

Tecnológico de Costa Rica  
Escuela de Ingeniería Electrónica



## **Diseño de un estimador de movimiento en secuencias de imágenes**

Informe de Proyecto de Graduación para optar por el título de  
Ingeniero en Electrónica con el grado académico de Licenciatura

Carlos Esteban Gómez Víquez

Cartago, 29 de noviembre, 2012



Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía he procedido a indicar las fuentes mediante las respectivas citas bibliográficas. En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.



---

Carlos Esteban Gómez Víquez

Cartago, 29 de noviembre, 2012

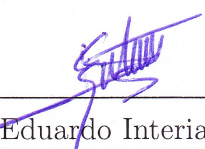
Céd: 1-1438-0221



Instituto Tecnológico de Costa Rica  
Escuela de Ingeniería Electrónica  
Proyecto de Graduación  
Tribunal Evaluador

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniera en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal



---

M.Sc. Eduardo Interiano  
Profesor Lector



---

M.Sc. Arys Carrasquilla Batista  
Profesora Lectora



---

Dr. Pablo Alvarado Moya  
Profesor Asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica.

Cartago, 29 de noviembre de 2012



# Resumen

En una aplicación para la creación de mosaicos en sistemas embebidos a partir de imágenes individuales capturadas por cuadricópteros, es necesario implementar un estimador de movimiento que permita acelerar la estimación de la transformación geométrica final de cada imagen utilizada en el mosaico. Para esto se propone el cálculo del flujo óptico de la secuencia de imágenes capturadas de manera que se pueda realizar un rastreo de puntos específicos de la imagen para determinar los coeficientes de la transformación geométrica que describen la rotación, el escalado y la traslación. Se llevará a cabo la implementación de dos algoritmos de flujo óptico clásicos, específicamente los métodos propuestos por Lucas-Kanade y Horn-Schunck, los cuales serán comparados con un tercer método de flujo óptico local multi-escala acelerado con imágenes integrales. Se utilizará un algoritmo genético para optimización de los métodos de flujo óptico, de manera que se presentarán frentes de Pareto con las combinaciones de parámetros con mejores características, evaluando tanto el tiempo de cálculo como la precisión de los resultados. Se logró reducir el tiempo de selección de puntos correspondientes en un 95.5% con respecto a un algoritmo de búsqueda exhaustiva de correspondencias para el caso de 1048 puntos de interés escogidos en una secuencia de imágenes de 320x240 píxeles.

**Palabras clave:** Estimador de movimiento, Flujo óptico, Lucas-Kanade, Horn-Schunck, Algoritmos genéticos, Frente de Pareto





# Abstract

In an application for mosaicing individual images captured by quadcopters, it is necessary to implement a motion estimation stage to accelerate the estimation of the final image geometrical transformations used in the mosaic. This work proposes the computation of the optical flow for a sequence of captured images in order to do a preliminary tracking of specific image locations that will be used in the final computation of the similarity transformation matrices. The implementation of two classical optical flow algorithms, specifically ones proposed by Lucas-Kanade and Horn-Schunck, serves as reference to compared with a third multiscale method for local optical flow accelerated with integral images. A genetic algorithm is used to find optimal parameterizations of the optical flow algorithms with respect to a set of fitness metrics such as the computation time and the accuracy of the results. It is possible to reduce the time of selection of correspondences points by 95.5% with respect to an exhaustive search algorithm of correspondences on a number of 1048 interest points taken of images of 320x240 pixels.

**Keywords:** Motion Estimator, Optical Flow, Lucas-Kanade, Horn-Schunck, Genetics Algorithms, Pareto Front



*a mis queridos padres*



# Agradecimientos

Agradezco principalmente a Dios por darme el entendimiento, la fortaleza y la sabiduría necesarias para enfrentar todos los retos y superar todas las dificultades que se me presentaron durante los años de mi vida dedicados a concluir el programa de estudios de ingeniería en electrónica en el Tecnológico de Costa Rica.

Al Dr. Pablo Alvarado Moya, por su apreciada ayuda y consejo durante la conclusión de mis estudios y en la realización de este trabajo.

A mis padres Juan Carlos y Edith, por su gran amor y apoyo incondicional que me han permitido alcanzar grandes metas para continuar ofreciendo siempre lo mejor de mi persona.

Y a todas aquellas personas que de una u otra forma han contribuido con mi formación a lo largo de todos estos años.

Carlos Esteban Gómez Víquez

Cartago, 29 de noviembre, 2012



# Índice general

Índice de figuras	iii
Índice de tablas	v
Lista de símbolos y abreviaciones	vii
<b>1 Introducción</b>	<b>1</b>
1.1 Objetivos y estructura del documento . . . . .	3
<b>2 Marco Teórico</b>	<b>5</b>
2.1 Flujo Óptico . . . . .	5
2.1.1 Método de Horn-Schunck . . . . .	6
2.1.2 Método de Lucas-Kanade . . . . .	8
2.1.3 Evaluación . . . . .	9
2.2 Imágenes Integrales . . . . .	10
2.3 Puntos Característicos . . . . .	10
2.4 Búsqueda de correspondencias . . . . .	11
2.5 Matriz de transformación . . . . .	11
2.6 Gradiente . . . . .	11
2.6.1 Derivadas de gaussianas . . . . .	11
2.6.2 Operador de Ando . . . . .	12
2.6.3 Método de Sobel . . . . .	12
2.6.4 Diferencias finitas . . . . .	12
2.7 Frente de Pareto . . . . .	12
<b>3 Estimador de movimiento en secuencias de imágenes</b>	<b>15</b>
3.1 Algoritmo de flujo óptico propuesto . . . . .	16
<b>4 Resultados y Análisis</b>	<b>19</b>
4.1 Evaluación del algoritmo de flujo óptico propuesto . . . . .	20
4.2 Evaluación del algoritmo Lucas-Kanade . . . . .	24
4.3 Evaluación del algoritmo Horn-Schunck . . . . .	25
4.4 Evaluación general . . . . .	27
<b>5 Conclusiones</b>	<b>35</b>

