

18 DE DICIEMBRE DE 2024



**SISTEMA DE ESTACIONAMIENTO
ASISTIDO
PROCESAMIENTO EMBEDIDO DE SEÑALES**

DANTE FAYANÁS - JULIAN VAN ROEY
UNIVERSIDAD TECNOLOGIA NACIONAL - FRA

Contenido

- 1. Introducción 2
- 2. Planteamiento del problema 2
- 3. Objetivos 2
 - Objetivo general 2
 - Objetivos específicos 2
 - Etapas propuestas 2
- 4. Justificación..... 3
- 5. Alcance del proyecto 3
- 6. Etapas de desarrollo..... 3
- 7. Herramientas de software y equipos utilizados..... 6
- 8. Resultados esperados 6
- 9. Conclusión 6
- 10. Referencias..... 7

1. Introducción

Este proyecto tiene como objetivo desarrollar un sistema de asistencia para el estacionamiento manual en espacios delimitados mediante cámaras. Utilizando visión asistida por computadora y técnicas de procesamiento de imágenes, el sistema identifica vehículos estacionados, mide distancias y emite alertas sonoras para guiar al conductor durante el estacionamiento.

2. Planteamiento del problema

En estacionamientos con espacios reducidos o mal señalizados, los conductores pueden tener dificultades para maniobrar de manera precisa. Esto puede resultar en accidentes menores, daños a los vehículos y tiempo perdido. Un sistema de asistencia basado en cámaras permite mejorar la precisión del estacionamiento y reducir estos problemas.

3. Objetivos

Objetivo general

Desarrollar un sistema de monitoreo y asistencia para estacionamientos mediante cámaras, utilizando visión por computadora para identificar vehículos y emitir guías auditivas para el estacionamiento manual.

Objetivos específicos

- Implementar un sistema de detección de vehículos basado en técnicas de visión utilizando Raspberry Pi 4.
- Calcular distancias entre el vehículo y puntos de referencia en el estacionamiento.
- Documentar cantidad y ubicación de los vehículos (almacenando los números e imágenes de las patentes identificadas).
- Generar alertas sonoras en tiempo real para guiar al conductor.
- Diseñar un algoritmo eficiente que se pueda aplicar en múltiples espacios de estacionamiento.

Etapas propuestas

Etapa 1:

- 1) Reconocimiento óptimo de las patentes, incluyendo los distintos tipos de patentes a través de los años (Patente Mercosur y Patente Rep. Arg)
- 2) Medición de distancias, ángulos y sus valores respecto de la cámara.

Etapa 2:

- 1) Implementación y adaptación del programa a Raspberry Pi 4
- 2) Interfaz gráfica para el usuario
- 3) Método de calibración

4. Justificación

El uso de cámaras para asistir en el estacionamiento ofrece ventajas significativas, como la prevención de colisiones menores y la optimización del uso del espacio. Este sistema mejora la experiencia del usuario, fomenta la adopción de tecnologías avanzadas en estacionamientos y sienta las bases para futuras integraciones con sistemas automatizados.

5. Alcance del proyecto

- Detección de vehículos estacionados: Identificación y localización precisa mediante cámaras en tiempo real.
- Documentación de las patentes y cantidad de patentes registradas.
- Medición de distancias: Cálculo de la proximidad entre el vehículo y otros objetos en el entorno.
- Alertas sonoras: Notificaciones en tiempo real para guiar al conductor.
- Escalabilidad: Diseño modular que permite la implementación en múltiples espacios de estacionamiento.

6. Etapas de desarrollo

- 1) Investigación y análisis: Estudio de métodos de detección de vehículos y cálculo de distancias utilizando visión por computadora. En esta primera etapa nos enfocamos en encontrar información en internet de cómo se podía realizar nuestro proyecto, y que debíamos utilizar.
- 2) Implementación inicial: Uso de bibliotecas como OpenCV y NumPy para procesar imágenes y detectar características clave. Luego de encontrar las bibliotecas necesarias para nuestro proyecto investigamos el alcance de estas, utilizando la literatura disponible para entender como funciona cada una y adaptarla a nuestro proyecto y necesidades, nos encontramos inicialmente con el problema que estábamos detectando una patente mas ancha que la utilizada en argentina, eso fue porque partimos de una medida modelo de una patente europea para los primeros intentos de detección.
- 3) Desarrollo del sistema de medición de distancia:

Para calcular las distancias, se emplearon parámetros conocidos como la longitud focal y las dimensiones del objeto de referencia. Una vez definido el entorno de medición y evaluadas las características técnicas de la cámara utilizada (Dahua DH-HAC-T1A21N), se realizaron los cálculos basados en las dimensiones de las placas de matrícula utilizadas en Argentina (MERCOSUR y REP. ARG). Ambas variantes, con medidas específicas, se consideraron para establecer la relación de aspecto necesaria.

El procesamiento de las imágenes comienza con la conversión a escala de grises, seguida de la aplicación de un filtro gaussiano para suavizar la imagen y la detección de bordes. A continuación, se identifican todos los contornos cerrados en la imagen y se aplican una serie de condiciones para filtrar los resultados y evitar descartar contornos relevantes:

- Tamaño del contorno.
- Que el contorno sea un polígono regular.
- Que el polígono sea más largo que alto.
- Que no exceda un ángulo de inclinación predefinido.

