



Universidad
Zaragoza

Trabajo Fin de Grado

Técnicas de tracking de puntos característicos
ante movimientos bruscos de cámara.

Autor:

Alejandro Gargallo Sureda

Director:

Daniel Gutiérrez Gómez

Ponente:

José Jesús Guerrero Campo

Escuela de Ingeniería y Arquitectura

2015



DECLARACIÓN DE
AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D^a. Alejandro Gargallo Sureda

con nº de DNI 73019891C en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) Grado _____, (Título del Trabajo)

Técnicas de tracking de puntos característicos ante movimientos bruscos de cámara.

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, a 7 de septiembre 2015 _____

Fdo: Alejandro Gargallo Sureda

Técnicas de tracking de puntos característicos ante movimientos bruscos de cámara

RESUMEN

Cada vez está más extendido el uso de cámaras portables o llevables, conocidas comúnmente por el anglicismo wearables, en actividades deportivas. Durante este tipo de actividades son frecuentes los movimientos bruscos y rápidos que dificultan el tracking (seguimiento) de puntos característicos del entorno en las imágenes, complicando en gran medida o llegando en ocasiones a impedir la aplicación de técnicas de SLAM¹ visual.

El objetivo de este trabajo es estudiar técnicas de tracking robustas ante movimientos bruscos de cámara que permitan la aplicación de odometría y mapeo en 3D ante este tipo de movimientos.

El trabajo se desarrolla sobre plataforma Linux, en lenguaje C++ sobre el editor de texto de programación Geany y haciendo uso de los algoritmos disponibles en la biblioteca libre de visión artificial desarrollada por Intel OpenCV[1], todo ello software libre.

Se procederá en primer lugar a crear un entorno de simulación y prueba que permita calcular el flujo óptico sobre imágenes reales grabadas con una cámara wearable. Se realizará un estudio del flujo óptico sobre imágenes andando e imágenes corriendo con el fin de definir el problema que generan los movimientos bruscos en el tracking, de ahí en adelante, se estudiarán formas de mejorar el seguimiento mediante dos técnicas: estudiando la frecuencia y la aplicación de pirámides de imágenes.

El desarrollo del trabajo requiere una fase previa de aprendizaje y complemento de formación en visión por computador dado que no es una disciplina sobre la que se imparte ninguna asignatura a lo largo del grado. Esta formación se obtiene mediante tutoriales, lecturas recomendadas y con la realización de un curso online de visión por computador impartido por la “Queensland University of Technology”, Brisbane, Australia, tutorizado por Peter Corke, utilizando como herramienta de aprendizaje y programación Matlab.

¹ del inglés Simultaneous Localization And Mapping.

Agradecimientos

Debo agradecer en primer lugar por la ayuda y el conocimiento prestado a Daniel Gutiérrez, director del presente proyecto, y a José Jesús Guerrero, ponente del mismo.

Me gustaría agradecer también de forma general a familiares, compañeros y docentes que de una forma u otra me han acompañado a lo largo de estos años, por la paciencia, conocimientos y experiencias compartidas.

Tabla de contenidos

Capitulo1.Introducción	1
1.1 Motivación	1
1.2 Justificación del proyecto	1
1.3 Objetivos	2
Capitulo2.Métodos de tracking.....	3
2.1 Template matching	3
2.2 Flujo óptico	4
2.2.1 Flujo óptico denso	5
2.2.2 Flujo óptico esparcido.....	6
2.3 Flujo óptico en pirámides	7
Capitulo3.Estudio del flujo óptico ante movimientos bruscos de cámara.....	9
3.1 Flujo óptico corriendo vs flujo óptico caminando	9
3.2 Neguentropía de flujo como herramienta de evaluación	10
Capitulo4. Realización de experimentos.....	14
4.1 Experimento I. Estudio de frecuencia.....	15
4.1.1 Test 1.....	15
4.1.2 Test 2.....	16
4.2 Experimento II: Pirámides de imágenes	18
Capitulo 5.Conclusiones.....	22
Anexo I. Descripción de las funciones de OpenCV utilizadas	24
Anexo II. Resultados completos experimento I.....	26
Test 1	26
Test 2	27
Anexo III. Resultados completos experimento II.....	28
Referencias	31

Índice de figuras

- Figura 2.1. Aplicación de Template Matching 4
- Figura 2.2. Ejemplo de flujo óptico denso. 6
- Figura 2.3. Ejemplo de flujo óptico esparcido..... 6
- Figura 2.4. Pirámide de imágenes..... 8

- Figura 3.1. Fotogramas seleccionados corriendo (a) y caminando (b) 9
- Figura 3.2. Flujo óptico caminando a la distancia de: 1 fotograma (a), 5 fotogramas (b) y 30 fotogramas (c) 10
- Figura 3.3. Flujo óptico corriendo a la distancia de: 1 fotograma (a), 5 fotogramas (b) y 30 fotogramas (c) 10
- Figura 3.4. Zonas de flujo óptico bien calculado, ordenado (a); zonas de flujo óptico mal calculado, desordenado (b)..... 11
- Figura 3.5. Ejemplo de cuadrícula proyectada sobre un fotograma 11
- Figura 3.6. Ejemplo de histograma de ángulos 12

- Figura 4.1. Esquema de secuencias aplicadas en el test 1 del experimento 1. 15
- Figura 4.2. Flujo óptico trazado para la misma sección de video: fotograma a fotograma (a), de 2 en 2 (b), de 5 en 5(c) y de 10 en 10 (d). 16
- Figura 4.3. Neguentropía mínima media para cada caso 16
- Figura 4.4. Flujo óptico cuando el corredor pasa por su eje vertical (a), cuando da un paso (b). 17
- Figura 4.5. Valores de neguentropía mínimos para cada fotograma. Picos resaltados en rojo. 17
- Figura 4.6. Neguentropía mínima media de las secuencias corriendo..... 18
- Figura 4.7. Evolución de la pirámide según el parámetro MaxLevel. 18
- Figura 4.8. Flujo óptico trazado para el mismo instante de video corriendo con el nivel de pirámide aplicado en cada caso en la parte superior derecha: nivel 1(a), nivel 2 (b), nivel 3 (c), nivel 4 (d), nivel 5 (e), nivel 6 (f), nivel 8(g), nivel 10 (h). 19
- Figura 4.9. Flujo óptico trazado para el mismo instante de video caminando con el nivel de pirámide aplicado en cada caso en la parte superior derecha: nivel 1(a), nivel 2 (b), nivel 3 (c), nivel 4 (d), nivel 5 (e), nivel 6 (f), nivel 8(g), nivel 10 (h). 20
- Figura 4.10. Neguentropía mínima media ante diferentes niveles piramidales 21

- Figura II.1. Neguentropía mínima calculada para los casos en los que la frecuencia de lectura es: fotograma a fotograma(a), de dos en dos fotogramas (b), de cinco en cinco fotogramas(c) y de 10 en 10 fotogramas (d) 26
- Figura II.2. Neguentropía mínima calculada para los casos en los que la frecuencia de lectura es: de 20 fotogramas en 20 fotogramas(a), de 21 fotogramas en 21 fotogramas (b), y de 22 en 22 fotogramas(c)..... 27

- Figura III. 1. Valores de neguentropía mínimos obtenidos corriendo para un nivel piramidal de: 0 (a), 1(b), 2(c), 3(d), 4(e), 5 (f), 6 (g), 8 (h) y 10 (i). 29
- Figura III. 2. Valores de neguentropía mínimos obtenidos caminando para un nivel piramidal de: 0 (a), 1(b), 2(c), 3(d), 4(e), 5 (f), 6 (g), 8 (h) y 10 (i). 30

Capítulo 1. Introducción

1.1 Motivación

Este proyecto, que se enmarca en el área de sistemas de la universidad de Zaragoza, parte de la idea de estudiar aspectos que pudieran mejorar un programa de SLAM existente surgido en 2003 del trabajo de A.J. Davison [3] y ampliado posteriormente en 2008 [4] y 2010 [5] que es capaz de realizar mapeo y posicionamiento con sensores de visión (visual SLAM) que ha sido adaptado con éxito para cámaras convencionales y catadióptricas y funciona correctamente con imágenes tomadas con cámaras wearables de los tipos mencionados por personas caminando y por robots cuadrúpedos o con ruedas.

Este programa ha sido probado a su vez con imágenes tomadas con una cámara wearable por una persona corriendo y a los pocos fotogramas el programa se bloquea siendo imposible realizar el posicionamiento ni el mapeo como sería deseable.

Las dificultades de dicho programa para realizar el seguimiento de los puntos característicos de las imágenes tomadas por una persona en el desarrollo de esta actividad física en concreto, correr, marca el punto de partida del presente trabajo fin de grado, que nace con la intención de estudiar el por qué del problema en el seguimiento de los puntos característicos en el desarrollo de esta actividad y que pretende analizar determinados aspectos que pudieran mejorar el funcionamiento del tracking ante este tipo de movimientos.

Para el estudio del problema y los experimentos posteriores se contará con imágenes reales tomadas por cámaras portables grabadas por personas andando y corriendo.

1.2 Justificación del proyecto

El problema del SLAM, posicionamiento y mapeo en 3D, ha sido ampliamente tratado en los últimos años. Plantea la posibilidad de que un móvil sea capaz de trazar un mapa de un entorno desconocido y localizarse en este con la información que aporten sus sensores de medición.

Se pueden utilizar muchos tipos de sensores para obtener información del entorno: sensores de radar, laser, sonar, visión... Debido al alto coste y a las limitaciones que presentan algunos de estos, en los últimos tiempos los más utilizados son los sensores de visión (visual SLAM) más baratos y capaces de obtener mayor información del entorno.

Este tipo de aplicación presenta dificultades en el seguimiento de los puntos característicos ante movimientos bruscos y rápidos de la cámara, movimientos que se dan a menudo en las actividades deportivas y especialmente corriendo.

Esto unido al cada vez más extendido uso de todo tipo de wearables en los deportes, incluidas las cámaras, sugiere la posible inclusión del SLAM en las actividades deportivas y de ocio motivando así un estudio del problema y de posibles soluciones ante este tipo de situaciones.

Cabe señalar también que los movimientos bruscos y rápidos no se producen únicamente en el desarrollo de actividades deportivas. Pueden producirse, por ejemplo, en vehículos aéreos no tripulados (drones), en vehículos que circulan por terrenos abrupto, etc. Existen situaciones en las que las aplicaciones de SLAM podrían tener problemas para el seguimiento de los puntos característicos así que alcanzar una solución o mejora en el caso concreto de las imágenes corriendo podría aportar una solución a otras situaciones en las que se den este tipo de inconvenientes.

1.3 Objetivos

Los objetivos marcados para este proyecto son:

- Estudio del problema del seguimiento del flujo óptico ante los movimientos bruscos y rápidos en videos grabados corriendo.
- Definición de los parámetros de estudio y de las posibles soluciones.
- Realización de pruebas en simulación y con imágenes reales.
- Análisis de los datos obtenidos y conclusiones.

Capítulo 2. Métodos de tracking

En las aplicaciones de Visual SLAM, para poder llevar a cabo el proceso de mapeo y posicionamiento, se comienza con la extracción de puntos característicos de la imagen, y con la asignación de un descriptor se indentifica cada uno de los puntos. El extractor de características procesa la imagen y selecciona los puntos clave de esta, que son aquellos en los que se produce una variación brusca de la iluminación. Estos puntos son conocidos como corners (esquinas). El tiempo de computación necesario para el procesamiento de la imagen y la extracción de corners es elevado, buscando reducir este coste computacional, se utilizará un extractor FAST desarrollado por Rosten et. al. [6].