5.1 Recomendaciones . . . . .	36
<b>Bibliografía</b>	<b>37</b>
A Secuencias de imágenes	41
Índice alfabético	43



# Índice de figuras

2.1	Ejemplo de sumatorias con imágenes integrales . . . . .	10
2.2	Kernel utilizado para obtener el gradiente con el método Sobel: (a) para la componente $x$ (b) para la componente $y$ . . . . .	12
3.1	Diagrama del estimador de movimiento en secuencias de imágenes . . . . .	15
4.1	Diagrama del error de punto final EP para un punto de interés $p_i$ . . . . .	19
4.2	Frente de Pareto del algoritmo de flujo óptico propuesto . . . . .	21
4.3	Flujo óptico de la secuencia de imágenes de la figura A.1 con el algoritmo propuesto: (a) de la primera a la segunda, (b) de la segunda a la tercera, (c) de la tercera a la cuarta, (d) de la cuarta a la quinta, (e) de la quinta a la sexta, (f) de la sexta a la séptima, (g) de la séptima a la octava. Parámetros utilizados: $w = 57$ , $\tau = 0,3812$ , $maxIterations = 47$ , $numLevels = 2$ , $maxError = 0,0023$ , $gradientKernelType = OGD$ . . . . .	23
4.4	Frente de Pareto de la mediana y nivel de agrupación de las mediciones de precisión del algoritmo de flujo óptico propuesto . . . . .	24
4.5	Frente de Pareto del algoritmo de flujo óptico Lucas-Kanade . . . . .	25
4.6	Flujo óptico de la secuencia de imágenes de la figura A.1 con el algoritmo Lucas-Kanade: (a) de la primera a la segunda, (b) de la segunda a la tercera, (c) de la tercera a la cuarta, (d) de la cuarta a la quinta, (e) de la quinta a la sexta, (f) de la sexta a la séptima, (g) de la séptima a la octava. Parámetros utilizados: $variance = 9.98924732$ , $gradientKernelSize = 5$ , $kernelSize = 9$ , $gradientKernelType = OGD$ . . . . .	26
4.7	Frente de Pareto de la mediana y nivel de agrupación de las mediciones de precisión del algoritmo de flujo óptico Lucas-Kanade . . . . .	27
4.8	Frente de Pareto del algoritmo de flujo óptico Horn-Schunk . . . . .	28
4.9	Flujo óptico de la secuencia de imágenes de la figura A.1 con el algoritmo Horn-Schunk: (a) de la primera a la segunda, (b) de la segunda a la tercera, (c) de la tercera a la cuarta, (d) de la cuarta a la quinta, (e) de la quinta a la sexta, (f) de la sexta a la séptima, (g) de la séptima a la octava. Parámetros utilizados: $lambda = 0.361504644$ , $maxError = 0.0170940179$ , $maxIterations = 3$ , $variance = 9.60215092$ , $kernelSize = 3$ , $gradientKernelType = OGD$ . . . . .	29

4.10 Frente de Pareto de la mediana y nivel de agrupación de las mediciones de precisión del algoritmo de flujo óptico Horn-Schunk . . . . .	30
4.11 Comparación de frentes de Pareto de los algoritmos de flujo óptico . . . . .	33
A.1 Secuencia de imágenes: (a) es la imagen inicial, (b) es una rotación de $1^\circ$ en el sentido de las manecillas del reloj, (c) es una rotación de $3^\circ$ en el sentido contrario de las manecillas del reloj, (d) es una traslación en (x,y) de (2,-1) píxeles, (e) es otra traslación de (-3,5) píxeles, (f) es una traslación larga de (7,4) píxeles, (g) es un escalado de 1,02 (2%) y (h) es otro escalado de 0,95 (-5%). Todas las transformaciones son respecto a la imagen anterior y se utilizó un interpolador cúbico. . . . .	41
A.2 Secuencia de imágenes: (a) es la imagen inicial, (b) es una rotación de $2,5^\circ$ en el sentido de las manecillas del reloj, (c) es un escalado de 1,04 (4%), (d) es una rotación de $3,5^\circ$ en el sentido contrario de las manecillas del reloj, (e) es otro escalado de 0,95 (-5%). Todas las transformaciones son respecto a la imagen anterior y se utilizó un interpolador cúbico. . . . .	42
A.3 Secuencia de imágenes: (a) es la imagen inicial, (b) es una traslación de (2,-1) píxeles, (c) es un escalado de 1,01 (1%), (d) es una rotación de $0,5^\circ$ en el sentido de las manecillas del reloj. Todas las transformaciones son respecto a la imagen anterior y se utilizó un interpolador cúbico. . . . .	42

# Índice de tablas

2.1	Valores error absoluto (EP) promedio para cada una de las seis secuencias y tiempo estimado con respecto a la secuencia Urban . . . . .	9
4.1	Valores destacados del frente de Pareto del algoritmo propuesto . . . . .	22
4.2	Evaluación de los algoritmos de rastreo de puntos para la secuencia de la figura A.1 . . . . .	31
4.3	Evaluación de los algoritmos de rastreo de puntos para la secuencia de la figura A.3 . . . . .	31
4.4	Evaluación del algoritmo de flujo óptico propuesto para diferentes parametrizaciones y secuencias de imágenes . . . . .	32
4.5	Comparación de resultados entre el algoritmo de flujo óptico propuesto como referencia y los clásicos para la secuencia de la figura A.1 . . . . .	32



# Lista de símbolos y abreviaciones

## Abreviaciones

EA	Error angular
EP	Error de punto final o absoluto

## Notación general

**A** Matriz.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

$\mathbb{C}$  Conjunto de los números complejos.

$\text{Im}(z)$  o  $z_{\text{Im}}$  Parte imaginaria del número complejo  $z$

$j$   $j = \sqrt{-1}$

$\text{Re}(z)$  o  $z_{\text{Re}}$  Parte real del número complejo  $z$

$\mathcal{T}[\cdot]$  Transformación realizada por un sistema

$\underline{\mathbf{x}}$  Vector.

$$\underline{\mathbf{x}} = [x_1 \ x_2 \ \dots \ x_n]^T = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$y$  Escalar.

