

Práctica Lucas-Kanade y Horn&Schunck

Sergio Domínguez Rodríguez

8 Marzo 2019

Índice

1. Lukas-Kanade	1
1.1. Análisis de tiempo LK	1
1.2. Observaciones LK	2
2. HornSchunck	3
2.1. Análisis de tiempo HS	4
2.2. Observaciones HS	5
3. Conclusiones	6
3.1. Resumen LK	6
3.2. Resumen HS	6
4. Enlace proyecto GitHub	7

1. Lukas-Kanade

El método de Lukas-Kanade lo hemos probado para 3 tipos de vídeo, dos bajados de internet y uno 'doméstico' grabando el movimiento de una pelota con el móvil.(Figura: 1)



((a)) Vídeo A



((b)) Vídeo B



((c)) Vídeo C

Figura 1: Muestra de los tres vídeos

1.1. Análisis de tiempo LK

Respecto al análisis de tiempo del algoritmo, apreciamos que cuanto mayor es el rango de las vecindades locales, mayor es el tiempo que tarda el algoritmo. Esperábamos apreciar una mayor diferencia de tiempos a la hora de calcular kernels parecidos, es decir, entre kernels de 3, 5 o 7 de dimensión. Sin embargo para este tipo de kernels tan próximos no apreciamos prácticamente diferencia de tiempos, nos tenemos que ir a una diferencia de kernels de mayor tamaño (10, 30, 50) para que notar una diferencia significativa en tiempos.

Esto creemos que es debido a que la mayor parte del tiempo que consume el algoritmo es en la interacción por los diferentes píxeles de la imagen en vez de las operaciones que tiene que hacer en cada una de las regiones. En la siguientes imágenes se puede apreciar que para unos valores de vecindad consecutivos no se aprecia una diferencia de tiempos significativa, sin embargo, como hemos explicado anteriormente, si que hay diferencia de tiempo cuando comparamos vecindades con bastante diferencia entre ellas.(Figura: 2)

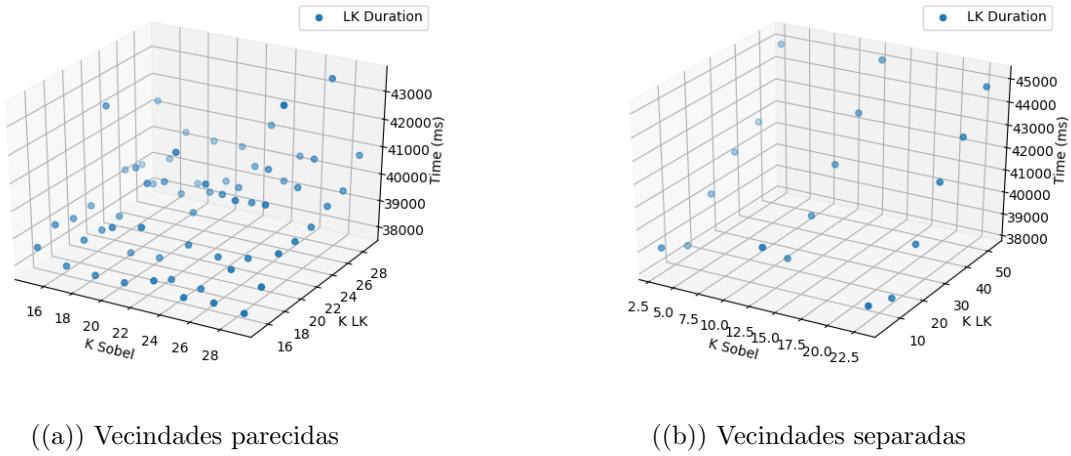


Figura 2: Diferencia entre vecindades LK

Por otro lado, respecto al parámetro que define el tamaño de la vecindad a la hora de calcular el filtro sobel, no encontramos diferencias en cuanto al tiempo, se puede apreciar también en la figura (Figura: 2). La coordenada correspondiente al eje de 'K Sobel' no afecta a la hora de incrementar el tiempo.