El descriptor tiene como labor proporcionar una identidad a cada uno de los puntos clave extraídos para que puedan ser reconocidos en medidas posteriores. Existen descriptores de diferentes tipos y complejidades, el más sencillo consiste en un parche que toma como centro el punto extraído aunque existen otros más sofisticados como SIFT[7], SURF[8], LBP[9], etc.

El objetivo de este trabajo está enfocado a estudiar el seguimiento de los puntos una vez extraídos y descritos. Para ello existen numerosas técnicas siendo especialmente relevantes en este caso las de template matching y flujo óptico, que se detallan continuación.

2.1 Template matching

La técnica del template matching se basa en realizar un proceso de correlación de un punto definido en la imagen actual con un punto definido en una imagen previa mediante la comparación de plantillas, de esta forma se estima el desplazamiento ocurrido entre una imagen y otra.

Es una técnica muy extendida para la búsqueda de partes concretas de una imagen, como la detección de rostros. En este caso el parche sería la parte de la imagen que queremos encontrar y se proyectaría sobre la imagen y se recorriendo esta hasta encontrarlo (Figura 2.1).



Figura 2.1. Aplicación de Template Matching para la detección de rostros.

En el caso del SLAM la técnica del template matching se aplica de forma parecida: dado un punto característico se predice su posición en el siguiente fotograma mediante un modelo de movimiento de la cámara. Alrededor de la posición predicha define una región de búsqueda en la que se toma como emparejamiento aquel pixel cuya región vecina tiene la mayor correlación en intensidad respecto al parche buscado, con la condición de que dicha correlación esté por debajo de un umbral preestablecido.

Esta técnica es la empleada en el programa de SLAM comentado anteriormente y recibe el nombre de *active search*. Su ventaja principal es que requiere un menor tiempo computacional, algo imprescindible en el SLAM a tiempo real, pero da problemas ante movimientos rápidos o bruscos de la cámara.

2.2 Flujo óptico

En visión por computador, se conoce como flujo óptico al movimiento aparente entre patrones de brillo de una imagen [10][11]. El cálculo del flujo óptico es un problema complejo, por ello los algoritmos existentes se basan en determinadas hipótesis: intensidad constante, rigidez de los objetos, coherencia espacial, etc. El problema se puede explicar de la siguiente manera:

- Tenemos dos imágenes en escala de grises: I, J .
- Definimos el valor de gris en la imagen I en las coordenadas (x, y) como $I(x, y)$.
- Tomamos $u = [u_x, u_y]$ como un punto en la imagen I .
- El objetivo es encontrar un punto v en J donde $I(u)$ y $J(v)$ sean parecidos, el mismo punto.

$$v = u + d = [u_x + d_x ; u_y + d_y]$$

- $d = [d_x, d_y]$ es el flujo óptico del punto u .

La principal hipótesis a la hora de plantear la ecuación de flujo óptico es la conservación de la intensidad o nivel de gris en el tiempo, pese a que el punto se haya desplazado[12]. Matemáticamente se puede expresar de la siguiente manera:

$$f(x, y, t) = f(x + \Delta x, y + \Delta y, t + \Delta t)$$

donde $f(x, y, t)$ es el valor de gris en la posición (x, y) en el instante de tiempo t , Δx y Δy son la variación de posición y Δt es la variación de tiempo. Aplicando el desarrollo en serie de Taylor a la parte derecha:

$$f(x + \Delta x, y + \Delta y, t + \Delta t) = f(x, y, t) + \frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial y} \Delta y + \frac{\Delta f}{\Delta t} \Delta t + \varepsilon$$

donde $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial y}$ y $\frac{\Delta f}{\Delta t}$, son las derivadas de la imagen en x , y , y t . Los términos restantes agrupados en ε podemos ignorarlos al ser de orden mucho mayor. Si combinamos la primera y la segunda ecuación obtenemos la siguiente expresión:

$$\frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial y} \Delta y + \frac{\Delta f}{\Delta t} \Delta t = 0$$

dividiendo entre Δt :

$$\frac{\partial f}{\partial x} \frac{\Delta x}{\Delta t} + \frac{\partial f}{\partial y} \frac{\Delta y}{\Delta t} + \frac{\Delta f}{\Delta t} = 0$$

que podemos expresar de la siguiente forma:

$$\frac{\partial f}{\partial x} V_x + \frac{\partial f}{\partial y} V_y + \frac{\Delta f}{\Delta t} = 0$$

siendo V_x , V_y las componentes en x e y de la velocidad o flujo óptico de $f(x, y, t)$, y $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial y}$, y $\frac{\Delta f}{\Delta t}$ las derivadas de la imagen en (x, y, t) que se pueden transformar en:

$$f_x V_x + f_y V_y = -f_t$$

Finalmente agrupando obtenemos la ecuación de flujo óptico:

$$\Delta f * V + f_t = 0$$

Hay que tener en cuenta que cada vector de flujo óptico V tiene dos componentes y una sola ecuación. Esto implica que el cálculo del flujo óptico entre dos imágenes es un problema indeterminado que admite infinitas soluciones. Según la forma de resolver esta ambigüedad los algoritmos de flujo óptico pueden agruparse en dos tipos: algoritmos de flujo óptico denso y algoritmos de flujo óptico disperso.

2.2.1 Flujo óptico denso

Los algoritmos de flujo óptico denso se caracterizan por computar el vector velocidad para cada pixel de la imagen (Figura 2.2)



Figura 2.2. Ejemplo de flujo óptico denso.

Tienen su origen en el trabajo de Horn y Shunk[13], donde se propone la minimización de la siguiente función de energía:

$$\sum \|\Delta f_i V_i + f_{t,i}\|^2 + \lambda (\|\nabla V_{ix}\|^2 + \|\nabla V_{iy}\|^2)$$

donde el primer sumando representa la restricción de flujo óptico, el segundo es un regularizador que introduce la restricción adicional de que el flujo óptico no cambia mucho en la vecindad de un pixel y el parámetro λ sirve para ajustar la contribución de ambos sumandos. El resultado final es un campo denso de descripción del movimiento que aportan gran cantidad de información, por otro lado son más sensibles al ruido y tienen un mayor coste computacional que los métodos locales.

2.2.2 Flujo óptico disperso

Los algoritmos de flujo óptico disperso no computan el vector velocidad de todos los pixeles de la imagen, únicamente lo hacen con un grupo reducido de puntos (Figura 2.3).



Figura 2.3. Ejemplo de flujo óptico disperso.

El método más extendido de este tipo es el de Lucas-Kanade[14]. Este método asume que los pixeles vecinos al punto característico detectado tienen un movimiento similar, por lo que toma un parche de tamaño n y considera que todos los puntos de este parche tienen el mismo movimiento:

$$\begin{aligned}
f_x(q_1)V_x + f_y(q_1)V_y &= -f_t(q_1) \\
f_x(q_2)V_x + f_y(q_2)V_y &= -f_t(q_2) \\
&\vdots \\
&\vdots \\
&\vdots \\
f_x(q_n)V_x + f_y(q_n)V_y &= -f_t(q_n)
\end{aligned}$$

donde $q_1, q_2 \dots q_n$ son los pixeles dentro del parche y $f_x(q_i), f_y(q_i), f_t(q_i)$ son las derivadas parciales de la imagen f con respecto a las direcciones x, y , y el tiempo t . Al escribir estas ecuaciones como matrices obtenemos:

$$A = \begin{pmatrix} f_x(q_1) & f_y(q_1) \\ f_x(q_2) & f_y(q_2) \\ \dots & \dots \\ f_x(q_n) & f_y(q_n) \end{pmatrix}, \quad v = \begin{bmatrix} V_x \\ V_y \end{bmatrix}, \quad y \quad b = \begin{bmatrix} -f_t(q_1) \\ -f_t(q_2) \\ \dots \\ -f_t(q_n) \end{bmatrix}$$

dando lugar a un sistema sobre dimensionado con n ecuaciones y dos incógnitas. El método de Lukas-Kanade obtiene una solución aplicando el método de mínimos cuadrados, planteando en primer lugar el siguiente sistema 2×2 :

$$A^T A v = A^T b$$

Despejando el vector de incógnitas v :

$$v = (A^T A)^{-1} A^T b$$

En las pruebas realizadas durante el desarrollo del presente proyecto se utiliza el método de Lukas-Kanade, se escoge este método con la intención de comprobar cómo responde ante los movimientos grabados corriendo y comprobar si es capaz de superar las limitaciones del método de búsqueda activa en el programa de monoSLAM.

2.3 Flujo óptico en pirámides

La ecuación de flujo óptico se obtiene a partir de una aproximación en serie de Taylor de primer orden lo que provoca que el algoritmo de Lukas Kanade solo sea efectivo ante pequeños desplazamientos de los pixeles. La solución a este problema es la implementación piramidal[15].

Si tenemos un fotograma I con un tamaño inicial I^0 se toma este como base de la pirámide, partiendo de esta base la implementación piramidal construye diferentes niveles:

$$I^0 \rightarrow I^1 \rightarrow I^2 \rightarrow I^3 \rightarrow \dots \rightarrow I^{Lm}$$

en cada nivel se reduce a la mitad cada coordenada, x e y , luego en cada nivel el tamaño de la imagen será $\frac{1}{4}$ de la anterior, $I^L = \frac{1}{4} I^{L-1}$ (**¡Error! No se encuentra el origen de la**

referencia.) Para calcular el flujo óptico se deben seguir los puntos característicos a través de los diferentes niveles de la pirámide.

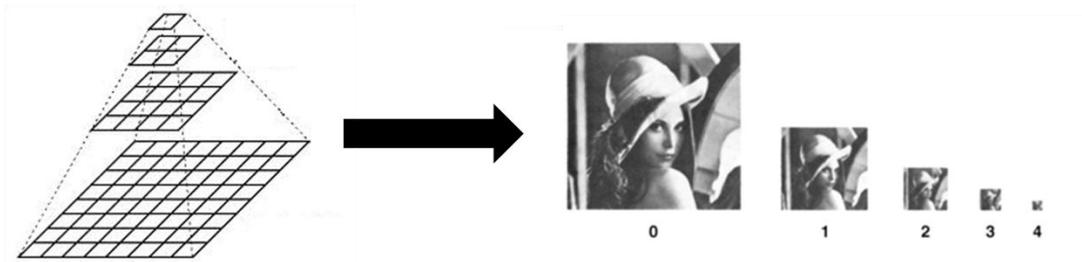


Figura 2.4. Pirámide de imágenes.

Dado un punto u del nivel más alto de la pirámide de una imagen I , el objetivo es localizar el punto v del nivel anterior tal que $v = u + d$. El punto correspondiente a $u(u^0)$ en el nivel piramidal I^l es $u^L = \frac{u}{2^L}$. El seguimiento se efectúa de la siguiente manera:

- Se calcula d^{Lm} en el nivel L_m , el más alto de la pirámide.
- Se hace una predicción inicial de d^{Lm-1} a partir de d^{Lm} y se proyecta en el nivel L_{m-1} .
- Así sucesivamente hasta llegar al nivel 0 de la pirámide, a la imagen inicial.

Si se supone $g^L = [g_x^L \ g_y^L]$ una predicción inicial en el nivel L , la función de error de la localización del punto en la nueva imagen, ϵ^L , será:

$$\epsilon^L = \epsilon^L(d_x^L, d_y^L) = \sum_{x=u_x^L-w_x}^{u_x^L+w_x} \sum_{y=u_y^L-w_y}^{u_y^L+w_y} \left(I^L(x, y) - J^L(x + g_x^L + d_x^L, y + g_y^L + d_y^L) \right)^2$$

Siendo $(2w_x + 1) \times (2w_y + 1)$ el tamaño de la ventana, constante en toda la pirámide, y $d^L = [d_x^L \ d_y^L]$ el vector de desplazamiento residual del pixel que al ser más pequeño resulta mejor aplicarle el algoritmo de Lukas Kanade mientras que g^L se utiliza para trasladar el parche de la imagen I a la segunda imagen J . Si se da por bueno el valor de d^L , la predicción g^{L-1} para el siguiente nivel será:

$$g^{L-1} = 2(g^L + d^L)$$

Asumiendo como predicción inicial en el nivel más alto de la pirámide (L_m) $g^{Lm} = [0 \ 0]$, se obtiene la ecuación final de flujo óptico piramidal d :

$$d = \sum_{L=0}^{Lm} 2^L d^L$$

Durante los experimentos del presente proyecto se estudiara la mejora producida por la implementación piramidal en el cálculo del flujo óptico sobre las imágenes tomadas corriendo.

