



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**Seguimiento de líneas basado
en OpenCV para AGVs
Documentación Técnica**



Presentado por Antonio de los Mozos Alonso
en Universidad de Burgos — 27 de junio
de 2018

Tutor: Jesús Enrique Sierra García

Índice general

Índice general	I
Índice de figuras	III
Índice de tablas	IV
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
A.3. Estudio de viabilidad	7
Apéndice B Especificación de Requisitos	9
B.1. Introducción	9
B.2. Objetivos generales	9
B.3. Catalogo de requisitos	9
B.4. Especificación de requisitos	9
B.5. Casos de uso	10
Apéndice C Especificación de diseño	11
C.1. Introducción	11
C.2. Diseño de datos	11
C.3. Diseño procedimental	11
C.4. Diseño arquitectónico	11
Apéndice D Documentación técnica de programación	13
D.1. Introducción	13
D.2. Estructura de directorios	13

D.3. Manual del programador	13
D.4. Compilación, instalación y ejecución del proyecto	13
D.5. Pruebas del sistema	13
Apéndice E Documentación de usuario	15
E.1. Introducción	15
E.2. Requisitos de usuarios	15
E.3. Instalación	15
E.4. Manual del usuario	15
Bibliografía	17

Índice de figuras

Índice de tablas

Apéndice A

Plan de Proyecto Software

A.1. Introducción

En esta sección se detallará la planificación del proyecto.

Como planificación se entienden las distintas etapas por las que va a pasar el proyecto hasta su finalización, y la duración de cada etapa. También tenemos que tener en cuenta la viabilidad del proyecto, la posibilidad de que el proyecto sea realizado.

En este proyecto no ha habido una planificación demasiado estricta en cuanto a tiempos de entrega, ya que es un proyecto con una gran componente de investigación. Aún así, se han seguido las directrices de las metodologías ágiles, en concreto dentro del marco de Scrum[2].

La viabilidad económica de una investigación puede llegar a ser difícil de estimar, ya que se intenta agotar los recursos disponibles al máximo, tanto temporalmente como económicamente. Aun así, llegado al punto temporal al que se ha llegado, podemos analizar la viabilidad económica en retrospectiva.

A.2. Planificación temporal

Inicialmente planteamos usar una metodología mas acorde a una investigación, basada en la exploración(etapa de investigación) y evolución(en base a realizar pruebas). Como el producto final ha de ser un software que recoja los resultados de todas las investigaciones y pruebas, finalmente aplicamos una metodología ágil, Scrum. Aun así, dentro de los sprints e iteraciones, es donde se han hecho las etapas de exploración y evolución.

El alumno tomará el papel de equipo de desarrollo y de Scrum Master. El tutor hará de cliente.

Los resultados de las diferentes pruebas realizadas se han ido registrando en el diario de pruebas[1].

Cabe decir que a pesar de que el proyecto estaba planeado para iniciarse en el segundo cuatrimestre, hemos empezado en el primero por motivos académicos del alumno, para poder tener más tiempo de desarrollo.

Sprint 0: 1/10/17-9/11/17

El proyecto parte de un sprint inicial, en el que se explicó el contexto y los elementos con los que se tenía que trabajar. Se preparó del entorno de trabajo, con el hardware y software necesario. Se buscó la mejor forma para realizar el proyecto, explorando dos opciones, trabajar con videos grabados, o trabajar conun streaming en directo.

Tareas

- Preparación de las librerías Python
- Preparación del IDE
- Preparación del Repositorio
- Preparación cámara y aplicación IPWebcam
- Preparación soporte cámara

Backlog

En este sprint lo que hemos hecho es la fase de preparación del proyecto, preparar todos los materiales. También se dieron los conocimientos básicos al alumno en reuniones con el tutor, sobre el contexto de trabajo, que se pretendía realizar, que se pretendía probar y que se pretendía conseguir.

Investigación

En esta fase se investigaron las formas de organizar los elemtnos con los que se iba a trabajar. Inicialmente se propuso usar la WebCam integrada en el portátil, pero finalmente se optó por usar un smartphone que retransmitiera vídeo, ya que nos da mucha mayor libertad.

Esto podría asociarse a las formas de organización de los diferentes elementos (cámara, ordenador y sistema de control del AGV) vistas en la memoria. Resaltar que para este proyecto solo hemos trabajado con la cámara y el ordenador, puesto que esta enfocado a la detección de líneas, no al control de un AGV.