1.2. Observaciones LK

Tras aplicar el algoritmo y proceder a pintarlo, nos dimos cuenta que había algunos píxeles que eran muy dispares respecto a la mayoría, incluso, varios órdenes de dimensión mayor o menor. Por este motivo, tras realizar el algoritmo, decidimos pasar un filtro gaussiano, a la salida tanto de la U como de la V, para así suavizar y evitar tener estos 'outliers' que luego nos afectaban a la hora de pintar correctamente el flujo óptico.

Para poder estudiar bien el algoritmo y ver las diferentes salidas que aporta al variar el kernel de vecindades de Lucas-Kanade así como el kernel a la hora de calcular los filtros de borde vertical y horizontal hemos realizado una batería de pruebas para los tres vídeos anteriormente citados.

Tras estas pruebas hemos observado que el método de Lucas-Kanade es muy dependiente del tamaño de la vecindad del kernel que elijas. En función de lo rápido o despacio que sea el movimiento entre las dos imágenes a comparar. Si tenemos un vídeo en el que el movimiento es muy alto, tendremos que aplicar un tamaño de vecindad de kernel mayor para poder capturar todo el movimiento dentro de ese kernel. Si usamos una vecindad más pequeña lo que conseguiremos son muchos puntos sin ninguna forma o ninguna forma clara lo que no nos servirá de mucho a la hora de calcular el flujo óptico.

Ejemplo para el vídeo A y Vídeo B en las figuras (Figura: 3) y (Figura: 4) respectivamente.

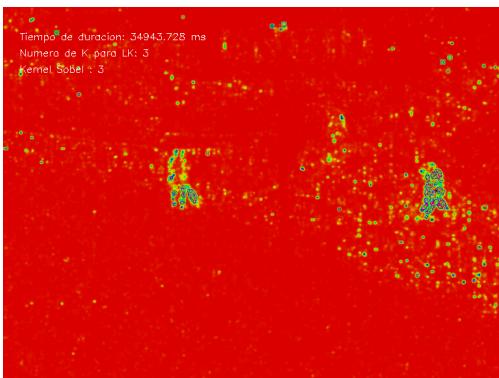


((a)) Vídeo A



((b)) Vídeo B

Figura 3: Muestra de vecindad grande y pequeña para el vídeo A



((a)) Vídeo A



((b)) Vídeo B

Figura 4: Muestra de vecindad grande y pequeña para el vídeo B

Se puede ver en las imágenes que para cada vídeo el tamaño de la vecindad es muy importante, siendo siempre una mejor opción para este método una vecindad grande. Con una vecindad grande podemos conseguir un área más difusa pero a su vez bien definida respecto al resto de la imagen.

Por último mostramos unas secuencias del flujo óptico adquirido en el vídeo que se ha realizado con la pelota. En estas imágenes se puede apreciar que, en este caso, al estar la pelota muy cerca de la cámara, podemos calcular el flujo óptico con mucha mayor fiabilidad incluso con un tamaño de vecindad muy pequeño.(Figura: 5)

2. HornSchunck

El método de HornSchunck también lo hemos probado para 3 tipos de vídeo, dos bajados de internet y uno 'doméstico' grabando el movimiento de una pelota con el móvil.(Figura: 1)