Capitulo 3. Estudio del flujo óptico ante movimientos bruscos de cámara

El estudio del flujo óptico ante movimientos bruscos de cámara así como los experimentos posteriores del presente proyecto están basados en el análisis de dos videos, uno corriendo y otro caminando, filmados con la misma cámara wearable, una GoPro[16], a 60 fps² con una resolución de 1280x720 píxeles.

3.1 Flujo óptico corriendo vs flujo óptico caminando

Como primera aproximación al problema en concreto, se lleva a cabo un estudio del flujo óptico corriendo en comparación con el flujo óptico caminando. Se extraen 4 fotogramas de cada uno de los videos y se analizan mediante un programa elaborado en c++ integrando funciones de OpenCV. El programa lee dos fotogramas indicados por el usuario y devuelve el flujo óptico entre ambos, calculado por el método de Lukas-Kanade, en forma de vectores dibujados sobre el primer fotograma. Esto se lleva a cabo con la intención de hacer un análisis visual del flujo óptico en ambos casos, ver las diferencias, escoger los parámetros de estudio y plantear las posibles soluciones. Todas las funciones de OpenCV utilizadas en este proyecto, tanto en este experimento como en los restantes, se especifican en el Anexo I.

Se escoge un primer fotograma de cada video que se utiliza como fotograma base (0), como segundo fotograma se elige el que va a continuación (1), como tercero el que está a 5 fotogramas de distancia de primero (5) y el cuarto a una distancia de 30 fotogramas (30) como se indica en la Figura 3.1.

Figura 3.1

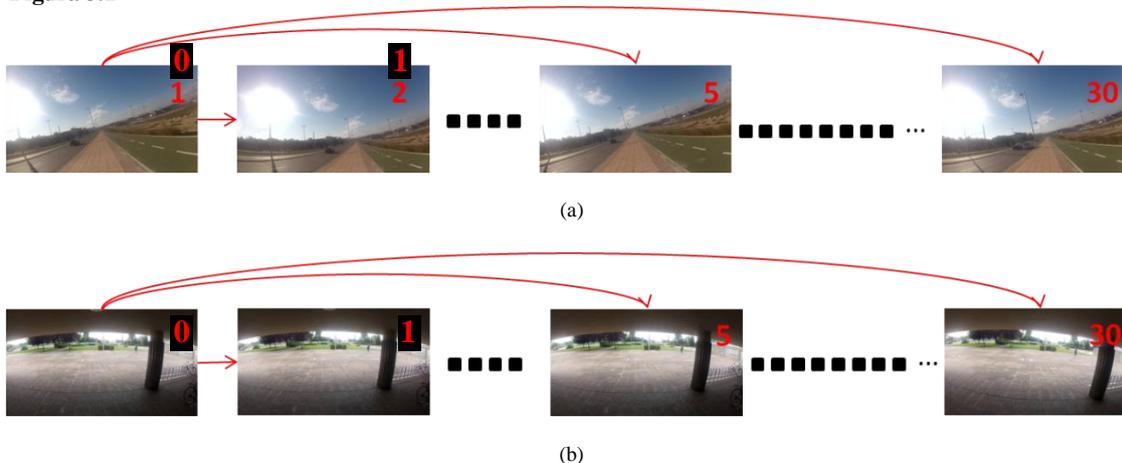


Figura 3.1. Fotogramas seleccionados corriendo (a) y caminando (b)

² fotogramas por segundo

Los resultados obtenidos (Figura 3.2) en el caso de los fotogramas caminando muestran, para la distancia de un fotograma, unos vectores de flujo óptico cortos, ordenados en dirección al movimiento, que en este caso es un giro a la izquierda, y que unen puntos aparentemente bien emparejados. Cuando la distancia es de 5 fotogramas, los vectores de flujo óptico aparecen con una longitud mayor pero siguen estando ordenados y la correlación de puntos sigue siendo correcta. A una distancia de 30 fotogramas, el flujo óptico que dibuja resulta algo caótico, sobre todo en la parte inferior derecha donde hay una mayor concentración de puntos característicos, apreciándose claramente puntos mal emparejados y desorden en los vectores.

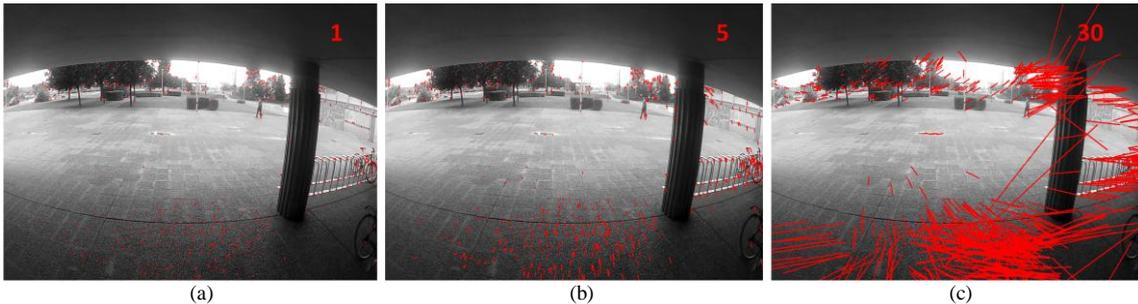


Figura 3.2. Flujo óptico caminando a la distancia de: 1 fotograma (a), 5 fotogramas (b) y 30 fotogramas (c)

En los fotogramas analizados del video grabado corriendo (Figura 3.3), para la distancia de un fotograma se observa que los vectores de flujo óptico son mucho más largos que caminando. En muchas regiones de la imagen aparecen desordenados, se aprecian puntos claramente mal emparejados e indican un movimiento de oscilación en torno al eje del corredor hacia la derecha. Sin embargo, para la distancia de 5 fotogramas, el flujo óptico aparece más ordenado y apunta en la dirección hacia la que avanza el corredor en lugar de representar la inclinación entorno a su propio eje. El resultado del flujo a la distancia de 30 fotogramas vuelve a ser caótico y con malos emparejamientos.

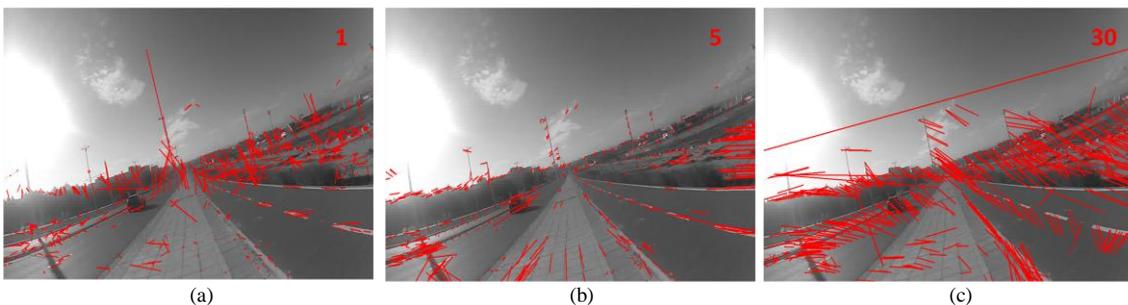


Figura 3.3. Flujo óptico corriendo a la distancia de: 1 fotograma (a), 5 fotogramas (b) y 30 fotogramas (c)

3.2 Neguentropía de flujo como herramienta de evaluación

En el estudio descrito en el apartado 3.1 se puede ver que la diferencia principal entre el flujo óptico trazado entre los fotogramas del video caminando y los del video corriendo es, a parte de la longitud del vector, el desorden de los vectores. Si en un área pequeña de la imagen tenemos vectores de flujo óptico que se cruzan entre sí con desorden, es indicativo de que el seguimiento de estos puntos característicos no ha sido correcto (Figura 3.4). Píxeles de la imagen cercanos entre sí deberían moverse en direcciones paralelas o casi paralelas entre fotogramas cercanos, esto implica describir el mismo ángulo sobre la horizontal.

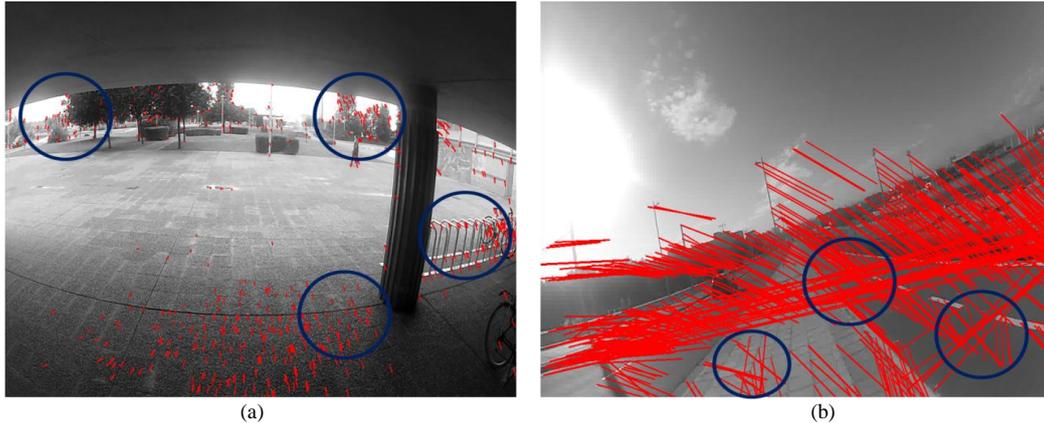


Figura 3.4. Zonas de flujo óptico bien calculado, ordenado (a); zonas de flujo óptico mal calculado, desordenado (b)

Para poder utilizar este hecho como herramienta de evaluación en los sucesivos experimentos se cuantifica el orden o desorden de los vectores de flujo mediante la neguentropía de la siguiente manera:

Se proyecta sobre cada fotograma una cuadrícula de $8 \times 4,5$ elementos, teniendo cada elemento 160×160 píxeles (Figura 3.5). Se escogen estas medidas suponiendo que cada elemento es lo suficientemente pequeño como para que los vectores de flujo óptico que se encuentren dentro de este sean paralelos pero el número de cálculos no incremente en exceso el coste computacional.

