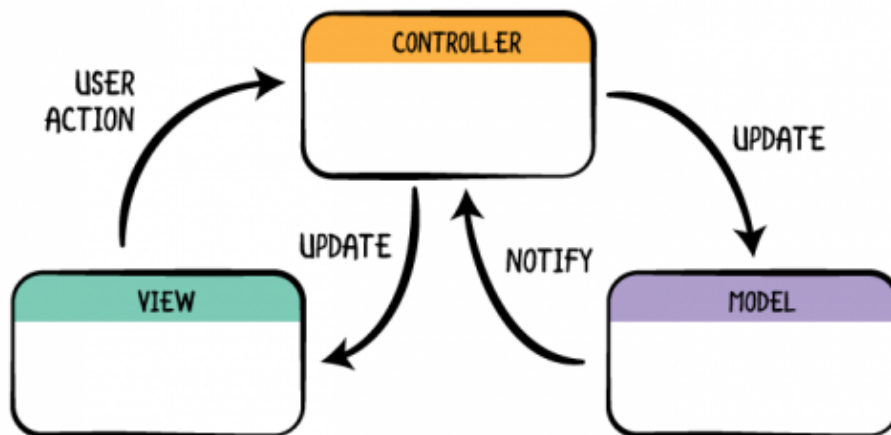


Figura 4.7: Patrón MVC

4.5. Gestión del repositorio

Para la gestión del proyecto y la creación y actualización de un repositorio online, se han barajado dos herramientas: *Github* y *BitBucket*. Finalmente se

ha optado por usar Github debido a la mayor habituación de uso a lo largo de la carrera en diversas asignaturas, partiendo con unos conocimientos que han permitido usar esta herramienta sin necesidad de aprendizaje.

GitHub

Github es una plataforma online para el control de versiones, basada en el sistema Git. Integra funcionalidades como documentación, revision de código, gestion de bugs, de tareas etc. La herramienta es gratuita para proyectos de código abierto, sin embargo, para crear un repositorio privado es necesaria una suscripción.

4.6. Gestión de las tareas

Existen multitud de herramientas y programas que se pueden utilizar para implementar scrum, como por ejemplo Jira, Trello, QuickScrum etc. Por razones de comodidad se ha optado por Trello, dado que es un software que ya se ha usado en la carrera, es intuitivo y su uso es fácil de aprender.

Aspectos relevantes del desarrollo del proyecto

5.1. Conocimientos aplicados en el proyecto

Son muchos los conocimientos que han sido necesarios de una u otra forma para llevar a cabo este proyecto. Muchos de esos conocimientos se han adquirido a lo largo de las asignaturas de este grado, mientras que otros han sido obtenidos mediante el autoaprendizaje haciendo uso de distintos tipos de recursos *Manuales de programación, tutoriales, documentación online...*)

Conocimientos aprendidos dentro del grado

Programación en Python

Este lenguaje ha sido clave para desarrollar todo el código del proyecto, un lenguaje que se ha ido aprendiendo a lo largo del grado en diversas asignaturas como Sistemas Inteligentes, Algoritmia, Nuevas Tecnologías y Empresa y Gestión de la Información.

Las nociones y conocimientos sobre este lenguaje que se han recibido han servido para poder realizar la parte de programación de este proyecto, así como para facilitar el aprendizaje de nuevas librerías y características necesarias para la realización del código.

Redes

Para manejar la conexión necesaria con el láser, se han empleado conocimientos sobre redes, concretamente sobre conexiones TCP y sockets. Esto ha sido posible gracias a lo aprendido en asignaturas como Redes o Sistemas Distribuidos.

Gestión de proyectos

La gestión de proyectos es la parte encargada en el proyecto de planear las tareas a realizar para llegar al objetivo final, adecuando los tiempos, costes tanto humanos como materiales y todo lo necesario.

Como ya se ha expuesto previamente, se ha utilizado una metodología ágil, SCRUM, la cual además de otros conocimientos necesarios para la gestión ha sido aprendida en la asignatura Gestión de Proyectos. También se ha aprendido en esta asignatura el concepto de los tableros KanBan , un método similar a la metodología SCRUM.