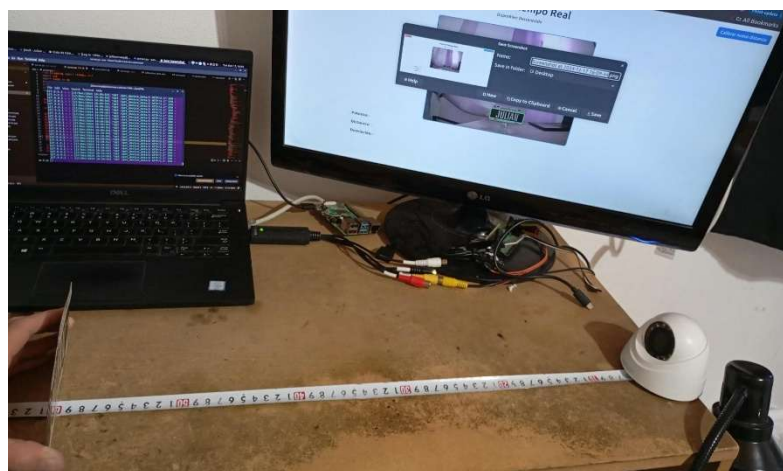
Tras cumplir estas condiciones, se realiza una lectura de texto sobre los contornos restantes. La validación del texto considera si la matrícula contiene 6 caracteres (REP. ARG) o 7 caracteres (MERCOSUR). Con esta información, se calcula la distancia entre la cámara y el vehículo.

Para determinar el ángulo del vehículo con respecto a la cámara, se utiliza la relación de aspecto ideal, que corresponde a la orientación paralela de la matrícula respecto a la cámara. Esta relación se compara con la relación de aspecto detectada en tiempo real, lo que permite identificar el ángulo actual y proporcionar indicaciones al conductor sobre cómo corregir su dirección.

Debajo se muestran 3 imágenes en tiempo real de cuando detecta erróneamente la patente (contorno rojo) y cuando la valida correctamente (contorno verde).



Entorno de prueba:



- 4) Integración del sistema de alertas: Uso de un sintetizador de voz para proporcionar instrucciones al conductor. En primera instancia se había propuesto utilizar avisos sonoros, pero se optó por mostrar indicaciones visuales en la pantalla mediante imágenes que le indican al conductor que debe realizar.

Debajo se detallan las imágenes que aparecen en pantalla:

Corregir dirección	Corregir dirección hacia la izquierda de forma paralela	Corregir dirección hacia la derecha de forma paralela
		

Avance	Retroceda	Detenerse
		

Vista de la interfaz grafica:



- 5) Pruebas y validación: Evaluación del sistema en entornos reales para medir precisión y eficiencia.

Obtuvimos una precisión medida de $\pm 0,7\text{cm}$ lo cual cumple en su totalidad con lo esperado para este desarrollo y dimensiones de un automóvil.

- 6) Calibración: Se realiza por medio de la interfaz grafica donde el usuario ingresa la distancia ideal entre la cámara y el vehículo a la hora de estacionar, en base a dicho valor el sistema luego lo toma como referencia para las alarmas descriptas en el punto 4.

Se envía el parámetro y se almacena en un archivo tipo json

7. Herramientas de software y equipos utilizados

- Python 3: Lenguaje de programación principal.
- Librerías: OpenCV, NumPy, PyTesseract, Pyttsx3.
- Computadora con cámara: Para capturar imágenes y procesar datos. Luego se lo busca integrar con una cámara externa (Dahua DH-HAC-T1A21N)
- Raspberry Pi. Se busca implementar el código y funcionamiento del programa en este dispositivo e independizarse de una computadora tradicional.
- Espacio de estacionamiento controlado: Para pruebas y validación.

8. Resultados esperados

- Implementación de un sistema funcional de asistencia para estacionamientos.
- Reducción de incidentes relacionados con maniobras imprecisas.
- Mejor experiencia para el conductor mediante guías claras y efectivas.

9. Conclusión

El presente proyecto demuestra su potencial para mejorar la seguridad y eficiencia en los estacionamientos mediante la aplicación de tecnologías avanzadas de visión por computadora. Al ofrecer asistencia en tiempo real durante el estacionamiento manual, aborda eficazmente un problema cotidiano, proporcionando una solución práctica y escalable.

Los resultados obtenidos cumplen con los objetivos planteados, garantizando la precisión requerida para un espacio de estacionamiento con dimensiones compatibles con un automóvil típico. Asimismo, el diseño del sistema permite la implementación de cámaras adicionales para replicar la funcionalidad en múltiples posiciones de estacionamiento, lo que representa un avance significativo para su integración en garajes concurrenciosos.

Durante el desarrollo, se adquirieron conocimientos clave sobre el procesamiento de señales de video y la extracción de información relevante para su conversión en datos útiles, esenciales para los cálculos necesarios en el sistema.

Sin embargo, se identificaron limitaciones al intentar integrar el proyecto en una Raspberry Pi. El sistema presentó una baja tasa de fotogramas al utilizar una cámara externa conectada a través de un conversor analógico-digital. Tras revisar la configuración del dispositivo, se concluyó que el problema probablemente reside en un conflicto de controladores entre la Raspberry Pi y el conversor utilizado. Es importante señalar que este inconveniente no se presentó al emplear una notebook con Sistema operativo Linux, donde el programa operó de manera completamente funcional.

Para resolver este desafío, se recomienda explorar alternativas, como el uso de cámaras diseñadas específicamente para la Raspberry Pi y conectadas a través de su puerto dedicado, lo que evitaría el conflicto identificado. A pesar de esta limitación, el proyecto demuestra ser una solución viable y escalable, sentando una base sólida para futuras mejoras e implementaciones.

10. Referencias

- Documentación de Python general, OpenCV y NumPy.
- Estudios sobre aplicaciones de visión por computadora en movilidad urbana.
- Patentes utilizadas actualmente en Argentina.