$z^*$  Complejo conjugado de  $z$



# Capítulo 1

## Introducción

En los últimos años se han desarrollado investigaciones alrededor de los vehículos aéreos no tripulados (UAV-Unmanned Aerial Vehicle) [21], esto con el fin de realizar tareas donde el uso de vehículos tripulados es peligroso o de difícil acceso, tales como la monitorización de incendios forestales, misiones de rescate y evaluación de riesgos por desastres naturales; también pueden aplicarse para fines más comerciales como la monitorización de fincas extensas dedicadas a la producción agrícola donde es necesario evaluar los sectores más afectados por plagas de insectos, hongos o bacterias; así como las condiciones de humedad del suelo. Por otra parte, también es posible utilizarlos en labores de carácter militar donde el espionaje y el reconocimiento de terreno cobran importancia.

Para cubrir todas las áreas de aplicación lo más común es equipar los UAV con cámaras sensibles a las longitudes de onda que no necesariamente son para capturar luz visible, sino que también pueden ser infrarrojas en cualquiera de sus bandas electromagnéticas para aplicaciones como creación de mapas de biomasa, humedad del suelo, vegetación, mapas térmicos, mapas de minerales, entre otras [16]. La utilización de estos dispositivos podría extenderse al usar otras bandas del espectro electromagnético desde los rayos gamma hasta las ondas de radio.

En un proyecto desarrollado en conjunto entre el Tecnológico de Costa Rica y la Universidad de Buenos Aires, se quiere reconstruir toda una superficie de terreno a partir de imágenes individuales capturadas mediante cámaras transportadas por cuadracópteros quienes recorren todo el terreno que se quiere monitorizar. Una vez capturadas las imágenes, se procede con la creación del mosaico. Sin embargo, se debe tomar en cuenta que el vehículo que transporta la cámara sufre perturbaciones como lo son leves pérdidas de altura y rotaciones, velocidades de avance variables e inclinaciones.

Estas perturbaciones causan alteraciones en el eje óptico de la cámara de manera que la fusión no puede llevarse a cabo sin antes realizar las correcciones necesarias para lograr que todas las imágenes estén sobre el mismo plano de referencia. Es necesario implementar un estimador de movimiento para reducir los tiempos de selección de puntos correspondientes en la secuencia de imágenes capturadas de manera que sea posible identificar, a partir de

dichos puntos, los sistemas lineales que describen las transformaciones de una imagen a otra en cuanto a escalado, rotación y traslación; y así poder estimar un sistema coordinado global para realizar la fusión de las imágenes.

En este proyecto, el estimador de movimiento va a optimizar el tiempo requerido para la selección de puntos característicos en cada imagen y la búsqueda de los puntos correspondientes dentro de la secuencia de imágenes, esto con la ayuda de técnicas de visión por computador, en particular el flujo óptico, para predecir la ubicación de los puntos característicos y reducir las áreas de búsqueda. Se debe tomar en cuenta que los algoritmos requeridos, ya sean de visión por computador o de procesamiento digital de imágenes, tanto para la selección de los puntos característicos como sus correspondientes en la secuencia de imágenes, son iterativos y basados en relaciones probabilísticas donde el objetivo es minimizar el error, por lo que reducir el número de comparaciones y las áreas de cálculo reducen considerablemente el tiempo de procesamiento. La reducción de la complejidad computacional es un requisito crítico puesto que el algoritmo completo para la creación de mosaicos de la superficie monitorizada se ejecutará sobre un sistema embebido incluido dentro del cuadracóptero.

La solución del problema se presenta en varias etapas, la primera es la identificación de puntos característicos los cuales normalmente son esquinas de las figuras existentes en las imágenes, para esto en la teoría de procesamiento de imágenes se han propuesto diversos métodos para la identificación de puntos característicos, por ejemplo el detector de esquinas Harris [24] se basa en el hecho de que las esquinas están asociadas a valores máximos de la función de auto correlación local; sin embargo en este proyecto se utiliza otro método basado en cálculos sobre la matriz Hessiana [6].

Una vez identificados los puntos de interés se deben rastrear dentro de la secuencia de imágenes, esto es identificar en la siguiente captura donde se encuentra dicho punto característico. Para esto se utilizan aproximaciones de asociación de puntos característicos por medio de correlaciones e interpolaciones que permitan encontrar los puntos correspondientes de los puntos característicos de una imagen en la siguiente captura [25]. Este es un proceso iterativo en el cual a cada par de puntos de interés entre dos imágenes se le asigna una probabilidad de correspondencia. El proceso termina hasta lograr una probabilidad de correspondencia global significativamente alta que está dentro de un margen aceptable definido por el usuario [23].

Se pretende mejorar considerablemente el tiempo de búsqueda implementando una etapa de cálculo del flujo óptico [5] que permita reducir el área de búsqueda de puntos correspondientes. El cálculo del flujo óptico se realizará mediante algún algoritmo basado en los métodos clásicos como Horn-Schunck [19] y Lucas-Kanade [20]. El objetivo del flujo óptico es tener conocimiento de la trayectoria de cada píxel a lo largo de la secuencia de imágenes capturadas. De esta manera se tiene el conocimiento de la ubicación de cada píxel entre dos imágenes y es posible determinar las correspondencias de los puntos de interés.



## 1.1 Objetivos y estructura del documento

Este proyecto tiene como objetivo desarrollar un estimador de movimiento en secuencias de imágenes que permita reducir los tiempos de selección de puntos correspondientes; reduciendo un 50% el tiempo con respecto a una búsqueda exhaustiva de correspondencias. Para esto se pretende evaluar el desempeño teórico en cuanto a precisión y velocidad de al menos tres algoritmos de flujo óptico; presentando diversos trabajos relacionados con el flujo óptico y un sondeo de dichos algoritmos. Además, se quiere implementar en C++ al menos dos algoritmos de flujo óptico capaces de rastrear puntos característicos; donde la ejecución de los algoritmos coincide en un 90% con alguna implementación equivalente de referencia. Y finalmente diseñar una estrategia de rastreo de puntos característicos basado en el flujo óptico que alcance un 70% de selección correcta de puntos característicos con respecto a un detector de correspondencias exhaustivo.

En el siguiente capítulo se esbozan los fundamentos teóricos necesarios para explicar en el capítulo 3 la propuesta realizada. En el capítulo 4 se muestran algunos resultados estadísticos del algoritmo de flujo óptico propuesto junto a dos métodos clásicos (Lucas-Kanade, Horn-Schunck) así como su análisis con referencia a un algoritmo de búsqueda exhaustiva de correspondencias. En el capítulo 5 se ofrecen las principales conclusiones, algunos aportes y recomendaciones para trabajos futuros.