Pruebas

Se realizaron las pruebas de conexión pertinentes. Inicialmente se usaba el router de la red local de casa, con el smartphone y el ordenador conectados vía wifi. Al ser la conexión inalámbrica, y el router estar recibiendo peticiones de otros dispositivos, había más latencia y pérdida de paquetes, lo que generaba un mal rendimiento de la transmisión de vídeo.

A si que finalmente, se optó por usar un router específico para este sistema, conectando por cable el ordenador para optimizar al máximo la conexión y evitar la pérdida de paquetes, y conectado vía wifi al smartphone.

Sprint 1: 9/11/17-23/11/17

En el primer sprint realizado se buscó la forma de obtener la tasa de Fps necesaria para cumplir con los requisitos establecidos, así como realizar una prueba de binarización de la línea guía, hecha de unos materiales concretos, buscar diferentes algoritmos, probarlos, generar una funcionalidad para medir la luminosidad y así establecer los parámetros bajo los que se realizan las pruebas.

Tareas

- Obtener una tasa de Fps que permitiera realizar el procesamiento de imágenes en menos de 2 centésimas de segundo.
- Crear la plantilla de fondo blanco y línea negra para hacer la binarización por luminosidad.
- Buscar diferentes algoritmos de binarización por luminosidad.
- Probar los diferentes algoritmos, dejar registro de las pruebas, y elegir el que mejor funcione.
- Generar una funcionalidad para medir la luminosidad en la imagen.
- Crear un archivo donde ir implementando todas las funciones probadas, con los parámetros necesarios.

Backlog

En este sprint hemos realizado los ajustes pertinentes en la cámara para cumplir con las restricciones de Fps, hemos buscado, probado y registrado diferentes algoritmos de binarización, y finalmente hemos dejado una implementación con el mejor algoritmo encontrado. Además para realizar las pruebas se ha necesitado una funcionalidad que permita medir la luminosidad de la imagen, por lo que también se ha implementado esta funcionalidad.

Investigación

Se han investigado los diferentes algoritmos de binarización por luminosidad.

Pruebas

Se han realizado pruebas con todos los algoritmos, estableciendo parámetros y dejando registros de los resultados.

Sprint 2: 23/11/17-14/12/17

En el segundo sprint buscamos el tomar medidas de la línea sobre la imagen, para comprobar la eficacia de los metodos de binarizacion, se probaron diferentes funciones que median el ancho real y el ancho calculado de la línea, estando la línea en perpendicular a la cámara.

Tareas

- Diseñar función que permita realizar medidas de la línea sobre la imagen.
- Diseñar función para estimar el ancho de la línea en perspectiva.
- Probar las funciones, y comprobar el fallo en la medición de la anchura en diferentes condiciones de luminosidad.

Backlog

Se implementaron las funciones necesarias para medir la línea normal y en perspectiva, estando perpendicular a la cámara. Aunque estas funciones son correctas, la funcionalidad para resolver la distorsión de perspectiva es muy limitada, funcionando solo cuando la línea esta en perpendicular a la cámara.

Investigación

Primeros inicios en la resolución de la distorsión de perspectiva, buscando el saber el ancho real de la línea en diferentes zonas de la imagen.

Pruebas

Se realizaron pruebas de medición de la línea y comparación con la anchura real calculada.

Sprint 3: 12/2/18-30/04/18

En el tercer sprint buscamos tomar medidas de una forma más avanzada, no solo si la línea esta de forma perpendicular a la cámara, sino estando en cualquier posición. Para esto necesitaremos entrar en la problemática de la perspectiva. Además iniciaremos la investigación del sistema de guiado a través de la trayectoria.

Tareas

- Avanzar en la resolución de distorsión por perspectiva
- Implementar una funcionalidad que permita obtener la vista de pájaro.
- Comprobar que la imagen en vista de pajarero guarda las mismas proporciones que la realidad.
- Implementar una funcionalidad que permita obtener la trayectoria (centro de la línea).
- Indagar en los sistemas de guiado, realizando una funcionalidad que calcule la distancia de la trayectoria al centro de la imagen.

Backlog

En este sprint se hacen avances muy importantes, sobre todo la resolución total de la distorsión por perspectiva, problemática principal del proyecto. Se consigue generar una funcionalidad que genera la vista de pájaro guardando las proporciones reales. Se obtiene la trayectoria mediante los bordes de la línea. Iniciamos el desarrollo del sistema de guiado.