Se han seguido desarrollos incrementales mediante sprints de una duración comprendida entre un mes y tres meses en las partes más complejas del proyecto, retrasándose cuando se arrastraban tareas de sprints anteriores por fallos en el código, entre otros contratiempos. Para seguir el desarrollo de los sprints se utilizó la aplicación **Trello**.

Hardware

El aspecto a nivel de hardware del proyecto comprende la alimentación eléctrica del propio láser y la conexión de datos con el mismo.

Para la alimentación se buscó una fuente de alimentación de corriente continua, 24v y por lo menos 1A como se especifica en los manuales del equipo láser. En cuanto a la conexión de datos con el láser, fue necesario comprender que función desempeñaba cada cable (*Polo positivo, polo negativo, tierra, reset...*)

Fue en la asignatura Mantenimiento de Equipos Informáticos donde se adquirieron los conocimientos necesarios para operar con el equipo láser y su fuente de alimentación.

Conocimientos aprendidos fuera del grado

Manejo del equipo láser

Durante el desarrollo del proyecto ha sido necesario comprender cómo funciona el láser a nivel interno, qué distintos comandos usa y cómo funciona el protocolo de comunicación para poder operar con él y aprovechar sus funcionalidades

Investigación del estado del arte

Con el fin de conocer el nivel de desarrollo de este problema a nivel global, se ha aprendido a la búsqueda de artículos científicos en diversas plataformas como son [ResearchGate](#), [Google Scholar](#) y [IEE Explore](#) entre otros.

Se ha adquirido la capacidad de filtrado a la hora de decidir qué artículos representaban información relevante para el proyecto y cuales no, así como la capacidad de sintetizar el contenido de los artículos para conocer otras alternativas y aproximaciones al problema de reconocimiento de palets.

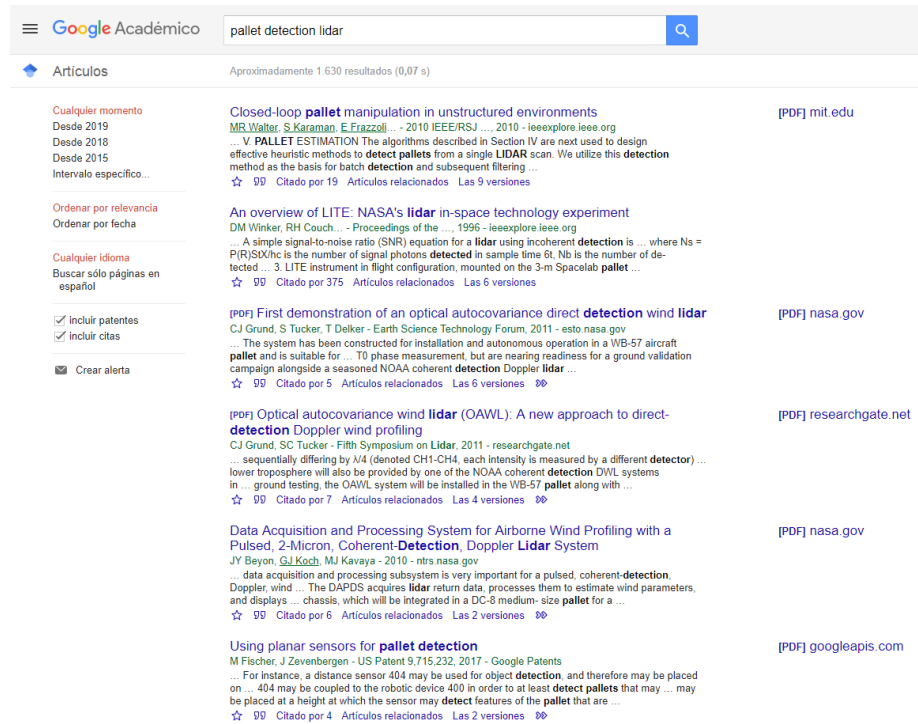


Figura 5.8: Ejemplo de búsqueda en Google Scholar

5.2. Desarrollo del algoritmo de detección

Buena parte del esfuerzo del proyecto se dedicó al desarrollo y programación del algoritmo para la detección de los palets.