# Capítulo 2

## Marco Teórico

### 2.1 Flujo Óptico

El *flujo óptico* es una aproximación del movimiento definido como una proyección de las velocidades reales de los puntos de una escena en el plano de la imagen [7]. Es decir, es una estimación del movimiento proyectado en las dos dimensiones de la imagen como velocidades instantáneas o desplazamientos discretos en la secuencia de imágenes capturada.

Existen diversas propuestas para la estimación de flujo óptico. La mayoría se basan en algoritmos diferenciales clásicos como *Horn-Schunck* [19] y *Lucas-Kanade* [20] que lo calculan a partir de las derivadas espacio-temporales. Por ejemplo, en [10] se expone una modificación del cálculo de flujo óptico que generaliza la técnica de Horn-Schunck para brindarle más inmunidad ante el ruido al combinarla con el método local de Lucas-Kanade. En [9] se ofrece un algoritmo que permite acelerar el cálculo de flujo óptico global utilizando una estructura multi-rejilla bidireccional que reduce considerablemente la cantidad de iteraciones necesarias en la resolución de los sistemas de ecuaciones del cálculo de variaciones que minimizan la función de error.

El método explicado en [22] realiza una modificación en la función de error del algoritmo local Lucas-Kanade utilizando proyecciones integrales de las componentes del vector de flujo óptico con el objetivo de disminuir el costo computacional. Propuestas como [17] utilizan ecuaciones diferenciales parciales de cuarto orden en un algoritmo que combina los métodos global y local de flujo óptico con el fin de mejorar la precisión y suavidad en la estimación de movimiento.

Otras propuestas no se basan en métodos diferenciales sino que se fundamentan en características como la frecuencia mediante filtros sensibles a la orientación en el dominio de Fourier [15]. También se han propuesto métodos que utilizan correlación de fase para determinar un conjunto de puntos donde el movimiento es más representativo y sobre el cual realizar una búsqueda de correspondencias para definir el desplazamiento [2].

Para calcular el flujo óptico con métodos diferenciales se establece inicialmente la restricción de intensidad de gris o brillo contante, de manera que entre dos imágenes consecutivas el brillo de cada uno de los puntos de los objetos sea igual. Definiendo  $I(x, y, t)$  como el brillo de un píxel cualquiera en un tiempo  $t$  dado, esta restricción dice que idealmente en la imagen consecutiva el brillo del mismo píxel pero desplazado y capturada un tiempo  $t + \Delta t$  después debe ser exactamente igual, es decir

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

Ahora suponiendo los desplazamientos espaciales y temporal diferenciales se obtiene la expresión

$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$

De esta forma es posible aproximar los desplazamientos diferenciales por medio de series de Taylor, donde  $I_x, I_y, I_t$  son las derivadas parciales de  $I(x, y, t)$

$$I(x, y, t) \approx I(x, y, t) + I_x dx + I_y dy + I_t dt$$

Finalmente se desarrolla la expresión

$$\begin{aligned} I_x dx + I_y dy + I_t dt &\approx 0 \\ I_x \frac{dx}{dt} + I_y \frac{dy}{dt} + I_t &\approx 0 \\ I_x u + I_y v + I_t &\approx 0 \end{aligned} \tag{2.1}$$

Definiendo  $u = \frac{dx}{dt}$  y  $v = \frac{dy}{dt}$  como las componentes del vector de velocidad  $\underline{c}$ . Entonces se obtiene que (2.1) es la ecuación del flujo óptico con el problema de tener dos incógnitas (las componentes de  $\underline{c}$ ) y solamente una ecuación, también conocido como el problema de *apertura*. Como consecuencia se han desarrollado diferentes propuestas para solucionar este problema incluyendo nuevas restricciones.

### 2.1.1 Método de Horn-Schunck

Para resolver el problema de apertura Horn y Schunck propusieron [19] añadir la restricción de suavidad, es decir, en una vecindad de píxeles dada de algún objeto se supone que las velocidades son similares por lo que el campo de velocidad de los patrones de brillo de la imagen varían con suavidad. Como medida de la suavidad de flujo óptico se puede utilizar el cuadrado de la magnitud del gradiente de velocidad

$$\begin{aligned} |\nabla u|^2 &= \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial u}{\partial y} \right)^2 = u_x^2 + u_y^2 \\ |\nabla v|^2 &= \left( \frac{\partial v}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 = v_x^2 + v_y^2 \end{aligned}$$

También es posible utilizar como medida de la suavidad la suma de los laplacianos de las componentes del vector  $\underline{c}$  definidos como:

$$\Delta u = \nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

$$\Delta v = \nabla^2 v = \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}$$

En este caso se utilizará como indicador de suavidad la suma de las magnitudes del gradiente al cuadrado. Entonces el problema radica en encontrar el vector  $\underline{c}$  que de acuerdo con (2.1) minimice el error  $E_b(u, v) = I_x u + I_y v + I_t$  y además minimice la medida de suavidad  $E_c^2(u, v) = |\nabla u|^2 + |\nabla v|^2$ , de esta manera se establece el error total en un área  $\Omega$  como:

$$E = \iint_{\Omega} (I_x u + I_y v + I_t)^2 + \alpha (|\nabla u|^2 + |\nabla v|^2) d\Omega \quad (2.2)$$

donde  $\alpha$  es una constante de regularización. Para realizar la minimización de (2.2) se recurrirá a las ecuaciones de Euler-Lagrange del cálculo de variaciones, utilizadas ampliamente en problemas de optimización. Entonces, definiendo el error como

$$E = \iint_{\Omega} L(x, y, u, v, u_x, u_y, v_x, v_y) d\Omega$$

las ecuaciones de Euler-Lagrange son:

$$\begin{cases} \frac{\partial L}{\partial u} - \frac{\partial}{\partial x} \frac{\partial L}{\partial u_x} - \frac{\partial}{\partial y} \frac{\partial L}{\partial u_y} = 0 \\ \frac{\partial L}{\partial v} - \frac{\partial}{\partial x} \frac{\partial L}{\partial v_x} - \frac{\partial}{\partial y} \frac{\partial L}{\partial v_y} = 0 \end{cases} \quad (2.3)$$

