

|  |  |
| --- | --- |
| ROBÓTICA  **TPL-1**  Visión Artificial | 4 de Mayo  2021 |

Profesor: Ing. Salariato, Ignacio

Jefe de Laboratorio: Ing. Suenaga, Gustavo

Alumnos: Napolitano, Gabriel

Comisión: Martes, Noche

|  |  |
| --- | --- |
| **LABORATORIO DE ELECTRÓNICA** | **ASIGNATURA: ROBÓTICA** |

|  |  |  |
| --- | --- | --- |
| **T.P.Nº 1** | **TÍTULO DEL TRABAJO PRÁCTICO:**  **Utilización de Python y OpenCV para Visión  Artificial** | **Código de Asignatura:**  **I0510** |
| **Horas de formación práctica asignada: 4** |

**OBJETIVOS:** Poner en práctica los conocimientos vistos en la teoría de visión artificial mediante distintos programas con distintas funcionalidades.

**OBJETIVO GENERAL:** Realizar programas que permitan desarrollar una cierta lógica o inteligencia en base al análisis de imágenes capturadas.

**OBJETIVOS ESPECÍFICOS:** Utilizar la visión artificial para el reconomiento y diferenciamiento de objetos y patrones.

**MARCO TEÓRICO Y CONOCIMIENTOS PREVIOS**

OpenCV es una librería de computación visual creada por Intel, esta librería está disponible para múltiples plataformas como: Windows, Linux, Mac, Android, además cuenta con soporte para diferentes lenguajes como: Python, Java, C/C++, Net.

La biblioteca OpenCV puede ser usado bajo licencia BSD para proyectos escolares o comerciales, las aplicaciones de esta librería incluyen, la robótica, análisis y procesamiento de imágenes o vídeos, seguimiento y detección de objetos, detección y reconocimiento de rostros, reconocimiento de placas de vehículos, análisis de formas, reconstrucción 3D, realidad aumentada

Dentro de las funciones que se puede realizar se puede modificar a nivel de pixel, para una imagen a color obtendremos los canales azul, verde y rojo, si la imagen es a escala de grises solo tendremos un canal, el cual indica la intensidad de gris de dicho pixel, se puede seleccionar una región de interés y trabajar solo sobre ella.

Podemos dibujar diversas figuras de una menara rápida y sencilla, como círculos, rectángulos.

Uno de los métodos más conocidos para la detección de esquinas es el llamado Harris Corner Detector el cual busca los cambios de intensidad producidos en cada pixel para una región de un tamaño definido, en OpenCV este método esta implementado por la fución cv2.cornerHarris().

La umbralización y umbralización adaptativa son unos de los métodos más simples y eficientes de segmentación, segmentar quiere decir, separar o extraer las regiones de una imagen que nos interese estudiar o analizar, para lograr separar la región deseada se establece un valor que define el umbral, los pixeles cuya intensidad superen el umbral serán rechazado o aceptados, según sea el caso.

Un histograma es una gráfica en donde se muestra la frecuencia con las que aparecen los distintos niveles de intensidad de una imagen a escala de grises, normalmente el nivel de intensidad está en el rango de 0 a 255, en donde el valor 0 representa los color negro y 255 el color blanco, utilizando el histograma de una imagen podemos modificar sus características, por ejemplo, el brillo y contraste.

Un contorno es un conjunto de puntos que conectados unos con otros de manera consecutiva que forman una figura que rodea un objeto determinado, la detección de contornos en OpenCV se aplica sobre imágenes binarias, la figura obtenida puede ser analizada posteriormente para determinar cuál es el objeto que hemos detectado, para obtener los contornos utilizaremos la función findContours(), mientras que drawContours() nos ayudará a dibujarlos.

**EQUIPOS Y DIAGRAMA**

|  |  |  |
| --- | --- | --- |
| **Equipo requerido** | **Versión (utilizada en desarrollo)** | **Observaciones** |
| Python | 3.9.4 | Intérprete del lenguaje. |
| OpenCV | 4.5.2 | Módulo externo (debe descargarse) |
| Numpy | 1.20.2 | Módulo externo (debe descargarse) |
| Matplotlib | 3.4.1 | Módulo externo (debe descargarse) |

El código fuente de toda la práctica, junto con los archivos necesarios para su ejecución pueden descargarse de:

**PROCEDIMIENTO**

Modificar un Pixel:

Tomar una Foto con la cámara.

