

Localización de productos en entorno comercial mediante códigos de barras usando un robot móvil

Autor: Juan de Dios Herrera Hurtado

Tutor: Miguel Ángel Ridao Carlini

Trabajo Fin de Grado

Grado en Ingeniería Electrónica, Robótica y Mecatrónica

Dpto. de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2023

Proyecto Fin de Grado

Grado en Ingeniería Electrónica Robótica y Mecatrónica

**Localización de productos en entorno comercial mediante códigos de barras usando un robot móvil**

Autor:

Juan de Dios Herrera Hurtado

Tutor:

Miguel Ángel Ridao Carlini

Profesor titular

Dpto. de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2023

Trabajo Fin de Grado: Localización de productos en entorno comercial mediante códigos de barras usando un robot móvil

|  |  |
| --- | --- |
| Autor: | Juan de Dios Herrera Hurtado |
| Tutor: | Miguel Ángel Ridao Carlini |

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocal/es:

Secretario:

Acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Agradecimientos

Los estilos adoptados por nuestra Escuela y utilizada en este texto es una versión y adaptación a Word® del la versión LATEX que el Prof. Payán realizó para un libro que desde hace tiempo viene escribiendo para su asignatura. Por ello, la Escuela le está agradecida. Por otro lado, la adaptación se hizo sobre un formato que el prof. Aguilera arregló, basándose en su tesis doctoral. Su aportación ha sido muy relevante para que este formato vea la luz. Esta adaptación la llevamos a cabo el alumno Silvio Fernández, becario del Centro de Cálculo, y yo mismo, sobre un trabajo preliminar del alumno Julián José Pérez Arias.

A esta hoja de estilos se le incluyó unos nuevos diseños de portada. El diseño gráfico de las portadas para proyectos fin de grado, carrera y máster, está basado en el que el prof. Fernando García García, de la Facultad de Bellas Artes de nuestra Universidad, hiciera para los libros, o tesis, de la sección de publicación de nuestra Escuela. Nuestra Escuela le agradece que pusiera su arte y su trabajo a nuestra disposición.

*Juan de Dios Herrera Hurtado*

*Sevilla, 2023*

Resumen

El trabajo que se presenta es una continuación de una serie de Trabajos Fin de Grado ya realizados sobre un proyecto acordado entre la empresa Tier1 y la ETSI.

La idea del proyecto en su conjunto es, mediante un robot móvil, realizar un mapeo del comercio en cuestión identificando pasillos que serán las celdas libres del mapa y estanterías que serán los obstáculos, localizar las coordenadas dentro de ese mapa de cada producto y, finalmente, planificar una ruta de coste mínimo y realizarla pasando por los diferentes productos que el cliente haya indicado que desea comprar.

En este trabajo nos centraremos en la segunda parte, es decir, la localización de los productos en el mapa que es una parte puramente basada en visión por computador, para ello, el robot deberá recorrer todas las estanterías del comercio identificando los productos por su código de barras y asignándoles unas coordenadas (x,y,z) en el mapa previamente calculado. Tomaremos como referencia [1], el trabajo de Pedro Tito Macías Roselló, que fue el último compañero en intervenir en el trabajo y en modificar esta segunda parte concretamente.

Antes de intervenir en el proyecto, el robot era capaz de identificar códigos de barras sobre una pared de azulejos, pero a veces identificaba falsos positivos, esto significa que el programa indicaba que en un lugar de la imagen existía un código de barras cuando realmente no era así. El otro problema que presentaba la propuesta previa era que estamos trabajando sobre una pared prácticamente lisa, luego en caso de introducir algún elemento más en la imagen, la efectividad del algoritmo se vería drásticamente mermada.

Por ello, los dos objetivos que se proponen cumplir con este trabajo son resolver ambos problemas expuestos en el párrafo anterior. En primer lugar, lograr que el programa identifique códigos de barras en un entorno comercial real, que es donde queremos implementar este proyecto y, en segundo lugar, reducir las probabilidades de obtener falsos positivos.