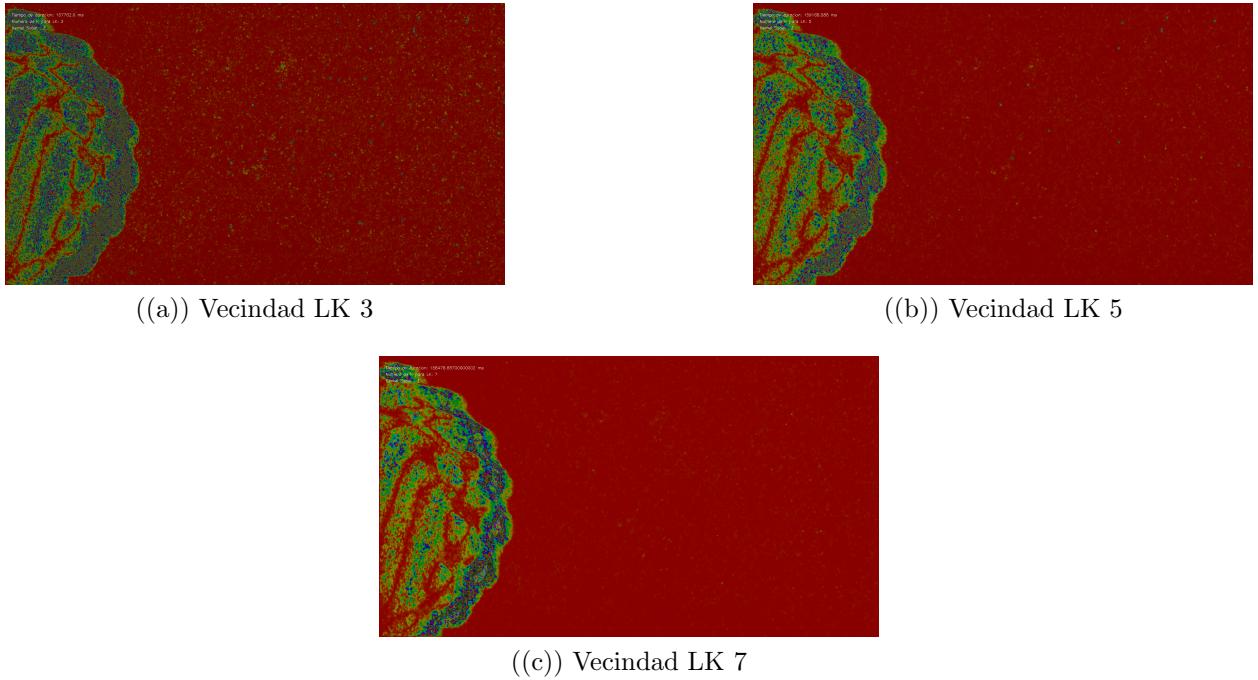


Figura 5: Muestra del vídeo C: Pelota

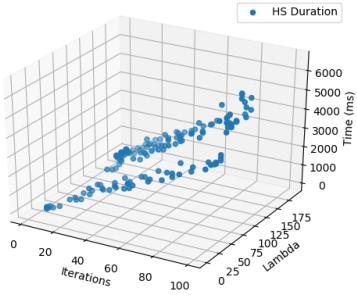
2.1. Análisis de tiempo HS

Hemos notado notables mejoras en cuanto a tiempo de procesado y calidad de adquisición del flujo óptico respecto al algoritmo de Lucas-Kanade. El motivo de que esto sea así, en gran parte, es debido a que el propio algoritmo va analizando la convergencia que tiene en las componentes U y V respecto a la interacción anterior y en el caso de que ya no mejore demasiado para el proceso.

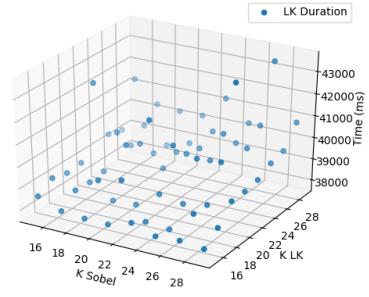
En cuanto al tiempo podemos observar con las siguientes gráficas la diferencia que hay entre los dos algoritmos para una misma imagen (Figura: 6). Hemos comprobado que la convergencia de este algoritmo es más eficaz cuanto más pequeña es la imagen y más grande es el valor de lambda, por lo que para estos casos el algoritmo es más rápido.