Desarrollando las ecuaciones se obtiene:

$$\begin{cases} \frac{1}{\alpha} (I_x^2 u + I_x I_y v + I_x I_t) - \Delta u = 0 \\ \frac{1}{\alpha} (I_x I_y u + I_y^2 v + I_y I_t) - \Delta v = 0 \end{cases} \quad (2.4)$$

Ahora realizando la aproximación  $\Delta u = \kappa(\bar{u} - u)$  y  $\Delta v = \kappa(\bar{v} - v)$  donde  $\bar{u}$  y  $\bar{v}$  son los promedios que pueden ser obtenidos al realizar la convolución de las señales con un filtro gaussiano. Entonces sustituyendo esta aproximación en (2.4) y acomodando las ecuaciones se obtiene

$$\begin{cases} (I_x^2 + \alpha) u + I_x I_y v = \alpha \bar{u} - I_x I_t \\ I_x I_y u + (I_y^2 + \alpha) v = \alpha \bar{v} - I_y I_t \end{cases} \quad (2.5)$$

que pueden ser resueltas de forma iterativa por medio del método Gauss-Seidel como:

$$\begin{cases} u^{k+1} = \frac{\alpha \bar{u}^k - I_x I_t - I_x I_y v^k}{I_x^2 + \alpha} \\ v^{k+1} = \frac{\alpha \bar{v}^k - I_y I_t - I_x I_y u^{k+1}}{I_y^2 + \alpha} \end{cases} \quad (2.6)$$

donde el superíndice  $k$  denota en este caso el número de iteración, o bien, como propone [19] utilizando únicamente los valores promedio:

$$\begin{cases} u^{k+1} = \bar{u}^k + I_x \frac{I_x \bar{u}^k + I_y \bar{v}^k + I_t}{\alpha^2 + I_x^2 + I_y^2} \\ v^{k+1} = \bar{v}^k + I_y \frac{I_x \bar{u}^k + I_y \bar{v}^k + I_t}{\alpha^2 + I_x^2 + I_y^2} \end{cases} \quad (2.7)$$

Por otra parte, tratándose de una secuencia de imágenes es posible reducir el costo computacional según [23] inicializando la secuencia con el primer par de imágenes normalmente con (2.7) y luego resolviendo para cada imagen  $l$  de la secuencia

$$\begin{cases} u^{l+1} = \bar{u}^l + I_x \frac{I_x \bar{u}^l + I_y \bar{v}^l + I_t}{\alpha^2 + I_x^2 + I_y^2} \\ v^{l+1} = \bar{v}^l + I_y \frac{I_x \bar{u}^l + I_y \bar{v}^l + I_t}{\alpha^2 + I_x^2 + I_y^2} \end{cases} \quad (2.8)$$

donde el gradiente espacial y temporal corresponde al de la última imagen evaluada.

### 2.1.2 Método de Lucas-Kanade

La ecuación (2.1) se puede escribir como

$$\begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -I_t$$

de manera que si se asume una vecindad de  $n$  puntos con vectores de velocidad  $u, v$  constantes expresada como

$$\underbrace{\begin{bmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \vdots & \vdots \\ I_{xn} & I_{yn} \end{bmatrix}}_A \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\vec{U}} = - \underbrace{\begin{bmatrix} I_{t1} \\ I_{t2} \\ \vdots \\ I_{tn} \end{bmatrix}}_B$$

entonces se obtiene el vector  $\vec{U}$  de la siguiente forma:

$$\begin{aligned} A\vec{U} &= B \\ A^T A\vec{U} &= A^T B \\ \vec{U} &= (A^T A)^{-1} A^T B \end{aligned} \quad (2.9)$$

donde

$$A^T A = \begin{bmatrix} \sum_{i=1}^n I_{xi}^2 & \sum_{i=1}^n I_{xi} I_{yi} \\ \sum_{i=1}^n I_{xi} I_{yi} & \sum_{i=1}^n I_{yi}^2 \end{bmatrix}$$

$$A^T B = - \begin{bmatrix} \sum_{i=1}^n I_{xi} I_{ti} \\ \sum_{i=1}^n I_{yi} I_{ti} \end{bmatrix}$$

Este método ofrece buenos resultados cuando la región de puntos sobre la cual se aplica tiene una textura apreciable; sin embargo, para regiones homogéneas la matriz  $A^T A$  es aproximadamente cero y en los bordes dicha matriz es singular, por lo que los resultados podrían ser incorrectos. Por otra parte, cuando el movimiento no es pequeño, cuando un punto no se mueve como sus vecinos o cuando el brillo no es constante también se inducen altos niveles de error.

### 2.1.3 Evaluación

En este proyecto se utilizará como medida de rendimiento el error absoluto o error de punto final (EP) definido como  $\sqrt{(u_{ref} - u_1)^2 + (v_{ref} - v_1)^2}$  donde se compara el error del vector de flujo óptico  $\underline{c} = (u_1, v_1)$  con respecto a algún vector de referencia  $(u_{ref}, v_{ref})$ . Por supuesto, en una secuencia determinada se realizarán los cálculos estadísticos básicos como el promedio, la mediana, la varianza, y los valores máximo y mínimo; de manera que represente en términos generales el desempeño del algoritmo de flujo óptico utilizado.

En [4] se ofrece una metodología de evaluación para flujo óptico, donde además de calcular EP también determinan el error angular (AE) como el coseno inverso del producto punto entre el vector a evaluar y el de referencia, normalizados. En la tabla 2.1 se muestra la evaluación con respecto al promedio del error absoluto (EP) de algunos algoritmos de flujo óptico analizados con la metodología de [4] disponibles en su página web [1] para diferentes secuencias diseñadas por los mismos autores. Se incluye el tiempo en procesar la secuencia “Urban”, aunque no se tienen datos del sistema computacional utilizado sirven como referencia entre los mismos métodos para determinar qué interesa más: si el tiempo de procesamiento o la precisión.

