

Práctica 4

Detector de esquinas de Harris

Visión Computacional Aplicada a la Robótica
UNAM, 2023-2

Francisco Arturo Meza Torres
franciscomeza1918@gmail.com

Resumen

En esta práctica se implementa el detector de esquinas de Harris sobre una imagen binaria de prueba y sobre un flujo de vídeo, que han sido procesados con el detector de bordes de Canny, para esto se da una introducción al tema y se mencionan los pasos para la implementación del algoritmo en Python. Se realizan pruebas variando los parámetros de entrada al detector de esquinas, se analizan los resultados y se concluye.

1. Introducción

La visión por computadora se utiliza para comprender el mundo tridimensional, en el que las escenas visualizadas contendrán una amplia variedad de objetos que se buscan reconocer [1], para esto existe una vasta teoría y algoritmos. Esta práctica se enfoca en el detector de esquinas de Harris, también conocido como de Harris y Stephens. El algoritmo es un descriptor local para búsqueda de puntos y regiones correspondiente entre imágenes, es de gran utilidad en aplicaciones de creación de panoramas, realidad aumentada y la computación gráfica [2].

2. Objetivos

- Aplicar el detector de bordes de Canny de la práctica 02.
- Aplicar el concepto de gradiente y matriz de segundo momento para detector de esquinas.
- Aplicar el detector de esquinas de Harris en un flujo de video.

3. Hipótesis

En las pruebas con el detector de esquinas de Harris se espera que, al variar el umbral para la supresión de no máximo, con valores altos solo tomará las esquinas más prominentes, mientras que con valores pequeños mantendrá la mayor cantidad de esquinas encontradas. Por otra parte, al variar el parámetro de la respuesta de Harris, con valores altos detectara la mayor cantidad de esquinas, mientras que para valores pequeños solo detectara las esquinas más prominentes.

4. Marco teórico

La idea principal del detector de esquinas, es localizar puntos de interés en donde la vecindad circundante muestra bordes en más de una dirección, lo que resultan ser esquinas en la imagen. Si se desea conocer más del tema, consultar [2]. En este trabajo solo se abordará de forma general los pasos del detector de esquinas.

Para obtener la autocorrelación de los valores del gradiente en una vecindad de un pixel se obtiene la siguiente matriz M_I

$$M_I = \nabla I \nabla I^T = \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x & I_y \end{bmatrix} \quad (1.1)$$

$$M_I = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (1.2)$$

Esta matriz es simétrica positiva semidefinida y tiene valores propios en $\lambda = |\nabla I|^2$ y $\lambda = 0$. Ahora bien, para que los valores propios cambien en función de las propiedades locales de la imagen, se

promedia M_I sobre los píxeles vecinos, esto se logra convolucionando con una matriz de pesos, que usualmente es un filtro gaussiano.

$$\overline{M} = W * M_I \quad (1.3)$$

La matriz \overline{M} es conocida como matriz de Harris o de segundo momento. Dependiendo de los valores del gradiente ∇I se tendrán 3 casos de los valores propios de \overline{M} .

- Si λ_1 y λ_2 son valores positivos grandes, hay una esquina en x.
- Si λ_1 es grande y λ_2 es cercano a cero, hay un borde en x.
- Si λ_1 y λ_2 son cercanos a cero, entonces no hay nada en x.

5. Desarrollo

El detector de esquinas consta de los siguientes pasos para un flujo de video.

1. Se identifican los bordes con el detector de bordes de Canny.



Figura 1. Detector de Bordes de Kanny.

2. Se obtiene la derivada en x y en y con un filtro Sobel.

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

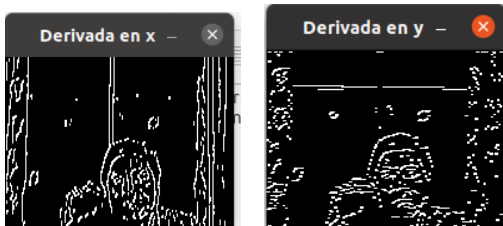


Figura 2. Filtro sobel en x y en y.

3. Se obtiene la matriz de segundo momento.



Figura 3. Matriz de segundo momento.

4. Se obtienen los valores propios y la respuesta de Harris.



Figura 4. Respuesta de Harris.

5. Se realiza la supresión de no máximos.



Figura 5. Supresión de no máximos.

6. Por último se muestran las esquinas encontradas sobre la imagen original.

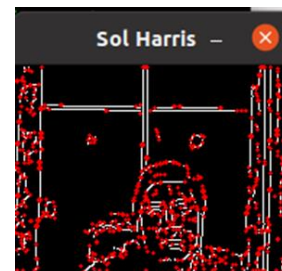


Figura 6. Detector de esquinas de Harris en un flujo de video.

Para comprobar que el detector funciona correctamente, se aplicó a una imagen con solo líneas.