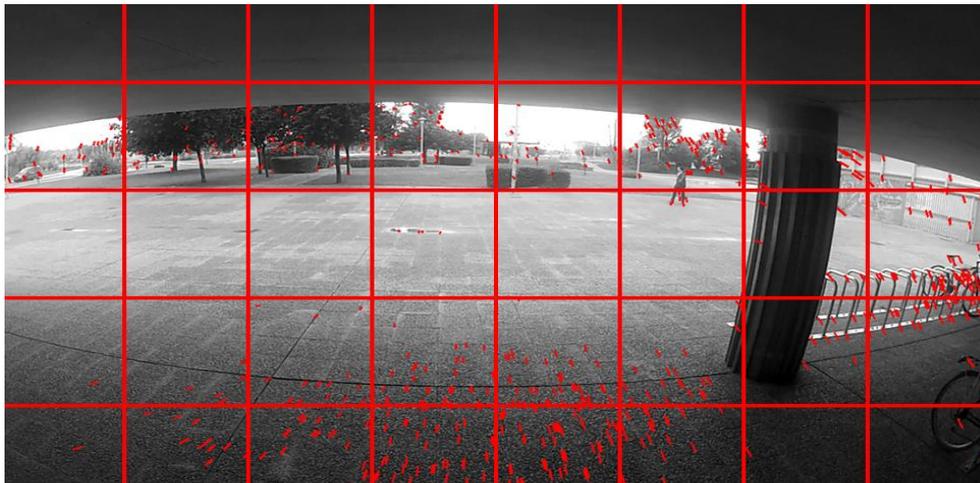


Figura 3.5. Ejemplo de cuadrícula proyectada sobre un fotograma

Cada vector de flujo que devuelve el programa posee dos componentes, una componente de longitud del vector y otra del ángulo:

$$v_i = \begin{pmatrix} V_{x,i} \\ V_{y,i} \end{pmatrix}$$

De cada vector podemos extraer el ángulo mediante una función *atan2*:

$$\theta_i = \text{atan2}(V_{y,i}, V_{x,i})$$

Para calcular el histograma [17], en primer lugar es necesario discretizar o agrupar los valores de los ángulos de los distintos vectores de flujo en N grupos. Cada

grupo tiene una anchura o paso (en inglés, gap) que depende del rango de posibles valores del ángulo y del número de grupos. En este caso se toma $N=24$.

$$gap = \frac{2\pi}{N} = \frac{\pi}{12}$$

Una vez calculado el paso, se le asigna a cada ángulo un grupo (bin, en inglés) mediante la siguiente ecuación:

$$bin_i = floor\left(\frac{\theta_i - 0}{gap}\right) \text{ en } \{0, 1, \dots, N - 1\}$$

siendo $floor()$ la función de redondeo del entero más próximo por defecto. Cada vez que un ángulo es clasificado dentro de un grupo, se incrementa en una unidad el número de sucesos en dicho grupo del histograma.

$$hist[bin_i] = hist[bin_i] + 1$$

El resultado final es un histograma como el de la Figura 3.6.

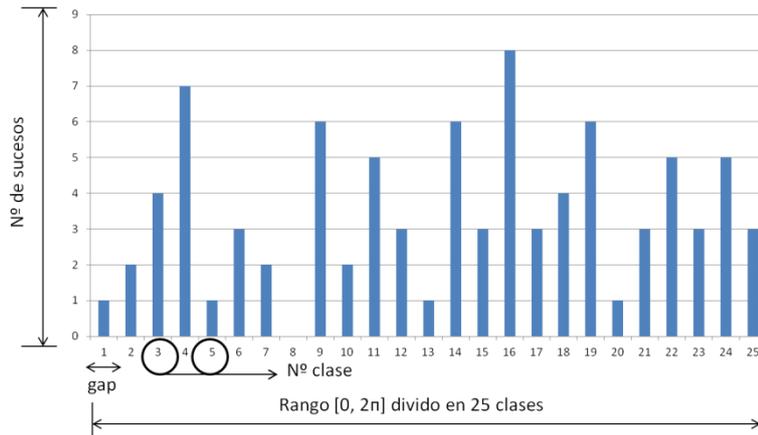


Figura 3.6. Ejemplo de histograma de ángulos

A continuación se normaliza el histograma dividiendo por el número total de sucesos para obtener la frecuencia de cada grupo:

$$f_j = hist[j]/M$$

donde f_j es la frecuencia del grupo j , debiéndose cumplir la restricción de $\sum_j f_j=1$. M es el número de sucesos totales, en este caso el número de vectores de flujo en el elemento analizado de la cuadrícula e $hist[j]$ es el número de veces que ha aparecido un ángulo de la clase j . A partir de un histograma normalizado se puede calcular la entropía del mismo. La entropía de un histograma es una medida de cómo están distribuidos el número de sucesos entre los distintos grupos. Un histograma con los sucesos uniformemente distribuidos por todos los grupos tendrá una entropía mayor, mientras que si los sucesos se agrupan en unos pocos grupos, la entropía será menor.

La entropía se calcula de la siguiente manera:

$$H = - \sum f_i \times \log f_i$$

Para el análisis de los sucesivos experimentos en deseable que la entropía esté normalizada. Para ello debemos dividir la entropía, por el máximo valor de que se alcanza cuando todos los grupos tienen la misma frecuencia. Dado que se debe cumplir la restricción de $\sum f_i = 1$, la entropía máxima se da cuando:

$$f_i = \frac{1}{N} \quad \forall j \text{ en } \{0,1, \dots N - 1\}$$

Y por tanto tenemos:

$$H_{max} = - \sum_{j=0}^{N-1} \left(\frac{1}{N} \times \log \frac{1}{N} \right) = - \log \frac{1}{N} = \log N$$

pudiendo obtener finalmente la entropía normalizada:

$$H_{norm} = \frac{H}{H_{max}}$$

De la entropía normalizada obtendremos valores entre 0 y 1 siendo más favorable cuanto más cerca de 0 esté ya que lo deseable es que los vectores estén ordenados. Como esto puede resultar poco intuitivo se opta por utilizar la neguentropía normalizada, NH_{norm} , que es lo contrario de la entropía normalizada:

$$NH_{Norm} = 1 - H_{Norm}$$

En los experimentos posteriores se calculara mediante código un valor de neguentropía para cada componente de la cuadrícula de cada fotograma analizado, el programa devuelve el valor mínimo, máximo y medio de neguentropía por fotograma, cuanto más próximos a 1 estén estos valores mejor calculado se considera el flujo óptico.

Capitulo4. Realización de experimentos

El presente proyecto se ha centrado en el estudio del flujo óptico ante movimientos brusco, en concreto, los que se producen al correr. En este capítulo se busca, mediante la realización de dos experimentos, proponer posibles mejoras para el tracking ante este tipo de situaciones.

Trabajos recientes han conseguido con éxito combinar las técnicas del SLAM visual con el estudio de la frecuencia de los movimientos en cámaras wereables, como por ejemplo, en el trabajo de D.Gutierrez y J.J.Guerrero[18], en el que se consigue calcular la escala en un programa de SLAM visual mediante la frecuencia de oscilación al caminar. En el primer estudio de los fotogramas extraídos del video corriendo, explicado en el apartado 3.1 del presente informe, se observaba como el seguimiento de los puntos característicos de la imagen parece mejor a una distancia de 5 fotogramas que el que se obtiene con el fotograma siguiente. En el primer experimento se trabaja con la idea de que saltando determinados fotogramas o estudiando la frecuencia a la que se mueve el portador de la cámara, se podría evitar el seguimiento de movimientos bruscos indeseados propios de la actividad y que pueden complicar el tracking, y centrarlo principalmente en el seguimiento de la trayectoria.

En el segundo experimento se evalúa la mejora que supone la aplicación de pirámides de imagen ante movimientos rápidos y bruscos de la cámara y ante movimientos más suaves, como los que se producen caminando, con la intención de encontrar una respuesta al bloqueo del programa MonoSLAM ante este tipo de secuencias.

Para la realización de ambos experimentos se toman 15 segundos de cada video de los mencionados con anterioridad, esto son 900 fotogramas de cada uno de ellos. El tamaño de las muestras está pensado para poder estudiar los resultados sin aumentar en exceso el coste computacional. Las herramientas de evaluación en ambos casos serán:

- **Análisis visual:** Una vez calculado el flujo óptico entre los 900 fotogramas, se estudian estos por separado y se vuelven a unir conformando los 15 segundos de video pero con el flujo óptico trazado para proceder a realizar el estudio visual.
- **Neguentropía:** En los programas realizados para ambos experimentos se incluyen líneas de código que calculan la neguentropía de la forma explicada en el apartado 3.2, devolviendo un archivo de texto con 3 columnas para cada fotograma analizado. En la primera de ellas la neguentropía mínima calculada, en la segunda el valor medio media y en la tercera con la máxima. El estudio se centra en los valores de neguentropía mínima. Estos datos se grafican posteriormente y se analizan.

4.1 Experimento I. Estudio de frecuencia

Para realizar este experimento, en primer lugar, se modifica el programa utilizado en el apartado 3.1 para que en vez de calcular el flujo óptico entre dos fotogramas lo haga entre una secuencia de estos y se incluye el cálculo de la neguentropía. Se estructura el programa de tal manera que variando un parámetro podemos indicarle con qué frecuencia leer los fotogramas. Llegados a este punto el experimento se puede dividir en dos test:

4.1.1 Test 1

En el primer test se calcula el flujo óptico por el método de Lukas-Kadane con tres niveles piramidales, por defecto en la función de OpenCV, de los 900 fotogramas seleccionados del video corriendo, con una frecuencia de fotograma a fotograma. Posteriormente se modifica el programa para que la frecuencia de lectura de los fotogramas sea cada 2 fotogramas, cada 5 y cada 10 (Figura 4.1) y se calcula de nuevo para cada caso.

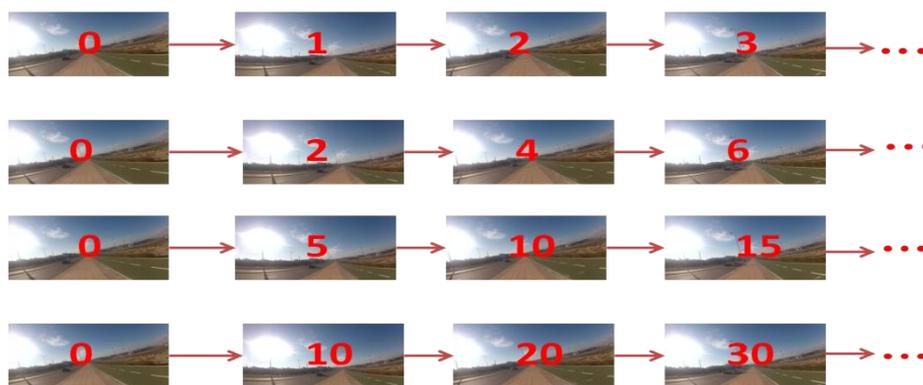


Figura 4.1. Esquema de secuencias aplicadas en el test 1 del experimento I.

Los fotogramas con el flujo óptico trazado obtenidos en cada caso, se vuelven a montar en un video y del análisis visual se extrae que hay pocas diferencias entre el flujo óptico calculado fotograma a fotograma y el que se obtiene de 2 fotogramas en 2 fotogramas. Al contrario de lo que cabía esperar tras el estudio explicado en el apartado 3.1 del presente proyecto, el flujo óptico trazado de 5 en 5 fotogramas y de 10 en 10 fotogramas es caótico y se aprecia claramente un gran número de malos emparejamientos. Como muestra, en la Figura 4.2 se muestra el mismo instante de video con el flujo trazado para cada caso.

En cuanto a la neguentropía calculada para cada frecuencia, muestra una clara tendencia a la baja conforme se aumenta la distancia entre fotogramas leídos, lo que indica un peor resultado cada vez. Esto se aprecia en la Figura 4.3, donde se muestra la entropía mínima media para cada caso estudiado.