<i>Método</i> \ <i>Secuencias</i>	<i>Army</i>	<i>Schefflera</i>	<i>Wooden</i>	<i>Grove</i>	<i>Urban</i>	<i>Yosemite</i>	<i>Tiempo(s)</i>
Pyramid LK	0.39	1.50	1.57	2.94	3.33	0.30	12
Horn&Schunck	0.22	1.01	0.78	1.26	1.43	0.16	49
Black&Anandan	0.18	0.95	0.49	1.08	1.43	0.15	328
PGAM+LK	0.37	0.94	1.40	1.37	2.10	0.36	0.37
Second-order prior	0.11	0.57	0.20	0.94	0.61	0.20	14

**Tabla 2.1:** Valores error absoluto (EP) promedio para cada una de las seis secuencias y tiempo estimado con respectos a la secuencia Urban

## 2.2 Imágenes Integrales

Las *imágenes integrales*  $I_{\Sigma}(x, y)$  [12] son una forma de representar un canal determinado  $I(x, y)$  de intensidad de gris por ejemplo, como:

$$I_{\Sigma}(x, y) = \sum_{i=0}^x \sum_{j=0}^y I(i, j)$$

por lo que realizar la sumatoria de los valores en una región rectangular determinada en la imagen solamente es necesario realizar cuatro sumas con los vértices de la dicha región, por ejemplo en la figura 2.1 la sumatoria del área  $I_{\Sigma}$  es igual a

$$I_{\Sigma} = D - B - C + A$$

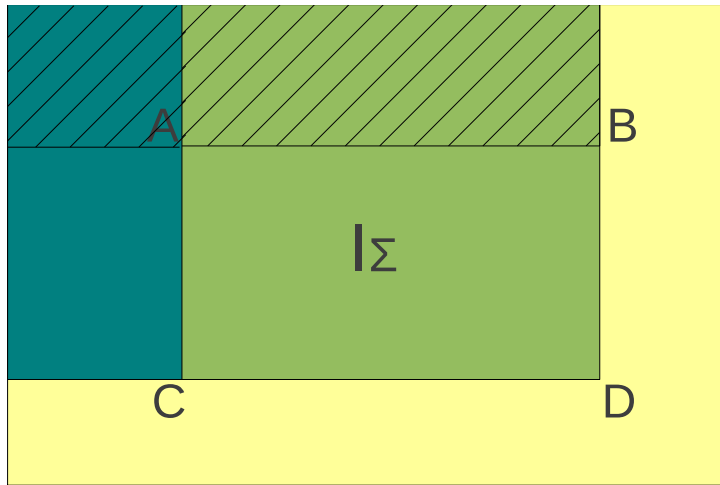


Figura 2.1: Ejemplo de sumatorias con imágenes integrales

## 2.3 Puntos Característicos

Uno de los detectores de *puntos característicos* más comunes es el *detector de Harris* [18]. Sin embargo propuestas más recientes [6] se basan en la *matriz Hessiana*

$$H(x, y, \sigma) = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{xy}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{bmatrix} \quad (2.10)$$

donde  $L_{xx}(x, y, \sigma)$  es la convolución de la segunda derivada de la Gaussiana  $\frac{\partial^2}{\partial x^2}g(\sigma)$  con la imagen  $I(x, y)$  en el punto  $(x, y)$ . La idea es encontrar estructuras burbuja (vecindades circulares al rededor de un punto de interés) en localidades donde el determinante de la matriz Hessiana es máximo. La ventaja más notable de la misma es su precisión.



## 2.4 Búsqueda de correspondencias

El algoritmo de *búsqueda exhaustiva de correspondencias* que se utilizará como referencia, está basado en la clase “locationMatch“ de la LTI-Lib 2 [13] que encuentra la mejor asociación entre los puntos de interés de un par de imágenes a partir de descriptores de la distribución de la intensidad contenida en una vecindad de un punto de interés según [6]. Los puntos característicos se obtienen de la clase “fastHessianDetection” y los descriptores de la clase “surfLocalDescriptor” que igualmente son parte de la LTI-Lib 2.

## 2.5 Matriz de transformación

A partir de las correspondencias entre las imágenes en una secuencia, es posible determinar una matriz de transformación que brinde información acerca de las traslaciones en los ejes coordenados  $(t_x, t_y)$ , el ángulo de rotación  $(\alpha)$ , el escalado  $(s)$  en el caso de estar frente a una *transformación geométrica* de similitud

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} s \cos \alpha & s \sin \alpha & t_x \\ -s \sin \alpha & s \cos \alpha & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \quad (2.11)$$

En la LTI-Lib 2 la clase “ransacEstimation” implementa el algoritmo RANSAC [14] que a partir de dos vectores de puntos correspondientes estima una relación lineal entre ellos de acuerdo con los parámetros de la matriz de transformación especificada, por ejemplo en el caso de transformación de similitud se determinan los valores de las traslaciones en los ejes coordenados  $(t_x, t_y)$ , el ángulo de rotación  $(\alpha)$  y el escalado  $(s)$ .

## 2.6 Gradiente

En imágenes discretas, el gradiente solo se puede aproximar. Se presentan a continuación algunos métodos de aproximación empleados en este trabajo.

### 2.6.1 Derivadas de gaussianas

Este método de estimación del gradiente tiene como principal ventaja la reducción de ruido mediante el filtrado con una máscara Gaussiana. Sea la función  $o(x, y)$  el *gradiente* de la imagen  $i(x, y)$  y  $g(x, y, \sigma)$  la función gaussiana

$$g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

entonces,

$$o(x, y) = \nabla(g(x, y, \sigma) * i(x, y)) = i(x, y) * \nabla g(x, y, \sigma)$$

Donde la máscara  $\nabla g(x, y, \sigma)$  se calcula como  $\nabla g = \left[ \frac{\partial g}{\partial x}, \frac{\partial g}{\partial xy} \right]$  con

$$\frac{\partial g}{\partial x} = -\frac{x}{\sigma^2} g(x, y, \sigma)$$

$$\frac{\partial g}{\partial y} = -\frac{y}{\sigma^2} g(x, y, \sigma)$$

y  $*$  es el operador de convolución lineal.

### 2.6.2 Operador de Ando

Este operador de estimación de gradiente está optimizado para invarianza a la rotación de acuerdo con las recomendaciones en [3], y cuyo criterio de diseño se basa en una descomposición ortogonal de la diferencia entre el gradiente continuo y discreto.