Se empezó por documentarse sobre otras soluciones al problema de detección y una clasificación de las mismas. Se encontraron multitud de artículos sobre el problema con diversas aproximaciones.

Posteriormente, basándose en el equipo láser disponible y su forma de capturar los datos en una nube de puntos 3D, se estudió la mejor colocación posible de cara a su potencial incorporación a un AGV. Se llegó a la conclusión de que la mejor perspectiva para observar el palet era a ras de suelo, montado sobre las propias palas del AGV de manera que el palet quede en el mismo plano que el láser.

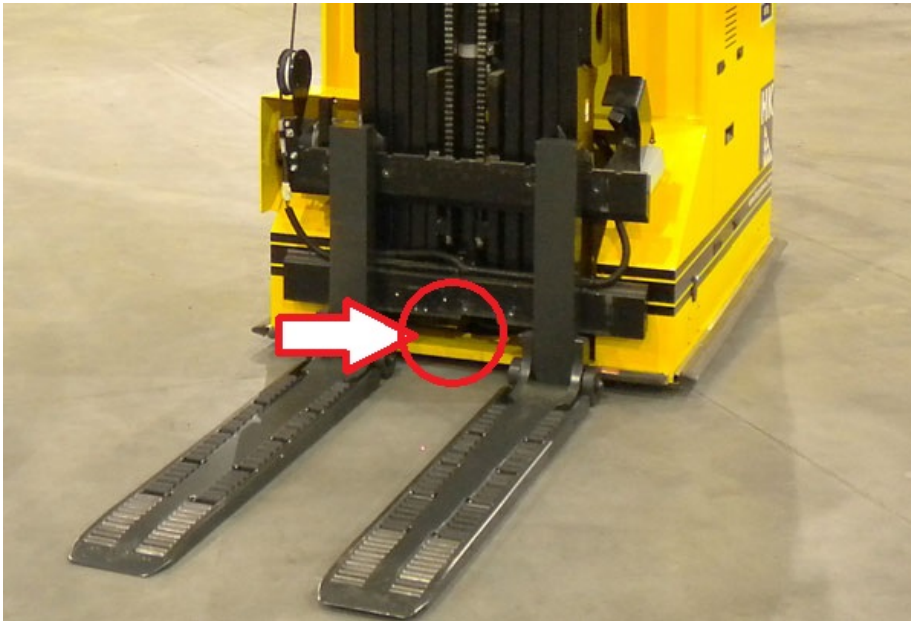


Figura 5.9: Punto óptimo de instalación del sensor láser en un AGV

Se optó por desarrollar el algoritmo en el lenguaje Python en contraposición a Java en base a su mejor adecuación para aplicaciones en tiempo real, por su amplio uso y soporte y por ser un lenguaje ampliamente usado durante el grado y así poder aplicar los conocimientos aprendidos.

El algoritmo de desarrollo mediante prueba y error, añadiendo pequeños incrementos de funcionalidad a lo largo del desarrollo, como podían ser añadir nuevas comprobaciones a la hora de la detección o hacer las que ya estaban implementadas más robustas.

Han sido varias las alternativas al algoritmo que se han estudiado, pero finalmente, optando por la simplicidad y dejando abierto el desarrollo de mejoras en la detección, se optó por un camino en el que la detección se basa en el reconocimiento del frontal del palet mediante las patas del mismo, y la distancia que las separa.

En la siguiente imagen se pueden observar desde un plano cenital las tres patas del palet, situadas aproximadamente a un metro del láser.

