**PRACTICA 4 DETECCION DE ESQUINAS**

**Visión computacional Aplicada a la Robótica**

**Kelly Del Moral Adame**

**Introducción**

La presente práctica se redacta con la finalidad de demostrar el funcionamiento de un detector de esquinas. Se programó el algoritmo de detector de esquinas de Harris Corner con la finalidad de comprender cada paso del algoritmo y así poder modificar los parámetros de este. Se realizaron pruebas en una imagen y posteriormente en video y se comprobó el funcionamiento de la detección de esquinas.

**Objetivo**

Aplicar el detector de bordes de Canny de la practica 02

Aplicar el concepto de gradiente y matriz de segundo momento para detectar esquinas

Aplicar el detector de esquinas de Harris sobre un video.

**Marco teórico**

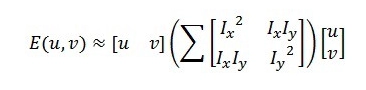
El algoritmo de **detección Harris Corner** fue desarrollado para identificar las esquinas internas de una imagen. Las esquinas de una imagen se identifican básicamente como las regiones en las que hay variaciones en la gran intensidad del gradiente en todas las dimensiones y direcciones posibles.

La detección de esquinas de Harris hace uso de este concepto fundamental. La idea es localizar puntos de interés desde donde hay bordes que se extienden en más de una dirección. El algoritmo funciona calculando la diferencia de intensidad para un desplazamiento en todas las direcciones. La expresión para la cual viene dada por

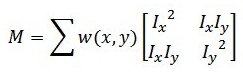
Imagen que contiene Texto

Descripción generada automáticamente

La función de ventana podría ser una ventana rectangular o podría ser una ventana gaussiana que da pesos a píxeles en (x,y). Aplicando la expansión de Taylor lo conseguimos.



Denotando la matriz M



Ix, Iy son las derivadas de imagen en las direcciones x e y.

La respuesta de Harris R viene dada por:



K es el factor de sensibilidad para diferenciar la esquina de las aristas. Los valores pequeños de k dan como resultado la detección de esquinas afiladas [1].

**Algoritmo**

Convierta la imagen en color a escala de grises.

Calcule las derivadas espaciales, es decir, IX (derivada horizontal) e Iy (derivada vertical).

Estructure la configuración del tensor (IxIx, IyIy e IxIy).

Calcule la respuesta de Harris(R).

Supresión de no máximos

**Desarrollo**

**1.Conversion de la imagen a escala de grises**

Primero se realizó la reducción de ruido, por medio del filtro gaussiano. Este primer paso se hace debido a que el ruido en una imagen puede ocasionar la detección de falsos bordes, por eso, es necesario aplicar un filtro para reducir el ruido como primera etapa. En la fig. 1 se puede observar la imagen original y en la fig. 2 se observa el resultado del suavizado.

Código QR

Descripción generada automáticamenteCódigo QR

Descripción generada automáticamente

Fig. 2. Imagen filtrada

Fig. 1. Imagen original

**2. Obtención de las derivadas parciales Ix e Iy**

Una vez filtrada la imagen por medio del filtro gaussiano se obtiene el gradiente de intensidad, aplica un filtro sobel en ambas direcciones vertical y horizontal para poder obtener las derivadas parciales de la imagen tanto en x como en y.

Gráfico, Texto

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

Fig. 3 Imagen con sobel en x

Fig. 4 Imagen con sobel en y

**3. Obtención de la matriz de segundo momento empleando una ventana de tamaño W**

Una vez obtenidas las derivadas parciales se forma la matriz de segundo momento por medio de la siguiente ecuación.

Texto

Descripción generada automáticamente

Ix e Iy son las derivadas parciales, w es el tamaño de la ventana. El tamaño de la ventana es arbitrario en el caso particular de esta particular se realizó una ventana de 2\*2.

**4. Cálculo de la respuesta de Harris**

Se calcula la respuesta de Harris con la siguiente ecuación

Texto, Carta

Descripción generada automáticamente

λ 1 y λ2 son los valores propios de M. Por lo tanto, los valores de estos valores propios deciden si una región es una esquina, un borde o un plano.

**5. Supresión de no máximos**

**Posteriormente se revisa el valor de R , para esto se establece un umbral y si pasa ese umbral entonces se detecta la esquina.**

Cuando | R| es pequeño, lo que sucede cuando λ 1 y λ2 son pequeños, la región es plana.

Cuando R<0, que ocurre cuando λ 1>>λ2 o viceversa, la región es una arista.

Cuando R es grande, lo que sucede cuando λ 1 y λ 2 son grandes y λ 1∼λ2, la región es una esquina.

En las fig. 5 se puede observar el resultado final de aplicar el algoritmo con un k = 0.04, la fig. 6 es el resultado de aplicar el algoritmo con k=0.5, en este caso la fig. 6 se puede ver que se detectaron todas las esquinas de mejor manera, debido a que k influye en la detección de esquinas .

Fig 9. Resultado final

Código QR

Descripción generada automáticamenteCódigo QR

Descripción generada automáticamente

Fig. 6 Imagen final con K=0.5

Fig. 5 Imagen final con K=0.04

**Conclusiones**

La mayor ventaja de Harris Corner Detector es que es invariante a factores como la rotación, la traslación y los cambios en la iluminación, lo que lo convierte en la primera opción para la recuperación de datos de imagen .Es importante escoger el valor adecuado de la K para poder detectar las esquinas ya que la mala elección del mismo nos puede afectar en la detección de las esquinas.

**Referencias**

[1] Open Source Computer Vision. [Detección de esquinas de Harris](https://docs.opencv.org/3.4/dc/d0d/tutorial_py_features_harris.html). Disponible en línea: https://docs.opencv.org/3.4/dc/d0d/tutorial\_py\_features\_harris.html

[2] V. Vasco, A. Glover and C. Bartolozzi, "Fast event-based Harris corner detection exploiting the advantages of event-driven cameras," 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea (South), 2016, pp. 4144-4149, doi: 10.1109/IROS.2016.7759610.