Investigación

Se investigan formas de resolver la distorsión por perspectiva adecuadamente, así como de relacionar la funcionalidad que ya tenemos con esta problemática. Se investigan los diferentes tipos de sistema de guiado.

Pruebas

Las pruebas realizadas en el ámbito de la distorsión por perspectiva consisten principalmente en enganchar las diferentes funcionalidades que tenemos y que funcionen en conjunto, y comprobar que la vista de pajarero guarda las proporciones de la realidad, para esto medimos ángulos. Dentro de los sistemas de guiado, hacemos una prueba que nos permite ver como actúa el sistema de guiado implementado.

Sprint 4: 30/04/18-28/06/18

En el cuarto sprint buscamos nuevas formas de binarización, en concreto la binarización por color. Crear una interfaz para dar una funcionalidad central al proyecto, y finalizar el sistema de guiado y su forma de representación. Además se planteó la posibilidad de integrar las librerías de canKin para conectar el sistema de guiado básico con un robot, si la evolución temporal lo permitía.

Tareas

- Indagar en la binarización por color.
- Implementar un sistema de binarización por color compatible con el resto de funcionalidad.
- Implementar una interfaz para dar una cohesión a todas las funcionalidades implementadas.
- Mostrar en la interfaz toda la información de los algoritmos.
- Integrar librerías canKin, opcional, en función de la evolución del resto de tareas.
- Generar ejecutable de la aplicación para hacer el release[3].

Backlog

Se cumplieron todas las tareas obligatorias, tanto implementar un sistema de binarización por color como la interfaz que conecta todas las funcionalidades implementadas. Además se consiguió mostrar la información en las propias ventanas de OpenCV sin sacrificar rendimiento de la aplicación. Se hizo el release de la aplicación en el repositorio.

Investigación

Se investigó acerca de los algoritmos de binarización por color, dando lugar a la implementación de un sistema de binarización por color.

Pruebas

Las pruebas realizadas parten desde los distintos materiales que se podían usar para la binarización por color. Inicialmente se uso cartulina amarilla, no funcionaba adecuadamente hasta que remarcamos su borde con cinta aislante negra, lo cual resaltó el borde e hizo que funcionara bien.

A.3. Estudio de viabilidad

Viabilidad económica

En este apartado se analizarán los costes del proyecto.

Costes de personal

Costes de hardware

Costes de software

Costes de materiales e infraestructura

Costes de totales

Viabilidad legal

En este apartado veremos las licencias del software, compatibilidades entre ellas y la licencia final de la aplicación. También, aunque no se ha trabajado con ello, veremos la legislación de robots y vehículos autónomos en España, ya que el software sí esta pensado para dirigir a un vehículo de forma autónoma.

Software

Documentación

Legislación de robots autónomos

Apéndice B

Especificación de Requisitos

B.1. Introducción

B.2. Objetivos generales

B.3. Catalogo de requisitos

B.4. Especificación de requisitos

R1

Versión

Autor

Descripción

Precondición

Secuencia Normal Paso Acción

1. Hacer algo

Postcondición

Excepciones

Importancia

Comentarios

B.5. Casos de uso

Apéndice C

Especificación de diseño

- C.1. Introducción
- C.2. Diseño de datos
- C.3. Diseño procedimental
- C.4. Diseño arquitectónico

Apéndice D

Documentación técnica de programación

- D.1. Introducción
- D.2. Estructura de directorios
- D.3. Manual del programador
- D.4. Compilación, instalación y ejecución
del proyecto
- D.5. Pruebas del sistema

Apéndice E

Documentación de usuario

- E.1. Introducción
- E.2. Requisitos de usuarios
- E.3. Instalación
- E.4. Manual del usuario

Bibliografía

- [1] Antonio de los Mozos Alonso. Diario de pruebas, 2018. [Internet; consultado 27-junio-2018] <https://github.com/ama0114/TFG-OpenCV/blob/master/Documentaci%C3%B3n/Diario.txt>.
- [2] Wikipedia. Scrum (desarrollo de software), 2018. [Internet; consultado 27-junio-2018] [https://es.wikipedia.org/wiki/Scrum_\(desarrollo_de_software\)](https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software)).
- [3] Wikipedia. Software release life cycle, 2018. [Internet; consultado 27-junio-2018] https://en.wikipedia.org/wiki/Software_release_life_cycle.