En la práctica, no se dispone del robot móvil en un entorno comercial real, sería susceptible de recibir algún golpe accidentalmente por parte de una persona. Además, se ha tratado de que los experimentos realizados emulen lo más fielmente posible las condiciones reales en las que se encontraría el robot. Luego se tomarán imágenes con la cámara del móvil y serán estas imágenes las que se procesen.

Abstract

The work presented here is a continuation of a series of Final Degree Projects already carried out on a project agreed between the company Tier1 and the ETSI.

The idea of the project as a whole is, by means of a mobile robot, to map the shop in question by identifying aisles that will be the free cells on the map and shelves that will be the obstacles, to locate the coordinates of each product on the map and, finally, to plan a minimum-cost route and carry it out by passing through the different products that the customer has indicated he wishes to buy.

In this work we will focus on the second part, that is, the location of the products on the map, which is a part purely based on computer vision. To do this, the robot must go through all the shelves of the shop identifying the products by their barcode and assigning them coordinates (x,y,z) on the previously calculated map. We will take as a reference [1], the work of Pedro Tito Macías Roselló, who was the last colleague to intervene in the work and to modify this second part specifically.

Before intervening in the project, the robot was able to identify barcodes on a tiled wall, but sometimes it would identify false positives, meaning that the programme would indicate that there was a barcode in one place in the image when there really wasn't. The other problem with the previous proposal was that we were working on a practically smooth wall, so if we introduced any other element into the image, the effectiveness of the algorithm would be drastically reduced.

Therefore, the two objectives of this work are to solve both problems outlined in the previous paragraph. Firstly, to get the program to identify barcodes in a real commercial environment, which is where we want to implement this project, and secondly, to reduce the probability of obtaining false positives.

In practice, the mobile robot is not available in a real commercial environment, as it would be susceptible to being accidentally hit by a person. In addition, we have tried to ensure that the experiments carried out emulate as closely as possible the real conditions in which the robot would find itself. Images will then be taken with the mobile phone camera and these images will be processed.

Índice

Agradecimientos vii

Resumen ix

Abstract xi

Índice xiii

Índice de Tablas xiv

Índice de Figuras xvi

Notación xviii

1 Introducción 11

1.1 Motivación 11

1.2 El código de barras 11

1.3 Descripción del problema 11

1.4 Objetivos del proyecto 12

1.4.1 Posibilidad de aplicar el algoritmo de visión en un entorno real 12

1.4.2 Robustez frente a la detección de falsos positivos 12

2 Software utilizado 13

2.1.1 Visual Studio Code (versión 1.78.2) 13

2.1.2 Python (versión 3.10.4) 13

2.1.3 OpenCV (versión 4.5.5) 13

3 Hardware utilizado 15

4 Consideraciones a la hora de tomar la imagen 17

4.1 Posición horizontal 17

4.2 Tamaño de la imagen 17

4.3 Zoom 18

4.4 Bandas superior e inferior cerca de los límites de arriba y de debajo de la imagen 18

4.5 Perpendicularidad del móvil respecto al suelo a la hora de tomar la imagen 18

5 Explicación del algoritmo 18

Referencias 19

Índice de Conceptos 20

Glosario 22

# **Índice de Tablas**

Tabla 3‑1. Características de la cámara 15

# **Índice de Figuras**

Figura 1‑1. Ejemplo de código de barras EAN 13 [2]. 11

Figura 1‑2. Esquematización de lo que se pretende realizar. 12

Figura 2‑1. Logo Visual Studio Code [3]. 13

Figura 2‑2. Logo Python [4]. 13

Figura 2‑3. Logo OpenCV [5]. 14

Figura 3‑1. Cámara Xiaomi Redmi Note 10 Pro [7] 15

Figura 4‑1. Relación distancia respecto a la estantería y amplitud de la estantería que entra en la imagen. 17