Respecto al tamaño de la imagen, esto tiene sentido debido a que es más fácil que la diferencia de la media de la imagen entre un instante y otro converja más fácil cuanto menos píxeles tiene.

Respecto al valor de lambda, hemos apreciado, aunque levemente, que a mayor valor de lambda el fondo es más homogéneo, tenemos menos ruido en la imagen como se puede ver en las siguientes figuras (hemos forzado los valores de lambda por encima de 60 para que se aprecie la diferencia) por lo que acaba convergiendo antes y siendo más rápido.(Figura: 9(c)):

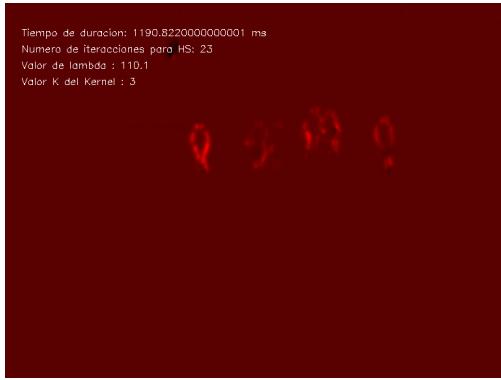


((a)) Tiempos para HS

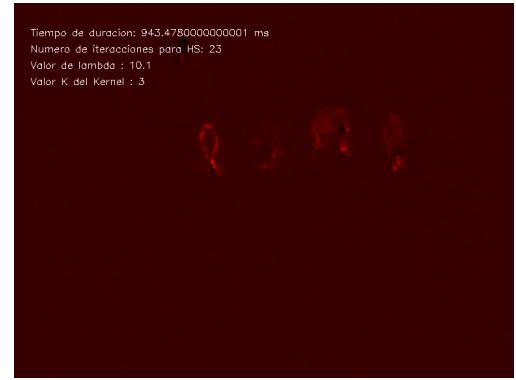


((b)) Tiempos para LK

Figura 6: Diferencia de tiempos entre HS y LK



((a)) Lambda 110



((b)) Lambda 10

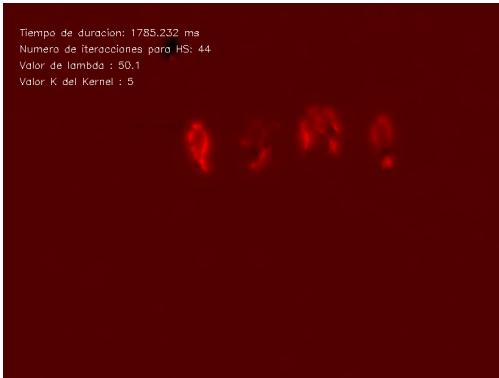
Figura 7: Diferencia entre valores de lambda

2.2. Observaciones HS

También hemos visto que el ajuste de los parámetros necesarios para obtener un buen resultado del flujo óptico son bastante más simples que para Lucas-Kanade, dado que no afectan tanto al resultado final del algoritmo.

Aquí podemos ver un ejemplo de como afecta el tamaño de la vecindad del kernel que hace la media para un mismo valor de lambda (Figura: 8). Únicamente apreciamos una ligera dispersión o difusión alrededor de los contornos de las figuras, pero el flujo óptico se aprecia perfectamente en ambos casos.

Por último dejamos una secuencia de 5s del vídeo A en la que calculamos el flujo óptico frame a frame Link al vídeo



((a)) Vecindad 5



((b)) Vecindad 11

Figura 8: Diferencia entre vecindades del kernel que hace la media

3. Conclusiones

Tras evaluar los dos algoritmos nos quedamos con que el algoritmo de HS respecto al de LK es bastante mejor por varios motivos:

- En cuanto a velocidad de cómputo es bastante más rápido.
- En cuanto a calidad a la hora de calcular el flujo óptico es más preciso, al menos para las secuencias estudiadas en esta práctica.
- En cuanto a la flexibilidad, sin ser el algoritmo más difícil de implementar, vemos una gran ventaja en que se utilice el método de convergencia dado que esto permite ahorra bastante tiempo
- Y por último vemos más fácil de ajustar para diferentes tipos de secuencias el algoritmo de HS respecto al de LK.