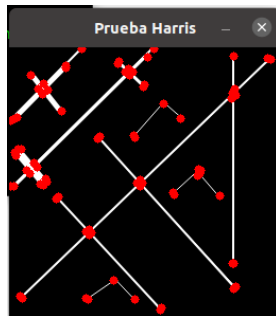


Figura 7. Detector de esquinas de Harris en una imagen.

6. Análisis de resultados

Se realizaron pruebas variando el parámetro de ajuste de la respuesta de Harris (k) y el parámetro de ajuste de los límites para la matriz de segundo momento. Se propusieron los valores mostrados en las tablas 1 y 2, obteniendo los siguientes resultados.

Casos 1.

Tabla 1. Parámetro k variable y parámetro w fijo.

w	3	3	3	3
k	0	150	230	249

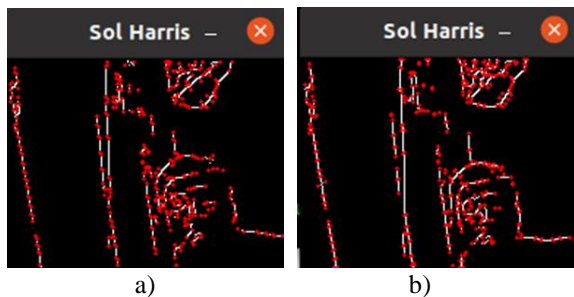


Figura 8. Detector de esquinas de Harris con $w = 3$ y con a) $k = 0$ y b) $k = 150$.



Figura 9. Detector de esquinas de Harris con $w = 230$ y con a) $k = 0$ y b) $k = 249$.

Caso 2.

Tabla 2. Parámetro k fijo y parámetro w variable.

w	0	3	9	12
k	50	50	50	50

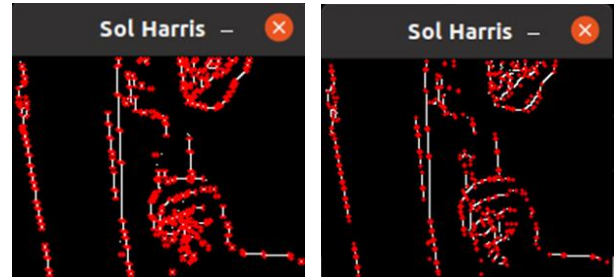


Figura 10. Detector de esquinas de Harris con $k = 50$ y con a) $w = 0$ y b) $w = 3$.

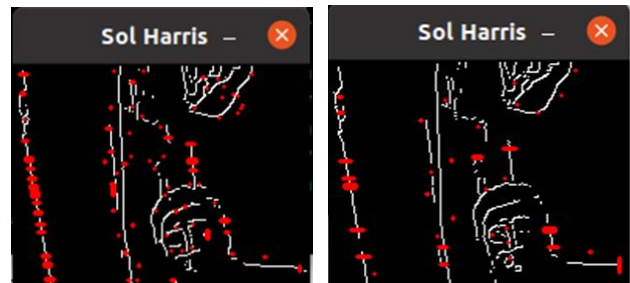


Figura 11. Detector de esquinas de Harris con $k = 50$ y con a) $w = 9$ y b) $w = 12$.

Con el análisis realizado se observó que al disminuir el parámetro k se consideran más puntos, por otra parte, al incrementar k se tienen menos puntos y se colocan puntos de un lado y del otro de una intersección, lo cual ayuda a identificar mejores las esquinas, el mayor cambio se tuvo en el intervalo de 230 a 250. En el caso del parámetro w al disminuirlo se identifican más puntos y al incrementarlo, se idéntica menos puntos y se acomoda los puntos en una pequeña línea, los cambio en w se mantuvieron a lo largo de todo el intervalo.

7. Conclusión

Con el filtro detector de esquinas de Harris se logró identificar las esquinas que se encuentran en una imagen, así como en un flujo de video. Al variar los parámetros k y w se logró observar que se puede sintonizar el filtro, para obtener un mejor resultado en la identificación de esquinas, cabe mencionar que también fue de utilidad variar los parámetros del filtro de Canny. Realizar una serie de pasos para la implementación del filtro, que contemplo desde el filtro de Canny y los gradientes de la derivada en x y

en y, hasta la solución de Harris, fue de gran utilidad para ir identificado y modificado cada parte del algoritmo, en conclusión, con la implementación del algoritmo de Harris y con el análisis realizado, se logró cumplir con los objetivos e hipótesis de la práctica, con excepción de la suposición del parámetro k , mostrando un comportamiento contrario.

6. Referencias

[1] C. Harris and M. Stephens. A Combined Corner and Edge Detector. In C. J. Taylor, editors, Proceedings of the Alvey Vision Conference, pages 23.1-23.6. Alvey Vision Club, September 1988. doi:10.5244/C.2.23

[2] Solem, J. E. (2012). Programming Computer Vision with Python: Tools and algorithms for analyzing images. " O'Reilly Media, Inc.".