Figura 4‑2. Tamaño de las imágenes obtenidas. 18

Figura 4‑3. Consulta de las dimensiones de la imagen en sus propiedades. 18

Notación

|  |  |
| --- | --- |
| ETSI | Escuela Técnica Superior de Ingeniería |
| c.t.p. | En casi todos los puntos |
| c.q.d. | Como queríamos demostrar |
| ∎ | Como queríamos demostrar |
| e.o.c. | En cualquier otro caso |
| e | número e |
| IRe | Parte real |
| IIm | Parte imaginaria |
| sen | Función seno |
| tg | Función tangente |
| arctg | Función arco tangente |
| sen | Función seno |
| sin*xy* | Función seno de *x* elevado a *y* |
| cos*xy* | Función coseno de *x* elevado a *y* |
| Sa | Función sampling |
| sgn | Función signo |
| rect | Función rectángulo |
| Sinc | Función sinc |
| ∂y ∂x  *x*◦ | Derivada parcial de *y* respecto  Notación de grado, *x* grados. |
| Pr(*A*) | Probabilidad del suceso *A* |
| SNR | Signal-to-noise ratio |
| MSE | Minimum square error |
| : | Tal que |
| < | Menor o igual |
| > | Mayor o igual |
| \ | Backslash |
| ⇔ | Si y sólo si |

# Introducción

## Motivación

Como se ha introducido en el resumen, este proyecto continúa una serie de trabajos ya realizados sobre una propuesta acordada entre la empresa Tier1 y la ETSI. En concreto, nosotros continuaremos el último trabajo realizado sobre el proyecto perteneciente al alumno Pedro Tito Macías Roselló y que denominaremos como “TFG de referencia” de aquí en adelante.

## El código de barras

[2] El código de barras fue creado en 1952 por los inventores [Joseph Woodland](https://es.wikipedia.org/wiki/Norman_Joseph_Woodland), [Jordin Johanson](https://es.wikipedia.org/w/index.php?title=Jordin_Johanson&action=edit&redlink=1) y [Bernard Silver](https://es.wikipedia.org/w/index.php?title=Bernard_Silver&action=edit&redlink=1), pero no fue posible su uso comercial hasta el año 1966. Se trata de un código basado en la representación de un conjunto de líneas paralelas de distinto grosor, color y espaciado que en su conjunto contienen una determinada información, es decir, las barras y espacios del código representan pequeñas cadenas de caracteres. De esta forma, los códigos permiten identificar un artículo de manera única y no ambigua debido a que el código de barras asociado a una determinada cadena de caracteres es único. Dicho de otro modo, cada código de barras representa una determinada cadena de caracteres y solo una cadena de caracteres.

Patrón de fondo

Descripción generada automáticamente

Figura ‑. Ejemplo de código de barras EAN 13 [2].

Existen varios tipos de códigos de barras, aunque los más extendidos son los del tipo EAN13. Los códigos de barras del tipo EAN (European Article Number), son un tipo de código de barras de carácter numérico adoptado por más de cien países y cerca de un millón de empresas. El 13 indica el número de caracteres que posee la cadena numérica, en la imagen anterior se pueden ver los 13 dígitos de la cadena correspondiente a ese código de barras.

A pesar de que los códigos de barras llamados matriciales o bidimensionales como los códigos QR poseen numerosas ventajas frente a los códigos de barras unidimensionales, no son estos los que encontramos en los comercios hoy en día, sino los convencionales de una dimensión. De ahí el motivo de que se quiera trabajar con códigos de barras unidimensionales y no con códigos QR por ejemplo, simplemente porque es lo que está presente actualmente en los comercios.

## Descripción del problema

El trabajo que se presenta es un trabajo puramente de visión por computador, ya que se basa en obtener información, en nuestro caso identificar los posibles códigos de barras, presentes en la imagen realizada con la cámara del robot, aunque en nuestro caso al no disponer del mismo en un entorno comercial real, se usará como ya se ha comentado la cámara del móvil para obtener las imágenes.

A continuación, se muestra un esquema de lo que se pretende que el algoritmo realice:

Sistema

Imagen de la estantería del comercio

Coordenadas (x,y,z) de los productos identificados en la imagen

Figura ‑. Esquematización de lo que se pretende realizar.