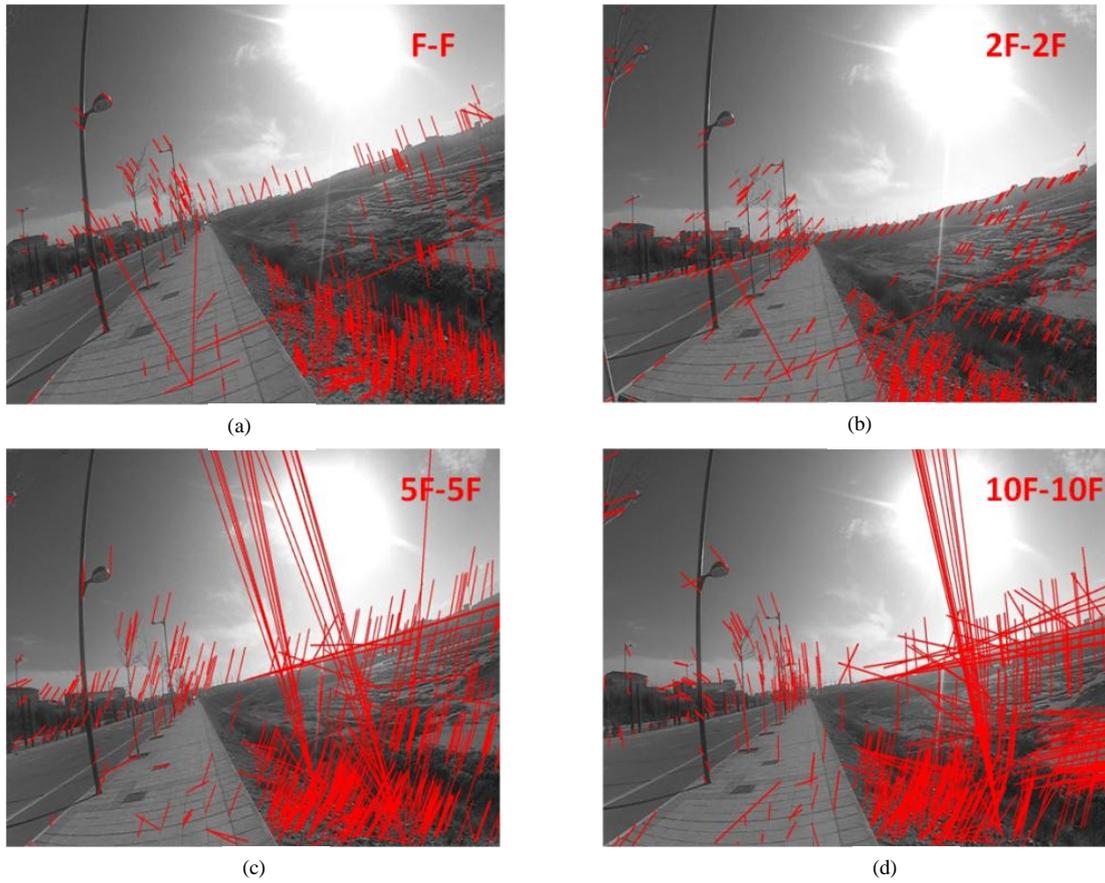


Figura 4.2. Flujo óptica trazado para la misma sección de video: fotograma a fotograma (a), de 2 en 2 (b), de 5 en 5(c) y de 10 en 10 (d).

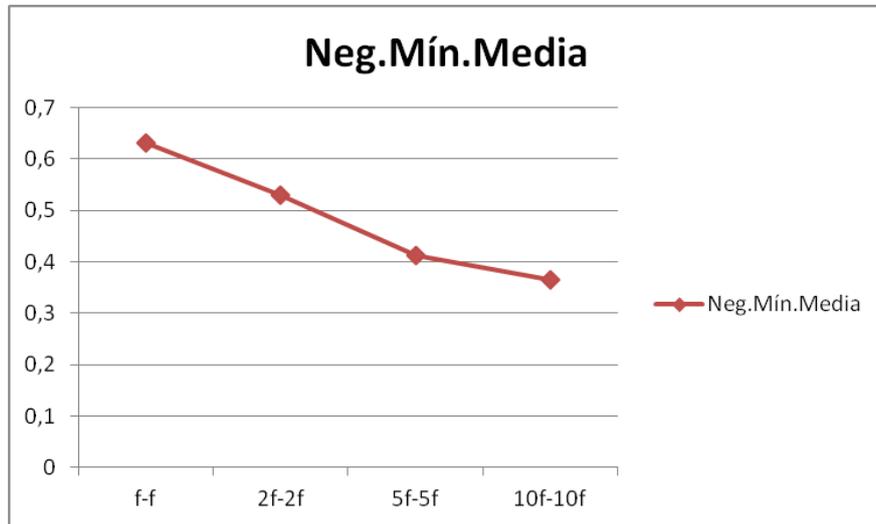


Figura 4.3. Neguentropía mínima media para cada caso

4.1.2 Test 2

En el análisis visual de los fotogramas obtenidos en test 1, para el caso en el que se analizan fotograma a fotograma los 900 seleccionados, se aprecia que el trazado del

flujo óptico varía en longitud y orden periódicamente. Cada vez que el corredor da un paso con cada pie, fruto de la gran aceleración lateral que se produce, se dificulta el emparejamiento de puntos y por tanto el cálculo del flujo óptico, mientras que cuando el corredor pasa por su eje vertical no parece haber esos problemas (Figura 4.4). En los valores mínimos de neguentropía de cada fotograma también se observa una cierta periodicidad entre los picos más bajos (Figura 4.5).



Figura 4.4. Flujo óptico cuando el corredor pasa por su eje vertical (a), cuando da un paso (b).

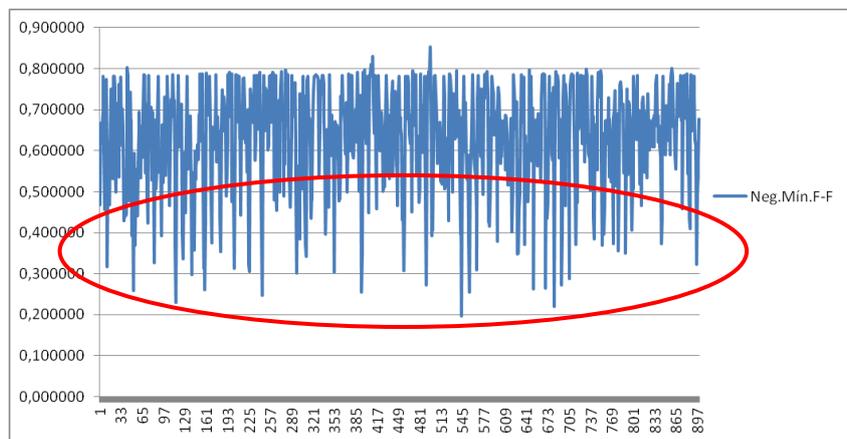


Figura 4.5. Valores de neguentropía mínimos para cada fotograma. Picos resaltados en rojo.

De la observación fotograma a fotograma de los 15 segundos de video se extrae que el corredor pasa por su eje vertical cada, más o menos, 21 fotogramas. Con el fin de evitar los cambios más bruscos entre fotogramas y calcular solo el flujo óptico en la dirección del movimiento, se repite el mismo experimento realizado en el test 1 pero en esta ocasión cada 20, 21 y 22 fotogramas. (Figura 4.6). En el análisis visual no se aprecian grandes mejoras, pero la neguentropía mínima media sí presenta un pico más alto en el cálculo entre 21 y 21 fotogramas.

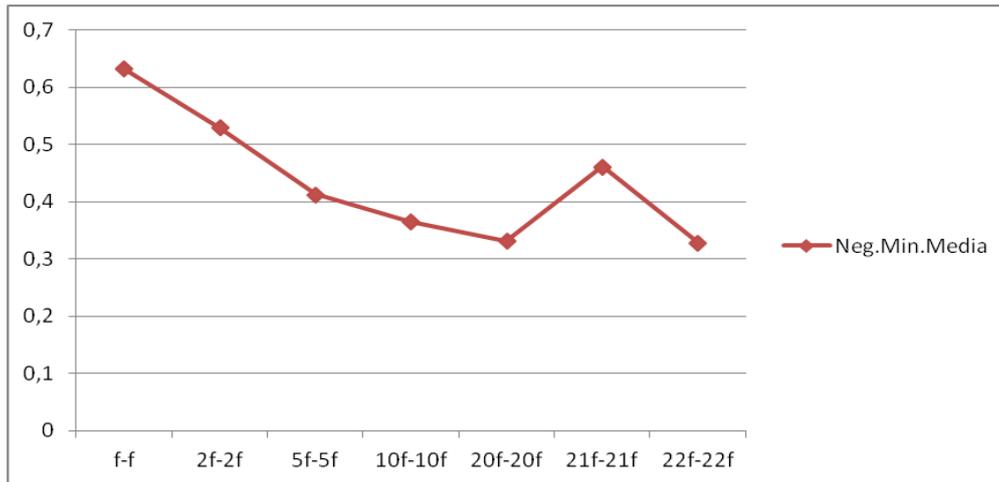


Figura 4.6. Neguentropía mínima media de las secuencias corriendo

Los resultados completos del experimento I pueden verse en el Anexo II del presente documento.

4.2 Experimento II: Pirámides de imágenes

En este segundo experimento, sobre el mismo programa, se varía el parámetro MaxLevel, de la función de OpenCV CalcOpticalFlowPyrLk, función que calcula el flujo óptico por el método de Lukas-Kanade (Anexo I), que indica el nivel de pirámide de imágenes a aplicar al cálculo. La intención es valorar si se puede trabajar con métodos sin pirámides ante movimientos de este tipo y comprobar si es este el motivo por el que no funciona el programa MonoSLAM. Se compararan los resultados nuevamente con los obtenidos en los, mismos supuestos, ante los movimientos suaves del video grabado caminando. En este caso se calculará el flujo óptico en los 900 fotogramas de cada muestra cada vez.

Se calculara el flujo óptico entre los valores (0-10), siendo el valor 0 una pirámide de 1 escalón, solo la base, y el valor 10 con 11 escalones (Figura 4.7).

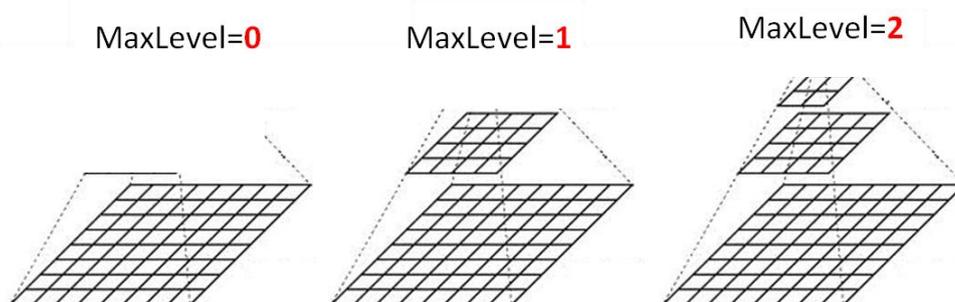


Figura 4.7. Evolución de la pirámide según el parámetro MaxLevel.

El análisis visual de los resultados revela una mejora clara en el trazado del flujo óptico conforme se va aumentando el nivel de pirámide en el caso de las imágenes corriendo (Figura 4.8). Nótese en la Figura 4.8 (a) como al calcular el flujo óptico directamente sobre los fotogramas originales sin usar pirámides, aparecen punto mal emparejados y vectores de flujo que se cruzan unos con otros. Sin embargo, al usar

pirámides de hasta 4 niveles se aprecia una notable mejoría. Con pirámides de 4 niveles en adelante apenas se aprecia variación en el cálculo del flujo óptico.

En los resultados andando, a nivel visual, no se aprecia la deferencia entre la aplicación de unos niveles de pirámide u otros (Figura 4.9).