### 2.6.3 Método de Sobel

Esta es una aproximación del método de Derivadas de Gaussianas utilizando un filtro binomial como el de la figura 2.2

+1	0	-1	+1	+2	+1
+2	0	-2	0	0	0
+1	0	-1	-1	-2	-1
(a)			(b)		

**Figura 2.2:** Kernel utilizado para obtener el gradiente con el método Sobel: (a) para la componente  $x$  (b) para la componente  $y$ .

### 2.6.4 Diferencias finitas

Esta es una de las aproximaciones por diferencias más sencillas y por tanto rápida, donde el gradiente es calculado como:

$$\frac{\partial}{\partial x} f(x, y) = \frac{1}{2} (f(x+1, y) - f(x-1, y))$$

$$\frac{\partial}{\partial y} f(x, y) = \frac{1}{2} (f(x, y+1) - f(x, y-1))$$

## 2.7 Frente de Pareto

El *frente de Pareto* es un concepto de la economía que a partir de una asignación inicial de bienes optimiza la situación de un individuo. Se utiliza para analizar diferentes opciones

con características óptimas de un individuo dada una variedad de objetivos y criterios de evaluación. Por ejemplo, en una compra de algún artículo con variedad de precios y calidades un comprador con un presupuesto limitado desearía obtener el artículo de mejor calidad al precio que pueda pagar.

En ingeniería existen algoritmos genéticos de optimización tales como Pareto Envelope-Based Selection Algorithm (PESA) [11] implementado en la clase “paretoFront” de la LTI-Lib 2 y provee un mecanismo de evaluación de algoritmos ante diferentes parametrizaciones. PESA es un algoritmo evolutivo que genera combinaciones aleatorias de parámetros con el fin de obtener los mejores rendimientos del algoritmo que dominen sobre una población que al final evolucionará a un frente de Pareto de las combinaciones de parámetros con las medidas de rendimiento óptimas.



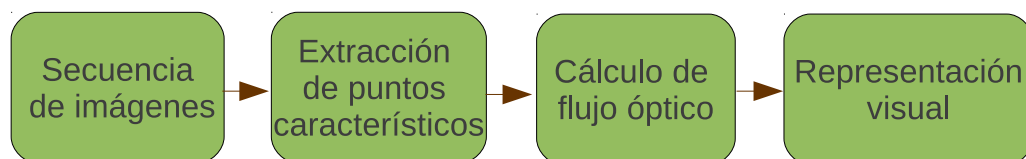
# Capítulo 3

## Estimador de movimiento en secuencias de imágenes

Se pretende obtener un algoritmo que permita realizar un rastreo de píxeles a lo largo de una secuencia de imágenes de manera que conociendo las posiciones iniciales y finales de los objetos en la secuencia de imágenes se pueda estimar una matriz de transformación geométrica que describa el movimiento. En este trabajo se utilizarán únicamente *transformaciones de similitud* donde los movimientos de interés serán traslaciones, rotaciones y escalado.

En el diagrama de la figura 3.1 se pueden observar las distintas etapas que conforman el estimador de movimiento basado en flujo óptico. Inicialmente se tiene la etapa de lectura de la secuencia de imágenes ya sea desde una cámara de vídeo con módulos de kernel Video4Linux2 API o bien desde archivos de imagen almacenados en memoria.

Para disminuir el costo computacional de operar sobre todos los píxeles de una imagen se requiere de un módulo de extracción de puntos característicos que obtenga solamente un conjunto reducido de píxeles sobre los cuales realizar la estimación de movimiento, esta etapa fue implementada con el método de “Fast Hessian Detection” por ser una de las propuestas mas recientes con características de velocidad y precisión adecuadas.



**Figura 3.1:** Diagrama del estimador de movimiento en secuencias de imágenes

En cuanto a la etapa de cálculo de flujo óptico, se utilizará un método local derivado del método Lucas-Kanade. En el capítulo 2 se explicó la implementación del método clásico de Lucas-Kanade con un enfoque en mínimos cuadrados, sin embargo en este trabajo se

utilizará un enfoque iterativo que permita mejorar la precisión del flujo óptico y a pesar de que resulta en un mayor costo computacional se compensa este hecho mediante el uso de imágenes integrales. En la sección 3.1 se muestra en detalle el algoritmo de flujo óptico propuesto.

Finalmente en la etapa de visualización se dispone en la LTI-Lib 2 de una clase que permite dibujar vectorialmente las componentes del vector de velocidad mediante flechas con la magnitud y ángulo de dicho vector. También es posible representarlo mediante códigos de color predefinido donde la intensidad está asociada a la magnitud y el matiz a la fase. Por otra parte, en términos numéricos es posible visualizar los parámetros de traslación, rotación y escalado de la matriz de transformación correspondiente.

### 3.1 Algoritmo de flujo óptico propuesto

Partiendo del concepto de flujo óptico y de la suposición del método Lucas-Kanade de que los puntos dentro de una vecindad tienen vector de velocidad  $\underline{\mathbf{c}} = (u, v)$  constante, se puede definir una función de error dependiente del vector de velocidad  $\underline{\mathbf{c}}$  como

$$E(\underline{\mathbf{c}}) = \sum_{\Omega} [I_1(\underline{\mathbf{x}} + \underline{\mathbf{c}}) - I_2(\underline{\mathbf{x}})]^2$$

donde  $\Omega$  representa la vecindad de puntos como una ventana de dimensiones  $\eta_x \times \eta_y$  y  $\underline{\mathbf{x}} = (x, y)$  es el vector de posición de algún píxel dentro de  $\Omega$  en la imagen  $I_1$  que tiene un punto correspondiente en la siguiente imagen  $I_2$  de la secuencia que ha sido desplazado de acuerdo con  $\underline{\mathbf{c}}$ . Idealmente se quiere que la función de error sea cero, de forma que la estimación de  $\underline{\mathbf{c}}$  indique una correspondencia exacta, por lo tanto dicho vector debe cumplir con la expresión  $f(\underline{\mathbf{c}}) = \frac{\partial E}{\partial \underline{\mathbf{c}}} = 0$  que se desarrolla como:

$$\frac{\partial E}{\partial \underline{\mathbf{c}}} = \frac{\partial}{\partial \underline{\mathbf{c}}} \sum_{\Omega} [I_1(\underline{\mathbf{x}} + \underline{\mathbf{c}}) - I_2(\underline{\mathbf{x}})]^2 = 0$$