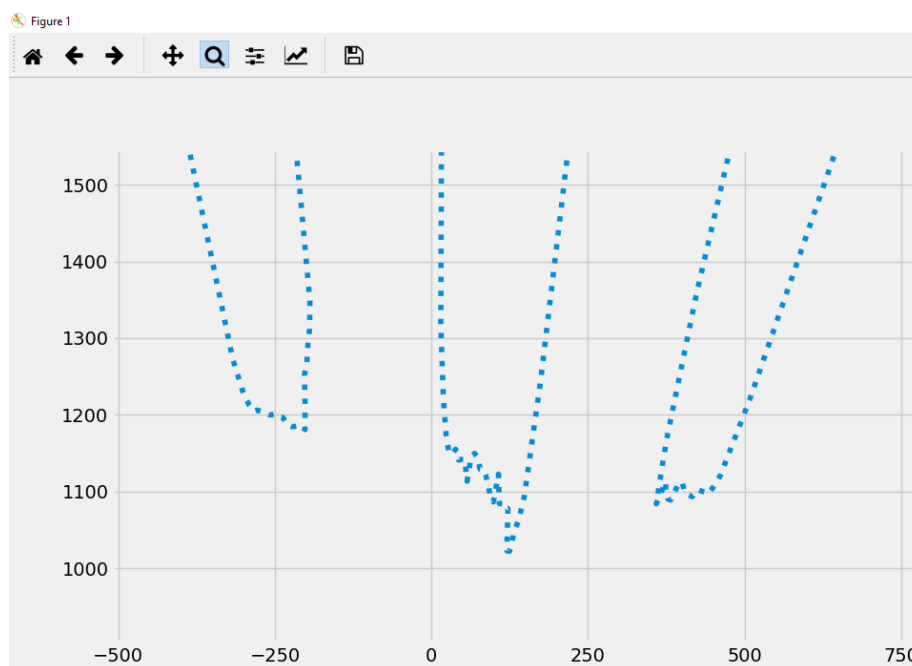


Figura 5.10: Trama de datos en la que se puede identificar visualmente las patas del palet detectado

5.3. Desarrollo del apartado gráfico

Inicialmente, se comenzó con la implementación de una interfaz gráfica en **TKinter**, pero la necesidad de aprender e investigar sobre esta librería de Python y varios problemas de funcionamiento causaron el cambio a otras librerías más sencillas.

Posteriormente, con el fin de poder observar de una manera sencilla pero a la vez suficientemente detallada los datos en tiempo real que transmite el láser, se ha utilizado en lugar de TKinter, las animaciones de la biblioteca **matplotlib**. Tras varios problemas de implementación que causaban que la gráfica no se refrescara, se consiguió solucionar el funcionamiento.

De esta manera el usuario puede observar en tiempo real las tramas que el láser está procesando, junto con la confirmación de palet en la trama actual.

Trabajos relacionados

En este apartado del proyecto se ha realizado una labor de investigación en el campo de los AGV's y la detección de objetos, profundizando en la detección de palets. Existen múltiples aproximaciones a este problema, en las que se usa una variedad de técnicas y sensores para intentar resolver la problemática.

Con el fin de resumir y facilitar la comparación entre los diferentes estudios consultados, se ha elaborado una tabla comparativa entre todos los estudios en función de su tecnología utilizada.

Tabla comparativa

Estudios de la tabla:

1. Automatic visual guidance of a forklift engaging a pallet
2. Pallet Pose Estimation with LIDAR and Vision for Autonomous Forklifts
3. Feature-to-Feature based Laser Scan Matching in Polar
4. An efficient Depalletizing System based on 2D Range Imagery
5. Concept of automated load detection for de-palletizing using depth images and RFID data
6. Docking to pallets with feedback from a sheet-of-light range camera
7. Real-time Pallet Localization with 3D Camera Technology for Forklifts in Logistic Environments
8. Autonomous pallet localization and picking for industrial forklifts based on the line structured light
9. Automatic segmentation of pallet images using the 2-D wavelet transform and YUV color space
10. Fast Edge-Based Detection and Localization of Transport Boxes and Pallets in RGB-D Images for Mobile Robot Bin Picking
11. A robust autonomous mobile forklift pallet recognition