En el capítulo “Explicación del algoritmo”, desarrollaremos el sistema que por ahora hemos identificado como una “caja negra”.

Dado que para este trabajo no se hace uso del robot real, no se considera oportuno mencionar las características y elementos del robot, aunque si resulta de interés, se puede consultar en el TFG de referencia [1] en el capítulo “Hardware”.

Asimismo, también se puede consultar más en detalle las funcionalidades adicionales previamente mencionadas en el resumen, como la creación del mapa o la creación de trayectorias una vez se recibe el listado de productos que el cliente desean comprar. Todo ello se puede consultar en el capítulo “Marco teórico” en la sección “Alcance” [1].

## Objetivos del proyecto

La meta de este proyecto es mejorar el sistema de visión ya existente haciéndolo más robusto y flexible al entorno en el que se aplique. Es por ello por lo que se marcan dos objetivos claros y que detallaremos a continuación.

### Posibilidad de aplicar el algoritmo de visión en un entorno real

En primer lugar y más importante, la plataforma robótica como producto final es algo que va a estar funcionando en un comercio real, luego debemos de elaborar un algoritmo que sea capaz de atajar todas las dificultades que ello plantea con la mayor efectividad posible.

Por este motivo, desde el comienzo de este trabajo se ha estado trabajando con imágenes tomadas a estanterías de comercios reales para asegurarnos que este objetivo se cumpliese en mayor o menor medida.

### Robustez frente a la detección de falsos positivos

Por otro lado, se pretende reducir el número de falsos positivos, ya que esto ralentiza directamente el funcionamiento de esta parte de localización de productos como se explicará más adelante.

No obstante, se avanza que este mayor tiempo de ejecución se debe a que una vez identificamos donde se encuentran los potenciales códigos de barras, debemos hacer zoom a dichas zonas de la imagen para que un algoritmo de identificación de códigos de barras sea capaz de decodificarlo y obtener el código numérico correspondiente a ese producto. Luego, si hacemos zoom a un lugar donde no existe ningún código de barras, el algoritmo no será capaz de decodificar nada y se habrá perdido ese tiempo que se podría haber empleado identificando un código de barras que sí existía.

# Software utilizado

En este capítulo comentaremos los programas utilizados para llevar a cabo el trabajo.

### Visual Studio Code (versión 1.78.2)

[3] Visual Studio Code es un editor de código fuente desarrollado por Microsoft. Se encuentra disponible para Windows, macOS y Linux. Contiene infinidad de extensiones para lenguajes de programación como C#, C++, Java o Python entre otros.

Icono

Descripción generada automáticamente

Figura ‑. Logo Visual Studio Code [3].

### Python (versión 3.10.4)

[4] Python es un lenguaje de programación interpretado, interactivo y orientado a objetos. Incluye módulos, excepciones, tipos dinámicos, tipos de datos dinámicos de muy alto nivel y clases. Admite múltiples paradigmas de programación más allá de la programación orientada a objetos, como la programación procedimental y funcional. Además, combina una potencia notable con una sintaxis muy clara. Se ejecuta en muchas variantes de Unix, incluidos Linux y macOS, y en Windows.

Logotipo, nombre de la empresa

Descripción generada automáticamente

Figura ‑. Logo Python [4].

### OpenCV (versión 4.5.5)

[5] OpenCV (Open Source Computer Vision Library) es una biblioteca de software de aprendizaje automático y visión artificial de código abierto. OpenCV se creó para proporcionar una infraestructura común para las aplicaciones de visión por computador y para acelerar el uso de la percepción de la máquina en los productos comerciales. La biblioteca dispone de más de 2500 algoritmos optimizados que van desde identificar contornos en una imagen, hasta cosas tan sofisticadas como por ejemplo clasificar acciones humanas en vídeos o extraer modelos 3D de objetos.

Icono

Descripción generada automáticamente

Figura ‑. Logo OpenCV [5].