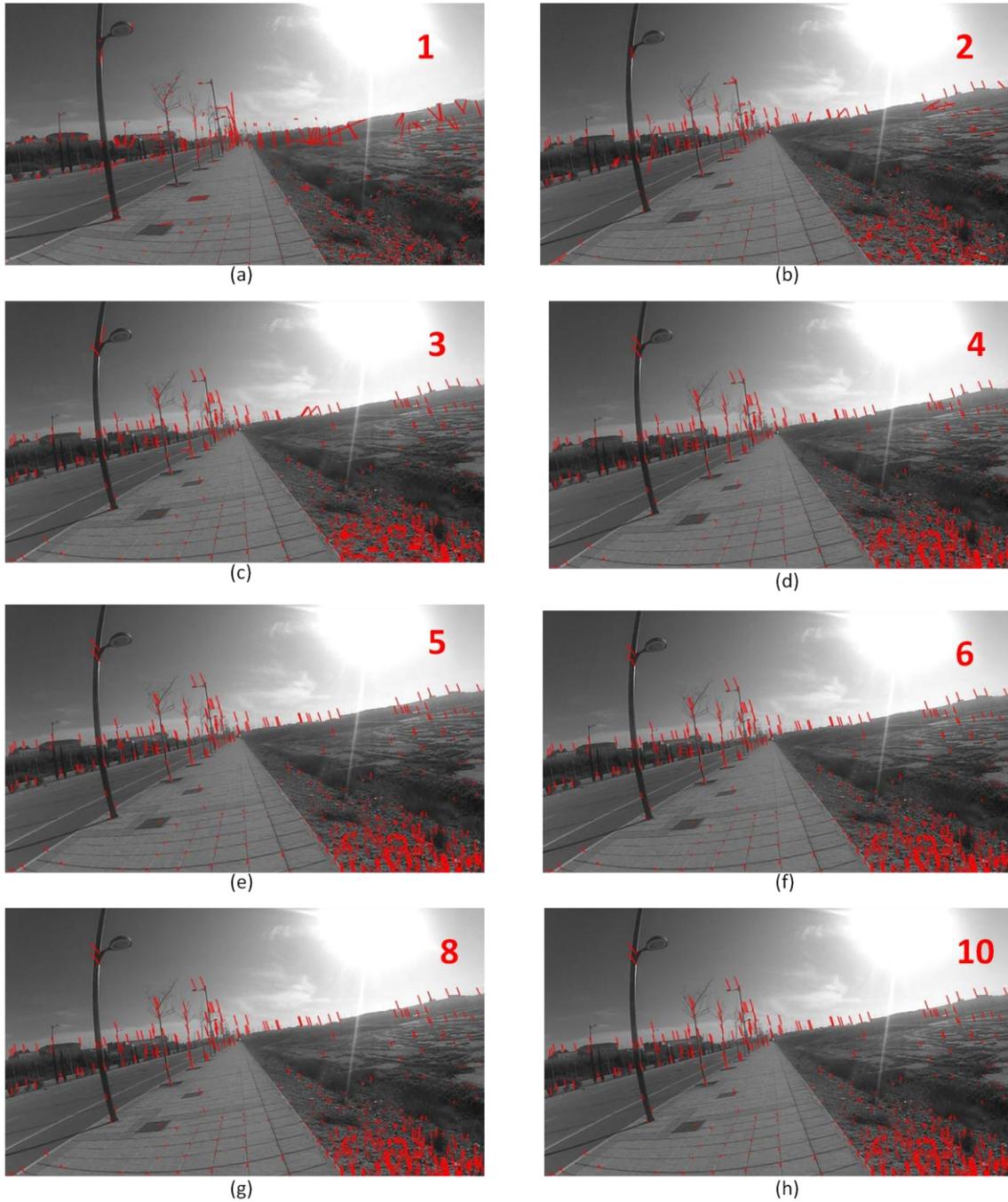


Figura 4.8. Flujo óptico trazado para el mismo instante de video corriendo con el nivel de pirámide aplicado en cada caso en la parte superior derecha: nivel 1(a), nivel 2 (b), nivel 3 (c), nivel 4 (d), nivel 5 (e), nivel 6 (f), nivel 8(g), nivel 10 (h).

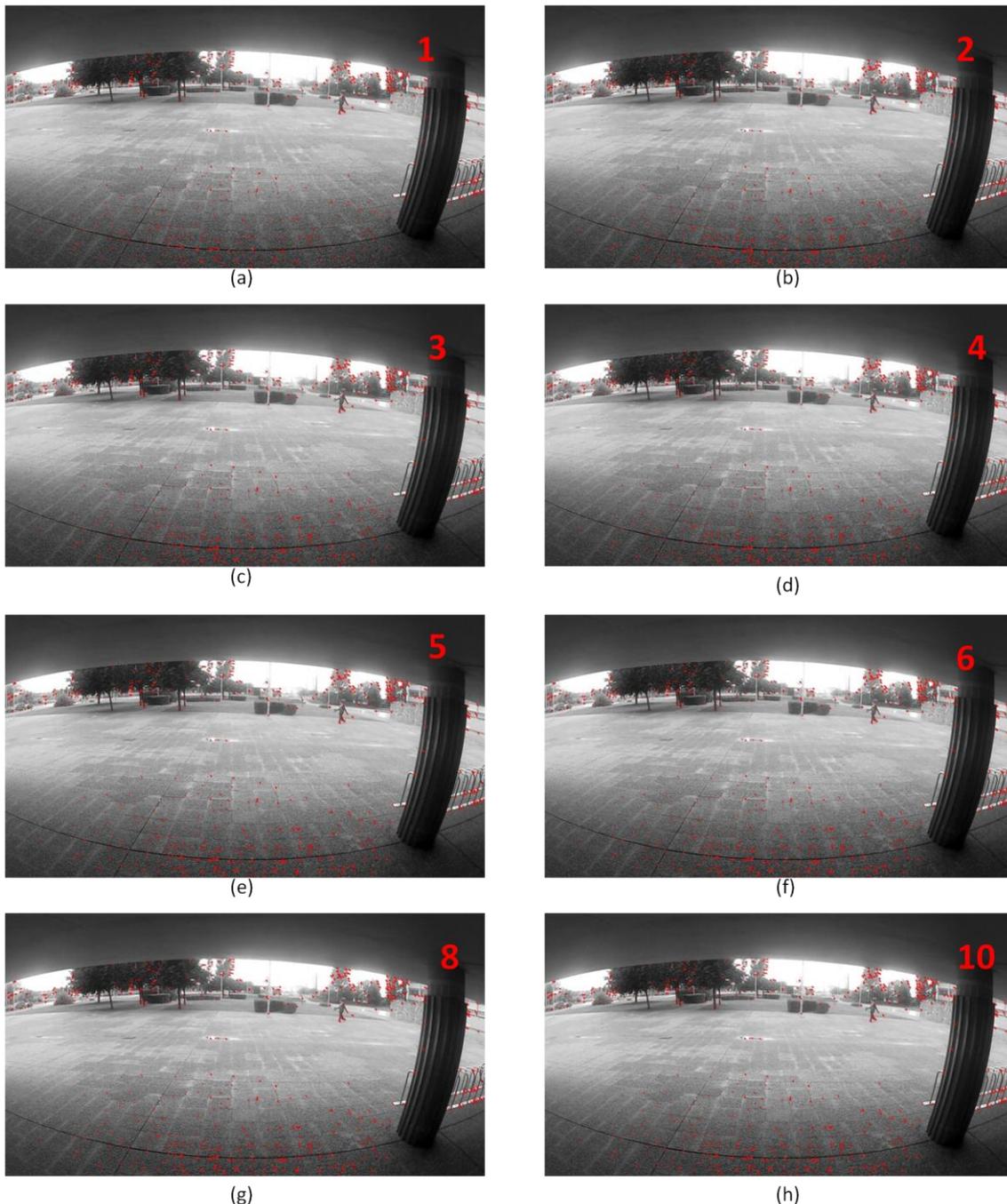


Figura 4.9. Flujo óptico trazado para el mismo instante de video caminando con el nivel de pirámide aplicado en cada caso en la parte superior derecha: nivel 1(a), nivel 2 (b), nivel 3 (c), nivel 4 (d), nivel 5 (e), nivel 6 (f), nivel 8(g), nivel 10 (h).

Analizando la neguentropía de flujo óptico (Figura4.9), comprobamos que la media de la neguentropía mínima en el caso del video corriendo es mucho menor que en el video caminando a un nivel 1 de pirámide, con tan solo la base, en cuanto se empiezan a añadir niveles los resultados ante los movimientos bruscos del video corriendo mejoran rápidamente.

Ante los movimientos suaves del video caminando, el valor de neguentropía permanece constante desde la pirámide de 3 escalones en adelante mientras que en el caso de los movimientos bruscos del video corriendo, el resultado mejora de manera

exponencial hasta el nivel 4 de pirámide, el valor aumenta algo del cuarto al quinto y permanece constante a partir de ahí aunque se sigan añadiendo escalones (Figura 4.10).

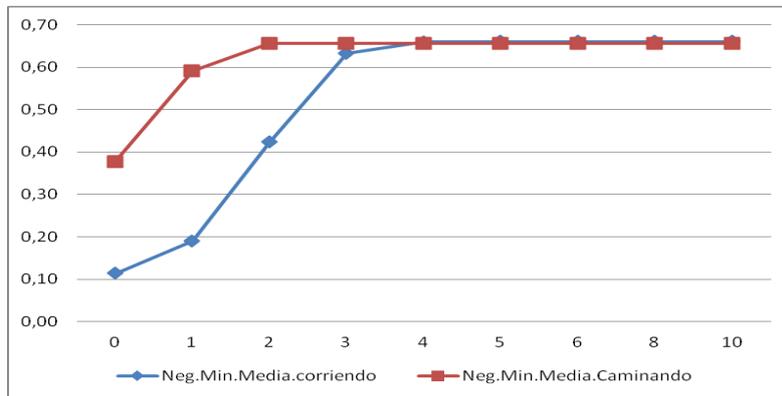


Figura 4.10. Neguentropía mínima media ante diferentes niveles piramidales

La saturación de la neguentropía a partir del nivel 4 de la pirámide se debe a la menor resolución de las imágenes ya que la magnitud de los vectores de flujo óptico a bajas resoluciones puede llegar a ser demasiado pequeña para el nivel de precisión dado por la discretización de la imagen en píxeles.

Capítulo 5. Conclusiones

El objetivo de este trabajo fin de grado ha sido el estudio de técnicas de tracking ante movimientos bruscos de la cámara. Más concretamente se han estudiado secuencias tomadas por la misma cámara wearable caminando, actividad durante la cual se producen movimientos suaves y lentos, y corriendo, actividad que conlleva movimientos bruscos y rápidos, con el fin de comprender las diferencias entre ambos y aportar soluciones.

La principal motivación de este trabajo ha sido el auge de las cámaras wearables para actividades deportivas combinado con las dificultades de las aplicaciones de SLAM visual existentes para realizar el seguimiento de los puntos característicos ante los movimientos propios de estas actividades.

En la primera fase del trabajo se ha estudiado el problema comparando las diferencias entre los flujos ópticos ante movimientos suaves y movimientos bruscos. La principal diferencia encontrada ha sido el orden de los vectores de flujo, siendo estos ordenados y paralelos en áreas próximas entre sí en el caso de los movimientos suaves y caóticos y secantes en el caso de los movimientos bruscos.

Del estudio preliminar del flujo, ante estas dos situaciones, surgió la idea de usar este orden o desorden de los vectores de flujo óptico como parámetro de medida en los experimentos a través de la neguentropía de flujo óptico. Este parámetro ha resultado de gran utilidad, cuantificando las mejoras de los métodos ensayados y llegando más allá de donde se puede llegar con el simple análisis visual. Además, ha resultado un método sencillo de programar y que no conlleva un gran coste computacional, por ello se cree que, con la ayuda de estudios posteriores, podría resultar un parámetro a tener en cuenta en las aplicaciones de SLAM visual asociadas a actividades deportivas pudiendo ser utilizado, por ejemplo, podría servir para la detección del tipo de deporte que se está realizando.

En el estudio preliminar se observó también que, en la secuencia corriendo, el trazado del flujo óptico resultó más claro y en dirección a la trayectoria saltando determinados fotogramas que calculando el flujo óptico entre todos ellos lo que llevo a la realización del experimento I, en el que se probó a calcular el flujo óptico variando la frecuencia de lectura de los fotogramas. El resultado no fue el esperado, en los tres casos ensayados, calculando cada 2 fotogramas, cada 5 y cada 10, tanto el análisis visual como la neguentropía empeoraron.