12. Robust pallet detection for automated logistics operations
13. Visualization of Pallets
14. Focus based Feature Extraction for Pallets Recognition

	1	2	3	4	5	6	7	8	9	10	11	12	13
LIDAR		x	x	x								x	x
Cámara estándar	x					x			x		x		
Camara 3D (Time of flight)					x		x						
SLAM													
Camara "Sheet of light"						x							
Structured light 3d sensor		x						x				x	
Camara RGB-D										x			

6.1. Artículos científicos

Detection, localisation and tracking of pallets using machine learning techniques and 2D range data

En 2018, los autores Ihab S. Mohamed, Alessio Capitanelli, Fulvio Mastrogiovanni, Stefano Rovetta y Renato Zaccaria publicaron este estudio en el cual presentan una arquitectura capaz de dotar a un AGV, mediante un sensor láser de tipo LIDAR similar al utilizado en el proyecto, de mantener un conteo de palets detectados, así como su posición.

La arquitectura se divide en dos componentes:

- En primer lugar, un detector de palets que utiliza una red neuronal convolucional basado en zonas (Region-based convolutional neural network), en cascada con un clasificador.
- En segundo lugar, se utiliza un filtro Kalman para rastrear los palets ya detectados.

Robust Pallet Detection for Automated Logistics Operations

En este documento publicado en 2016 por Robert Varga y Sergiu Ne-devschi, se estudian las alternativas ya existentes para la visión artificial de los AVG's y se proponen soluciones para mejorarlas.

Se utilizan para la detección imágenes captadas por una cámara estereoscópica y imágenes en escala de grises. Estas últimas proporcionan la información de la localización del palet en 2D mientras que las primeras se usan para conocer la posición 3D y la orientación del palet respecto de la cámara. Emplean un algoritmo compuesto de detección de bordes y líneas de los palets, para después generar posibles candidatos a ser considerados un palet, y probarlos contra las características ya conocidas. Por último, se usa un clasificador para confirmar si se trata o no de una detección positiva.

Focus based Feature Extraction for Pallets Recognition

En este estudio realizado por Rita Cucchiara, Massimo Piccardi y Andrea Prati se utiliza una cámara estandar para extraer características de un palet sobre la imagen que se esta capturando, concretamente se buscan las patas laterales y la pata central del palet mediante el procesamiento de líneas y bordes, después de haber definido una región de interés (ROI). Posteriormente mediante un árbol de decisión se confirma si es o no un palet lo que se esta viendo.

Este estudio se centra en localizar las mismas características que lo planteado en el proyecto, no obstante, utiliza una manera más refinada de confirmar la existencia del palet.

Visualization of Pallets

Roger Bostelman, Tsai Hong y Tommy Chang publicaron este estudio en 2006 en el que utilizan dos sensores de tipo LIDAR, uno en la base del AGV y otro en las 'palas'. Este estudio se especializa en la detección de palets dentro de un camión, de modo que usan las paredes laterales y la pared del fondo del camión para conocer la posición del palet.

El algoritmo utilizado realiza los siguientes pasos:

1. Se convierten los datos obtenidos en plano cartesiano X Y
2. El sistema estima la localización de las cuatro esquinas del camión. (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , (x_4, y_4) .
3. Se aplica la transformada de Hough a los puntos.
4. Encuentra las líneas que delimitan el camión a ambos lados y el fondo.

5. Si alguna de las paredes del camión concuerda con las predicciones, la distancia desde el centro del sensor a esta pared se sobrescribe.
6. Se calcula la distancia hacia cada una de las tres paredes de los puntos que están dentro de los límites predichos.
7. Se agrupan los puntos cuya distancia a la pared izquierda es menor a la pared contraria como el palet izquierdo y viceversa.
8. Se calcula la distancia mínima del sensor a cada uno de los palets encontrados.

Pallet recognition and localization using an RGB-D camera

Autores: Junhao Xiao, Huimin Lu, Lilian Zhang y Jianhua Zhang. En este estudio se utiliza una cámara RGB, concretamente la Kinect 2.0 de Microsoft. En primer lugar el algoritmo realiza un segmentado de las imágenes y filtrado, para después compararlas con muestras de palets.