# Hardware utilizado

Para tomar las imágenes se ha usado la cámara de un Xiaomi Redmi Note 10 Pro. Se recoge ahora en una tabla las características de la cámara trasera que es la que usaremos y que consta de un total de tres cámaras y un sensor de profundidad. Toda esta información se ha extraído de la página oficial de Xiaomi [6].

Tabla ‑. Características de la cámara.

|  |  |  |
| --- | --- | --- |
| Tipo | Tamaño del sensor (megapíxeles) | Características |
| Cámara ultra gran angular | 108 | Tamaño de píxel de 0,7 μm, superpíxel 9 en 1 de 2,1 μm  Tamaño del sensor de 1/1,52”  f / 1,9 |
| Cámara ultra gran angular | 8 | Campo de visión de 118°  f / 2,2 |
| Cámara telemacro | 5 | f / 2,4  AF (autofocus) |
| Sensor de profundidad | 2 | f / 2,4 |

Se muestra ahora una imagen de la cámara y la correspondencia con cada uno de los elementos mencionados en la tabla anterior.

Diagrama, Esquemático

Descripción generada automáticamente

Figura ‑. Cámara Xiaomi Redmi Note 10 Pro [7]

# Consideraciones a la hora de tomar la imagen

## Posición horizontal

En primer lugar, la posición del móvil será horizontal. De esta manera, logramos una imagen que abarca más estantería respecto a si la posición del móvil fuese vertical, todo ello considerando que se toman las imágenes a la misma distancia respecto a la estantería. Mientras más alejados estemos de la estantería, más estantería entrará dentro de la imagen. A continuación, se muestra esto más claramente en un dibujo:

Estanterías

d1

d2

A

B

Figura ‑. Relación distancia respecto a la estantería y amplitud de la estantería que entra en la imagen.

Por lo tanto:

|  |  |
| --- | --- |
|  | (‑) |

## Tamaño de la imagen

Usando el móvil de manera horizontal, se obtienen imágenes de 3000 píxeles de alto y 4000 píxeles de ancho tal y como se muestra en la imagen.

4000 píxeles

3000 píxeles

Imagen que contiene tabla, comida, llenado, alimentos

Descripción generada automáticamente

Figura ‑. Tamaño de las imágenes obtenidas.

Esta información se obtiene fácilmente haciendo uso del comando ***.shape*** o simplemente consultando las propiedades de algunas de las imágenes tomadas.

***Texto

Descripción generada automáticamente***

Figura ‑. Consulta de las dimensiones de la imagen en sus propiedades.

## Zoom

Se ha usado la función ***imread*** de la librería de OpenCV

## Bandas superior e inferior cerca de los límites de arriba y de debajo de la imagen

## Perpendicularidad del móvil respecto al suelo a la hora de tomar la imagen

# Explicación del algoritmo

Mayoría de conocimientos usados en el algoritmo extraídos de las diapositivas de la asignatura de “Sistemas de Percepción”

# Referencias

|  |  |
| --- | --- |
| [1] | P. T. Macías Roselló, «Robot movil con ros para la lectura de codigos de barras mediante zoom optico en un entorno comercial,» *Dirigido por Miguel Angel Ridao Carlini,* 2021. |
| [2] | *Web Wikipedia,* [En línea]. [Último acceso: 2023]. Disponible en: https://code.visualstudio.com/. |
| [3] | *Web Visual Studio Code,* [En línea]. [Último acceso: 2023]. Disponible en: https://code.visualstudio.com/. |
| [4] | *Web Python,* [En línea]. [Último acceso: 2023]. Disponible en: https://www.python.org/. |
| [5] | *Web OpenCV,* [En línea]. [Último acceso: 2023]. Disponible en: https://opencv.org/. |
| [6] | *Web Xiaomi,* [En línea]. [Último acceso: 2023]. Disponible en: https://www.mi.com/es. |
| [7] | *Web XiaomiAdictos,* [En línea]. [Último acceso: 2023]. Disponible en: https://www.xiaomiadictos.com/. |

# **Índice de Conceptos**

conceptos 9

# **Glosario**

ISO: International Organization for Standardization 4

UNE: Una Norma Española 4