Para concluir, en la siguiente figura podemos ver una muestra de los dos algoritmos, con sus tiempos para diferentes combinaciones así como una imagen de una de las mejores combinaciones de cada uno.

3.1. Resumen LK

En la (Figura: 9) se puede ver un resumen del algoritmo de LK.

3.2. Resumen HS

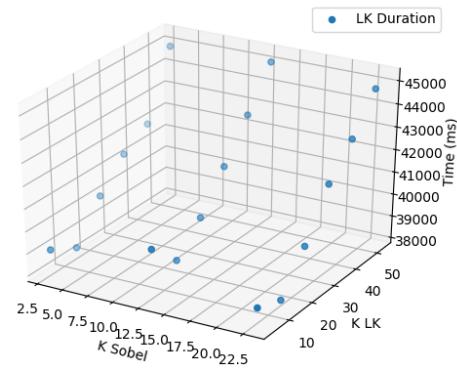
En la (Figura: 10) se puede ver un resumen del algoritmo de HS.

4. Enlace proyecto GitHub

Para mayor información sobre el código realizado y la totalidad de las imágenes y vídeos obtenidos para la evaluación de estos algoritmos se proporciona el enlace al proyecto, alojado en GitHub Proyecto GitHub.



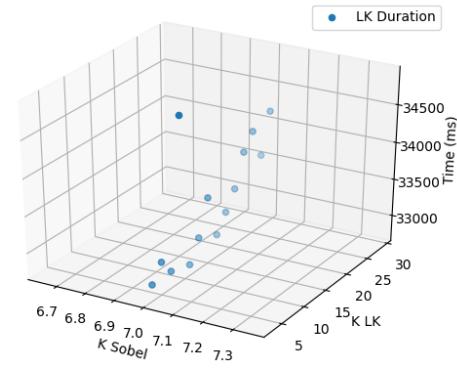