Automatic visual guidance of a forklift engaging a pallet

En este estudio, Michael Seelinger^a, y John-David Yoder^b usan dos cámaras estándar para capturar imágenes en escala de grises. No obstante, también hacen uso de marcas visuales tanto en el propio AGV como en los palets, y aún siendo un sistema igualmente válido, posee la desventaja de tener que equipar todos los palets del almacén o fábrica con dichas marcas. Puesto que en el ámbito del transporte de mercancías los palets 'viajan' mucho entre la industria, es una tarea demasiado laboriosa mantener todos los palets que pasan por un almacén o fábrica con las marcas necesarias.

Pallet Pose Estimation with LIDAR and Vision for Autonomous Forklifts

Los autores de este estudio (N. Bellomo, E. Marcuzzi, L. Baglivo, M. Pertile, E. Bertolazzi y M. De Cecco) proponen un método de reconocimiento de palets que extrae los datos de un sensor de tipo LIDAR, y utiliza un algoritmo genético de optimización para encontrar la mejor coincidencia entre la imagen que se está recibiendo, y un modelo existente de un palet conocido.

Feature-to-Feature based Laser Scan Matching in Polar Coordinates with Application to Pallet Recognition

Este artículo publicado en 2011 por Zhendong Hea, Yaonan Wanga y Hong Yuc hace uso de un sensor de tipo LIDAR para recoger la información del entorno. Después, su algoritmo emplea segmentación y clustering sobre los puntos recibidos por el LIDAR, detección de las esquinas del palet y finalmente lo compara con un modelo de palet conocido para concluir si se trata de un palet.

Docking to Pallets with Feedback from a Sheet-of-Light Range Camera

Los autores J. Nygirds, T. Hogstrom y A. Wernersson presentan en este estudio un algoritmo que para el reconocimiento de palets, para la correcta inserción de las palas del AGV una vez es conocida ya la existencia del palet.

Hace uso de dos tipos de cámaras: una cámara estándar, y otra de tipo 'sheet of light' (Se trata de una cámara que recoge la reflexión de un haz de luz proyectado para así conseguir una imagen en tres dimensiones).

Se compone de los siguientes pasos:

1. Se define una región de interés (ROI) en base a la predicción del palet en la imagen.
2. Haciendo uso de la transformada de Radon se detecta la orientación del palet y la distancia al mismo.
3. Buscando los intervalos en los que los valores de intensidad del láser son mayores, se determina cual es la pata central del palet.
4. Se transforman las distancias de los puntos obtenidas a coordenadas para el AGV.

Vision-based autonomous load handling for automated guided vehicles

En este estudio de 2014, Robert Varga y Sergiu Nedevschi presentan un método para detectar la existencia de un pallet y estimar su posición y orientación en el espacio.

Se utilizan dos cámaras posicionadas en el AGV, una entre las palas del AGV que se mueve solidariamente con ellas, y otra en una posición fija. También se dispone de una fuente de luz compuesta por varios leds para garantizar las condiciones de iluminación necesarias.

Mediante las dos cámaras se generan imágenes en 3D, que posteriormente se tratan con filtros y una ventana deslizante para elegir los candidatos a pallets de la imagen obtenida.

Real-time Pallet Localization with 3D Camera Technology for Forklifts in Logistic Environments

En este estudio, se utiliza una cámara Kinect v2 de Microsoft montada en las palas del AGV. Mediante un algoritmo de región creciente (Region growing algorithm) y segmentación de las nubes de puntos que se reciben como entrada, se buscan particularidades que identifican los pallets, concretamente los bloques de las patas, y la superficie que los une.

Cuando se detecta un candidato a pallet, se comprueban las medidas que la cámara está viendo con las medidas reales de un pallet europeo con un pequeño margen de error y así verificar el pallet.

Similar caso que el utilizado en el proyecto, diferenciándose en el tipo de tecnología utilizada para la captura de datos.

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.