Obtener el valor de los canales BGR del Pixel Central.

Modificar dicho Pixel para que se visualice en color Rojo.

Seleccionar una Región de Interés ROI de 100 pixeles en el eje x por 50 pixeles en el eje y donde el pixel modificado quede centrado.

Detección de esquinas:

Dada una imagen de una figura geométrica detectar los vértices de dicha figura mediante el método de Harris Corner y marcarlos en la imagen original.

Umbralización y Umbralización adaptativa:

Realizar la umbralización de la imagen ¨Herramientas¨ cargando valores de umbral y máximo en forma manual para que las herramientas queden negras usando la función “cv2.THRESH\_BINARY”.

Realizar la misma operación cargando el valor aportado por “cv2.THRESH\_TRIANGLE”, que calcula el valor óptimo de umbralización.

Realizar la Umbralización adaptativa de la imagen “Herramientas”.

Histograma:

Calcular la frecuencia con la que aparecen los niveles de intensidad Azul, Verde y Rojo de la imagen tomada en el punto 1.

Seguimientos de Objetos:

Seguir un objeto de color verde con la cámara con un enmarcado que delimite el objeto a medida que se está moviendo.

Busqueda patrones Match Template:

Tomar una foto con la cámara, seleccionar un objeto a identificar en la imagen, hacer un recorte y guardar como patrón. Identificar ese patrón en la imagen original recuadrándola e indicando su posición. El patrón debe ser encontrada en una posición definida por el desarrollador. Si se carga una nueva imagen con el mismo patrón pero en otra posición debe avisarnos patrón desfasado.

Seguimiento de líneas (line detection)

En una foto o video mp4 en donde se vea una línea (para seguimiento de robots o de autos), realizar el seguimiento de líneas con la función cv2.HoughLines(), como lo haría un robot móvil.

**TABLAS Y GRAFICOS**

Imágenes originales y modificadas con sus respectivos nombres.

**ECUACIONES Y CALCULOS**

Adjuntar código de las funciones utilizadas para lograr cada manipulación.

**APLICACIÓN DE LAS NORMAS DE SEGURIDAD EN LA PRESENTE PRÁCTICA.**

Según reglamentación del Laboratorio.

**INFORME SOBRE LAS CONCLUSIONES Y OBSERVACIONES DE LA PRÁCTICA**

Se indican las conclusiones de la práctica agregando referencias sobre el tema.

**Desarrollo**

1. **Introducción**
2. **Objeto**
3. **Alcance**
4. **Desarrollo (Redactar lo realizado indicando el porqué de las mediciones realizadas)**
5. **Resultados obtenidos (para cada caso estudiado resumir los resultados obtenidos en tablas, gráficos y diagramas)**
6. **Conclusiones y recomendaciones.**

**ANEXO I: Esquema del informe**

**El informe del trabajo práctico debe contener como mínimo esta información:**

1. **Introducción**
2. **Objeto**
3. **Alcance**
4. **Desarrollo (Redactar lo realizado indicando el porqué de las mediciones realizadas)**
5. **Resultados obtenidos (para cada caso estudiado resumir los resultados obtenidos en tablas, gráficos y diagramas)**
6. **Conclusiones y recomendaciones.**
7. **Referencias**

**ANEXO II: Criterios de Evaluación**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Materia:* Robótica** | | | |
| ***Criterio de Aprobación*** | **SÍ** | **NO** | ***Observación*** |
| Trabajo original sin similitudes con otros. |  |  |  |
| La introducción, objeto y alcance del enunciado son diferentes a las del enunciado del TP. |  |  |  |
| Redacción es correcta. Debe ser entendido como un informe técnico a presentar a un tercero con conocimientos técnicos que no haya realizado las  experiencias. |  |  |  |
| Está desarrollado el informe según lo indicado en el Anexo: Esquema del Informe. |  |  |  |
| Todas las preguntas están claramente respondidas. |  |  |  |
| Definir cada una de las variables. |  |  |  |
| Las unidades empleadas deben ser consistentes a lo largo de todo el informe. |  |  |  |
| Deben expresarse conclusiones y deben ser consistentes con los resultados obtenidos. |  |  |  |
| Las citas a textos deben quedar reflejadas en las referencias. |  |  |  |
| Las diversas entregas de los TP con las correcciones solicitadas deben ser acompañadas de los originales corregidos. |  |  |  |