Se calculó posteriormente la frecuencia con la que el corredor pasaba por su eje vertical, con la intención de leer solo estos fotogramas y trazar tan solo la trayectoria y no el movimiento lateral que se produce en el desarrollo de esta actividad, que además es el más brusco y genera más irregularidades. El corredor resultó pasar por su eje vertical cada más o menos 21 fotogramas, por lo que se probó a calcular el flujo óptico

saltando los fotogramas de 20 en 20, de 21 de 21 y de 22 en 22. El resultado nuevamente no encaja del todo con lo esperado, si bien es cierto que se produce una pequeña mejora de la neguentropía de flujo leyendo los fotogramas de 21 en 21, este valor está por debajo del inicial, así que no se obtuvo mejora con respecto al inicio.

Estos malos resultados en el experimento I se deben principalmente a que el movimiento de una persona corriendo no es armónico, es decir, la frecuencia con la que se da un paso no es constante. Estas irregularidades hacen que sea imposible obtener una frecuencia exacta de paso por el eje vertical, además, al tratarse de un movimiento rápido, se pierde mucha información en los alrededores de 20 fotogramas que se descartan en cada iteración.

El segundo experimento se estructuró pensando en la aplicación de SLAM visual existente que no podía calcular el flujo en la secuencia corriendo. Esta aplicación no utiliza pirámides de imágenes por lo que se realizó este experimento para valorar en qué medida estas podían suponer una mejora en el seguimiento de los puntos en este caso. Tanto el análisis visual como la neguentropía mostraron claras mejoras aplicando las pirámides de imágenes.

Puede observarse en los resultados del experimento II que la neguentropía de flujo óptico calculada ante los movimientos suaves que se producen caminando, antes de añadir las pirámides, es mucho mejor que la calculada ante el movimiento corriendo, que es muy baja. Los resultados para ambas secuencias mejoran notablemente conforme se le añaden escalones a la pirámide hasta que se llega a una saturación debido a que la baja resolución de las imágenes no permite calcular el flujo con precisión. Este desorden inicial tan alto en el caso del video corriendo, indica un mal seguimiento de los puntos característicos, razón por la cual podría no funcionar el programa de SLAM mencionado, así que se puede pensar que añadiendo un cálculo piramidal de imágenes al método de *active search* que utiliza dicho programa se lograría completar el cálculo del mapeo y posicionamiento ante este tipo de movimiento.

Anexo I. Descripción de las funciones de OpenCV utilizadas

En este proyecto, como se ha explicado, se ha programado en lenguaje C++ y sirviéndose de las funciones de la biblioteca libre de visión por computador OpenCV, en el presente anexo se especifican las funciones utilizadas de dicha biblioteca y sus parámetros más característicos.

imread (filename, flags)

- **Descripción:** Carga una imagen es de un archivo del ordenador
- **Parámetros:**
 - filename:** nombre del archivo a cargar.
 - flag:** especifica el tipo de color de la imagen cargada:
CV_LOAD_IMAGE_GRAYSCALE: carga la imagen en blanco y negro.

imwrite(filename, img, params)

- **Descripción:** Guarda una imagen producto del programa en un archivo.
- **Parámetros:**
 - filename:** nombre del archivo.
 - image:** archivo a guardar.
 - params:** especifica la extensión del archivo.

goodFeaturesToTrack(image, corners, maxCorners, qualityLevel, minDistance, blockSize)

- **Descripción:** Selección los puntos característicos, corners, de la imagen.
- **Parámetros:**
 - image:** fotograma de entrada.
 - corners:** vector de salida con los puntos característicos detectados
 - maxCorners:** el número máximo de corners a devolver de la función.
 - qualityLevel:** parámetro que determina el mínimo aceptable para marcar un punto como corner.
 - minDistance:** mínima distancia posible entre dos corners devueltos.

cornerSubPix(image, corners, zeroZone, criteria)

- **Descripción:** Afina la localización de los corners.

- **Parámetros:**
 - image:** fotograma de entrada.
 - corners:** parámetro de entrada y salida, entra el vector de corners calculado con la función `goodFeaturesToTrack` y devuelve el vector de corners afinados.
 - criteria:** indica el criterio de refinamiento de la localización.

calcOpticalFlowPyrLK(prevImg, nextImg, prevPts, nextPts, status, winSize, maxLevel)

- **Descripción:** calcula el flujo óptico discreto para un conjunto de corners por el método de Lukas-Kanade, con o sin pirámides.
- **Parámetros:**
 - prevImg:** imagen o pirámide de imágenes de entrada.
 - nextImg:** segunda imagen e entrada con el mismo tamaño y del mismo tipo que la anterior.
 - prevPts:** vector 2D de entrada que contiene los corners de la primera imagen:
 - nextPts:** vector 2D de salida que contiene las nuevas posiciones de los corners de prevImg en nextImg.
 - status:** indica con el valor de 1 o de 0 si un punto ha sido localizado o no.
 - winsize:** tamaño de la ventana de búsqueda en cada nivel piramidal.
 - maxlevel:** indica el numero de niveles de la pirámide de imágenes a implementar, si es un 0 la pirámide solo tiene la base, no hay pirámide, si es un 1 la pirámide tiene dos escalones, y así sucesivamente.

Anexo II. Resultados completos experimento I

En el presente proyecto se han comentado los aspectos más destacados del experimento I, en este anexo se pueden ver todos los datos completos graficados. En este experimento se han obtenido también como resultado los fotogramas en cada caso con el flujo óptico trazado que se han utilizado para el análisis visual, estos datos no pueden adjuntarse debido al espacio y el volumen que ocupan

Test 1

Resultados de la neguentropía mínima para la variación de la frecuencia de lectura.

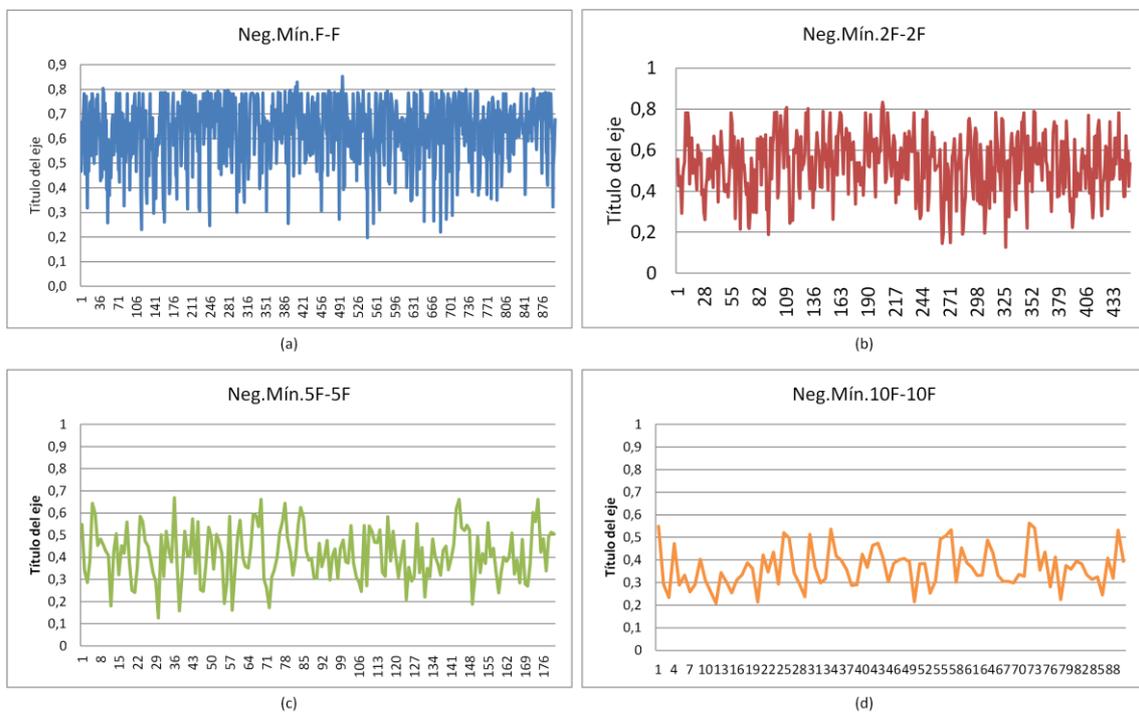


Figura II.1. Neguentropía mínima calculada para los casos en los que la frecuencia de lectura es: fotograma a fotograma(a), de dos en dos fotogramas (b), de cinco en cinco fotogramas(c) y de 10 en 10 fotogramas (d)

Test 2

Neguentropía mínima de las frecuencias de lectura resultantes del estudio del movimiento del corredor.