((a)) Vídeo A Flujo Óptico



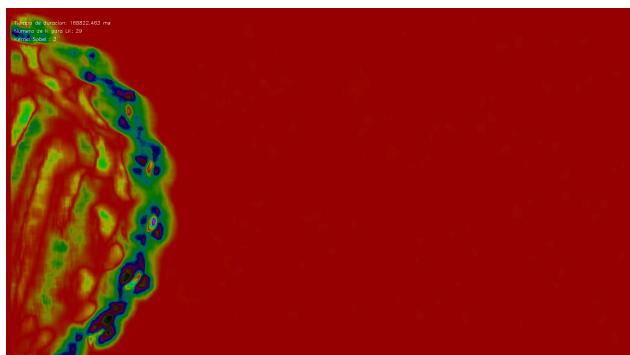
((b)) Vídeo A Tiempos



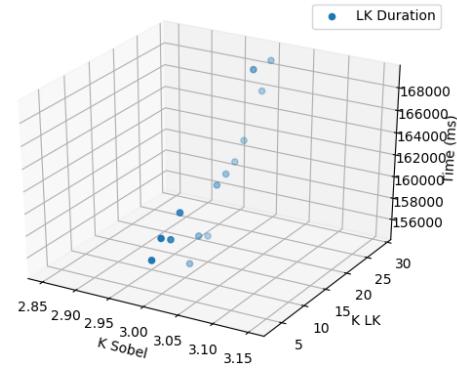
((c)) Vídeo B Flujo Óptico



((d)) Vídeo B Tiempos

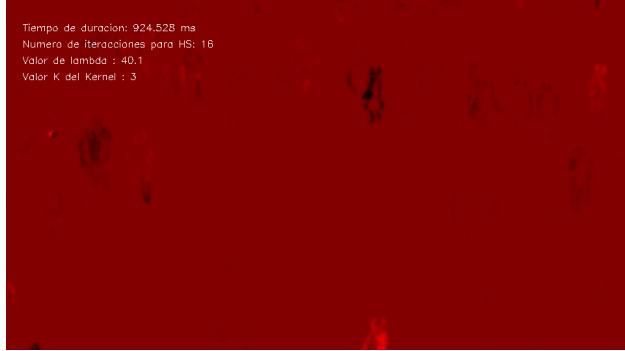


((e)) Vídeo C Flujo Óptico

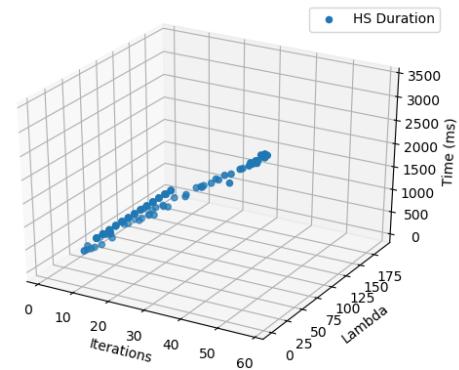


((f)) Vídeo C Tiempos

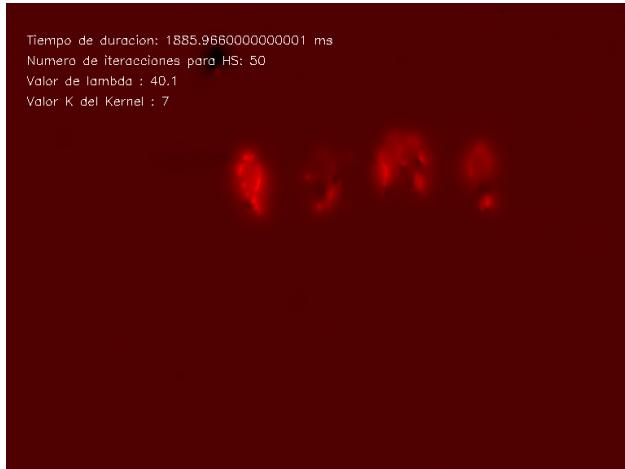
Figura 9: Resumen LK



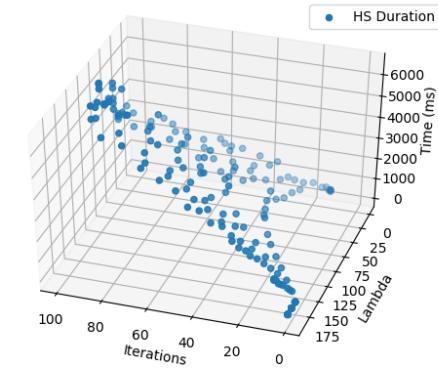
((a)) Vídeo A Flujo Óptico



((b)) Vídeo A Tiempos



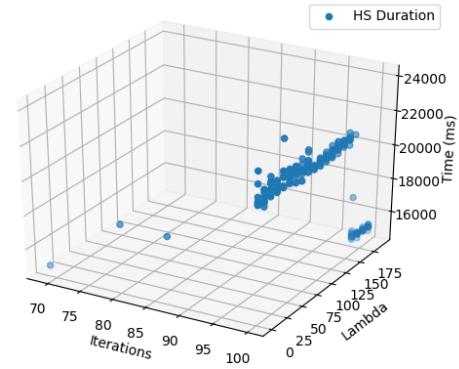
((c)) Vídeo B Flujo Óptico



((d)) Vídeo B Tiempos



((e)) Vídeo C Flujo Óptico



((f)) Vídeo C Tiempos

Figura 10: Resumen HS