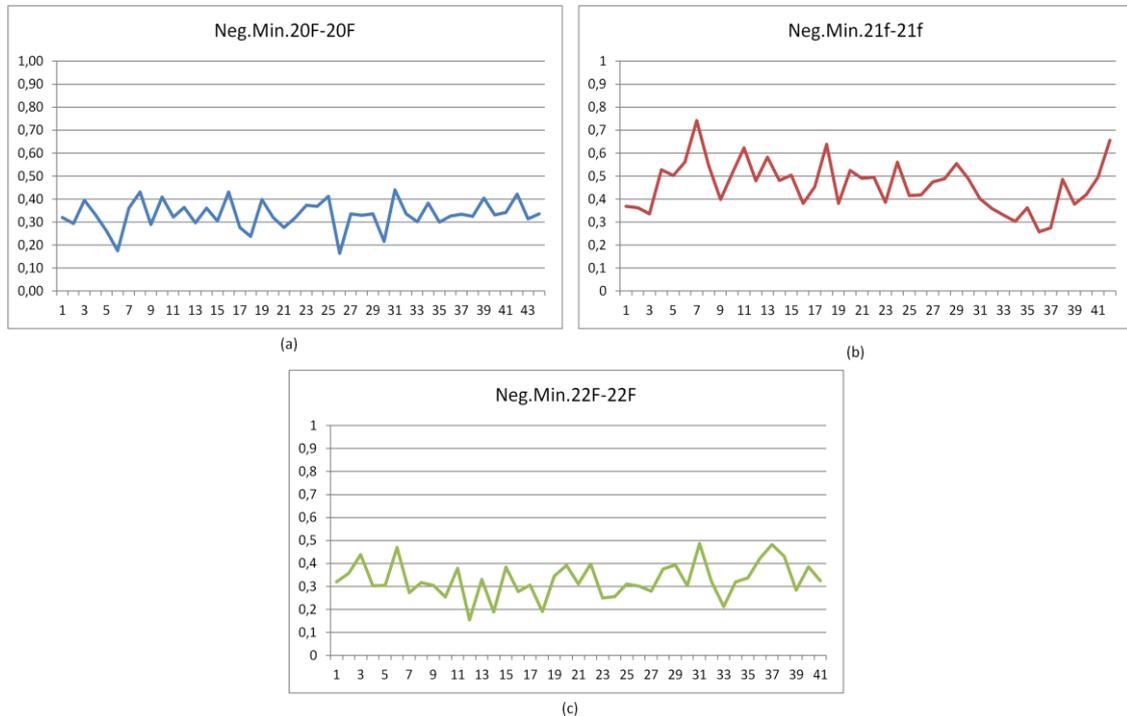
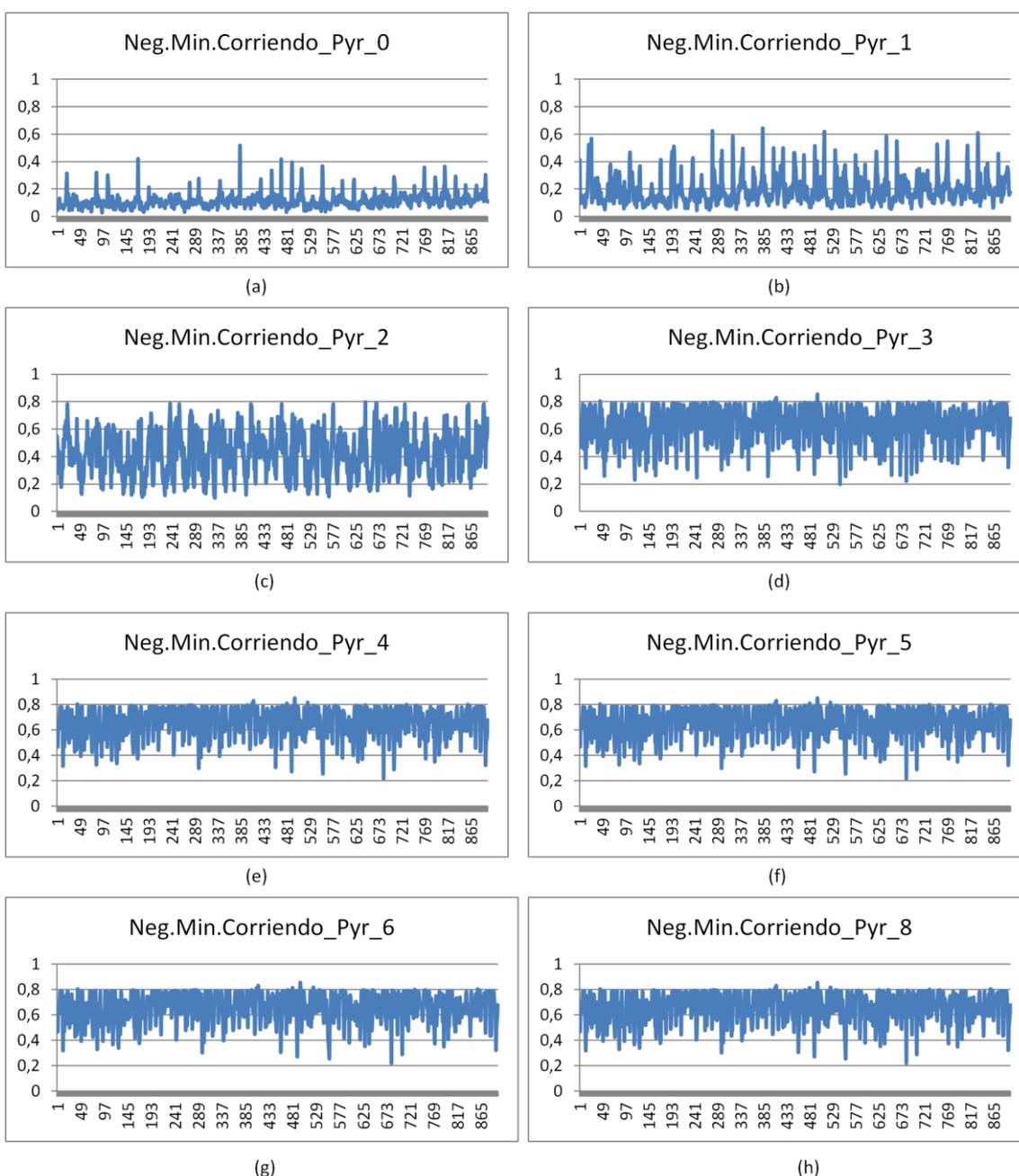


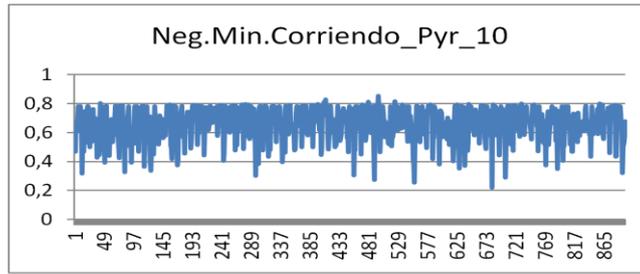
Figura II.2. Neguentropía mínima calculada para los casos en los que la frecuencia de lectura es: de 20 fotogramas en 20 fotogramas(a), de 21 fotogramas en 21 fotogramas (b), y de 22 en 22 fotogramas(c)

Anexo III. Resultados completos experimento II

En el apartado 4.2 del presente proyecto se presentan las conclusiones sacadas de los datos obtenidos del experimento II, en el presente anexo se pueden ver todos los datos. Al igual que en el experimento I, en el experimento II se han obtenido gran cantidad de fotogramas, 900 por cada ensayo, con el flujo óptico trazado que se han vuelto a montar en videos para el análisis visual, debido a la gran cantidad de espacio que ocupan no pueden adjuntarse en el presente proyecto asique se muestran las gráficas de la neguentropía mínima media en cada caso.

Neguentropía mínima media de las secuencias corriendo:

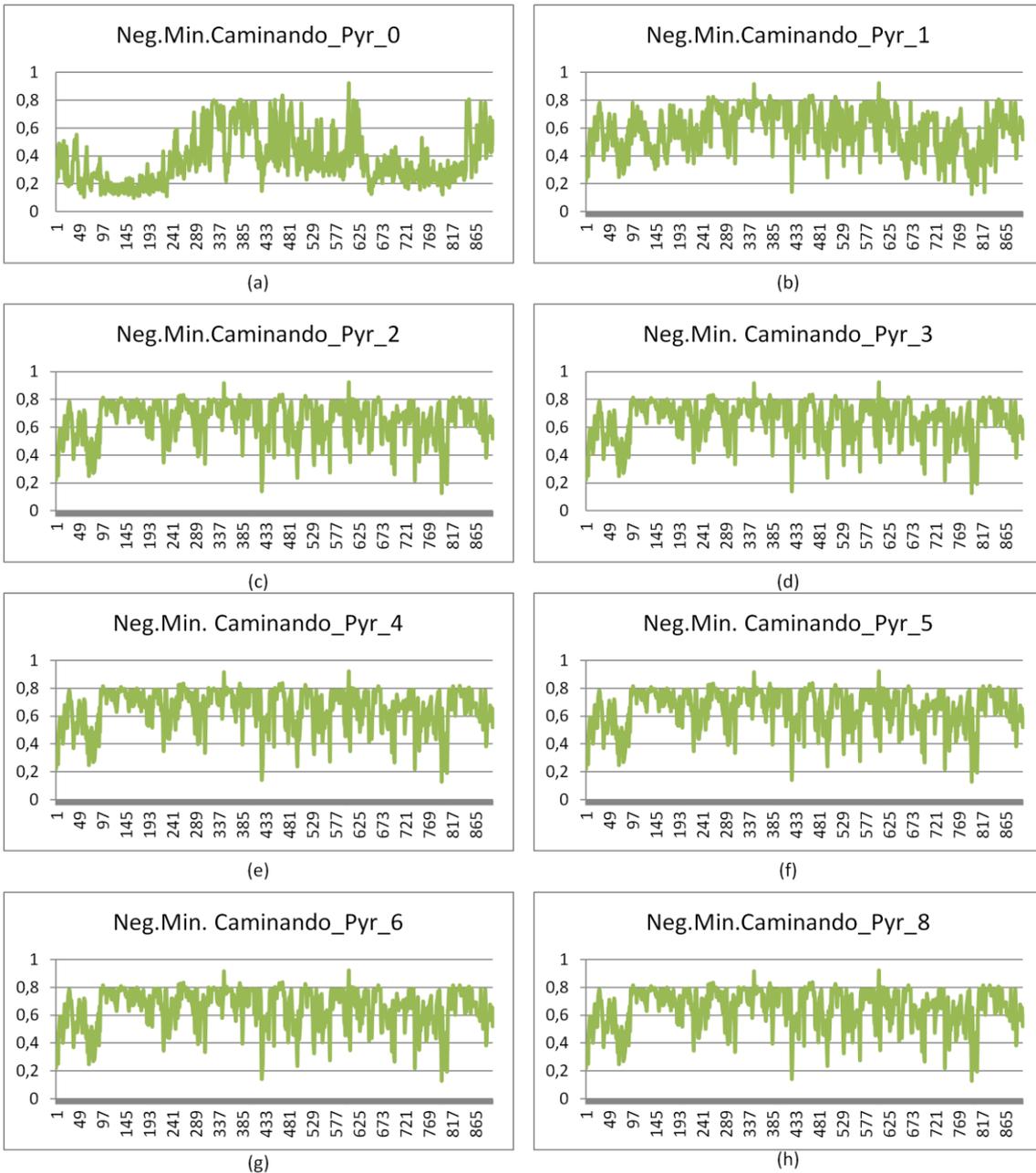


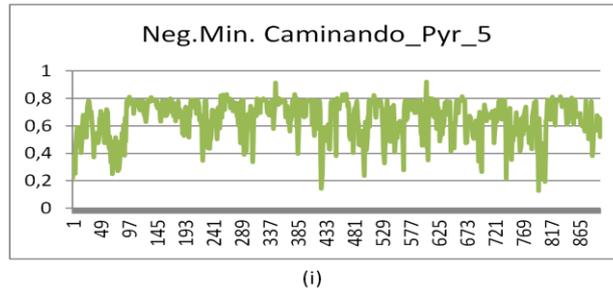


(i)

Figura III. 1. Valores de neguentropía mínimos obtenidos corriendo para un nivel piramidal de: 0 (a), 1(b), 2(c), 3(d), 4(e), 5 (f), 6 (g), 8 (h) y 10 (i).

Neguentropía mínima media de las secuencias caminando:





(i)
Figura III. 2. Valores de neguentropía mínimos obtenidos caminando para un nivel piramidal de: 0 (a), 1(b), 2(c), 3(d), 4(e), 5 (f), 6 (g), 8 (h) y 10 (i).

Referencias

- [1]. Itseez 2015 <http://opencv.org/about.html>
- [2]. Gutierrez D., Localización por visión omnidireccional para asistencia personal. Proyecto fin de carrera dirigido por Alejandro Rituerto y Jose J. Guerrero, Universidad de Zaragoza, 2011.
- [3]. A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In ICCV, 2003.
- [4]. Javier Civera, Andrew J. Davison, and J. M. M. Montiel. Inverse depth parametrization for monocular slam. *IEEE Transactions on Robotics*, 24(5):932–945, October 2008.
- [5]. J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel. 1-Point RANSAC for EKF Filtering: application to real-time structure from motion and visual odometry. *Journal of Field Robotics*, 27(5):609–631, October 2010.
- [6]. Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32:105–119, 2010.
- [7]. David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60:91–110, November 2004.
- [8]. Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110:346–359, June 2008.
- [9]. Marko Heikkilä, Matti Pietikainen, and Cordelia Schmid. Description of interest regions with local binary patterns. *Pattern Recognition*, 42(3):425–436, mar 2009.
- [10]. Andrew Burton and John Radford (1978). *Thinking in Perspective: Critical Essays in the Study of Thought Processes*
- [11]. David H. Warren and Edward R. Strelow (1985). *Electronic Spatial Sensing for the Blind: Contributions from Perception*
- [12]. Beauchemin, S., Barron, J. *The computation of optical flow* (1995)
- [13]. Berthold K.P. Horn and Brian G. Shunk. *Determining optical flow*. MIT. Cambridge. 1980
- [14]. B. Lucas and T. Kanade. *An iterative image registration technique with an application to stereo vision*, Vancouver, Canada, 1981
- [15]. Jean-Yves Bouguet. *Pyramidal implementation of the Lucas Kanade Feature Tracker*. Intel corporation.
- [16]. <http://es.gopro.com/>
- [17]. <https://es.wikipedia.org/wiki/Histograma>
- [18]. Daniel Gutierrez-Gomez y J.J Guerrero. True Scaled 6 DoF Egocentric Localisation with Monocular Wearable Systems. Universidad de Zaragoza. 2015.