Ahora aplicando la expansión de Taylor de primer orden a  $I_1(\underline{\mathbf{x}} + \underline{\mathbf{c}})$  se obtiene que es aproximadamente  $I_1(\underline{\mathbf{x}}) + \underline{\mathbf{c}} \frac{\partial I_1}{\partial \underline{\mathbf{x}}}$  entonces se obtiene que:

$$\frac{\partial}{\partial \underline{\mathbf{c}}} \sum_{\Omega} \left[ I_1(\underline{\mathbf{x}}) + \underline{\mathbf{c}} \frac{\partial I_1}{\partial \underline{\mathbf{x}}} - I_2(\underline{\mathbf{x}}) \right]^2 = 0$$

$$\sum_{\Omega} 2 \left[ I_1(\underline{\mathbf{x}}) + \underline{\mathbf{c}} \frac{\partial I_1}{\partial \underline{\mathbf{x}}} - I_2(\underline{\mathbf{x}}) \right] \frac{\partial I_1}{\partial \underline{\mathbf{x}}} = 0$$

Para resolver la ecuación anterior se decide utilizar el método de Newton-Rhapson, por lo tanto:

$$\underline{\mathbf{c}}_{k+1} = \underline{\mathbf{c}}_k - \frac{f(\underline{\mathbf{c}})}{f'(\underline{\mathbf{c}})}$$

$$\begin{aligned}
\underline{\mathbf{c}}_{k+1} &= \underline{\mathbf{c}}_k - \frac{\sum_{\Omega} \left[ I_1(\underline{\mathbf{x}}) + \underline{\mathbf{c}}_k \frac{\partial I_1}{\partial \underline{\mathbf{x}}} - I_2(\underline{\mathbf{x}}) \right] \frac{\partial I_1}{\partial \underline{\mathbf{x}}}}{\sum_{\Omega} \left[ \frac{\partial I_2}{\partial \underline{\mathbf{x}}} \right]^2} \\
\underline{\mathbf{c}}_{k+1} &= \underline{\mathbf{c}}_k + \frac{\sum_{\Omega} \nabla I_1(\underline{\mathbf{x}})^T [I_2(\underline{\mathbf{x}}) - I_1(\underline{\mathbf{x}}) - \nabla I_1(\underline{\mathbf{x}}) \underline{\mathbf{c}}_k]}{\sum_{\Omega} \nabla I_1(\underline{\mathbf{x}}) \nabla I_1(\underline{\mathbf{x}})^T} \\
\underline{\mathbf{c}}_{k+1} &= \underline{\mathbf{c}}_k + \frac{\sum_{\Omega} \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_t - I_x u_k - I_y v_k]}{\sum_{\Omega} I_x I_x + I_y I_y} \\
\begin{bmatrix} u_{k+1} \\ v_{k+1} \end{bmatrix} &= \begin{bmatrix} u_k \\ v_k \end{bmatrix} + \frac{\sum_{\Omega} \begin{bmatrix} I_x I_t - I_x I_x u_k - I_x I_y v_k \\ I_y I_t - I_y I_x u_k - I_y I_y v_k \end{bmatrix}}{\sum_{\Omega} I_x I_x + I_y I_y} \tag{3.1}
\end{aligned}$$

Se puede observar que de una forma iterativa con inicialización  $\underline{\mathbf{c}} = 0$  es posible obtener el flujo óptico a partir de las derivadas espaciales  $I_x, I_y$  y la derivada temporal  $I_t$  con respecto a la imagen  $I_1$ . La derivada temporal se aproxima como la diferencia de la intensidad de gris entre  $I_1$  e  $I_2$  y para el cálculo del gradiente es posible estimarlo por medio de filtrado con máscaras tales como la de “Derivadas de Gaussianas” (DoG), Sobel que utiliza un filtro binomial y aproxima a DoG, el operador de Ando que está optimizado para invarianza a rotaciones, entre otros.

Por otra parte, se debe recordar que este algoritmo es iterativo y que para cada píxel se debe calcular (3.1) ya sea hasta que el error esté por debajo de un umbral definido o hasta alcanzar un máximo de iteraciones establecido. Por lo tanto, realizar las sumatorias en la ventana  $\Omega$  hasta lograr la convergencia resulta costoso computacionalmente. Como una forma de acelerar este cálculo se deciden implementar imágenes integrales para los productos  $I_x I_t, I_y I_t, I_x I_x, I_y I_y, I_x I_y$  aprovechando la propiedad de linealidad de la sumatoria. De esta forma se reduce la sumatoria de  $\eta_x \times \eta_y$  elementos a solamente la suma de cuatro puntos correspondientes a los vértices de la ventana  $\Omega$ .

Además, debido a la suposición del flujo óptico de que los desplazamientos son diferenciales, en la aplicación existirán problemas cuando se estime algún desplazamiento de por ejemplo tres píxeles o más; los errores de magnitud se verán incrementados. Por esta razón es necesario implementar *estructuras piramidales* o multiescala de manera que los desplazamientos largos de los niveles más finos se verán cortos en los niveles más gruesos. Entonces, realizando estimaciones gruesas inicialmente e ir afinando el cálculo al descender de nivel en la pirámide es posible lograr una buena estimación con desplazamientos grandes. De acuerdo con [8], si inicialmente un método tiene un alcance de longitud de

$d_{max}$ , utilizando estructuras piramidales se incrementa el alcance hasta  $(2^{L_m+1} - 1)d_{max}$ , donde  $L_m$  es el número de niveles de la pirámide.

Se puede observar que se han realizado varias suposiciones, desde las hechas en la definición del flujo óptico y las del método Lucas-Kanade, hasta las aproximaciones hechas tales como la expansión de Taylor de primer orden y los métodos numéricos necesarios para la implementación de (3.1) y demás modificaciones realizadas con el fin de mejorar las propiedades del algoritmo propuesto. Es de esperar que el error en la estimación pueda subir principalmente en la magnitud de los vectores de flujo óptico, por lo tanto se introduce un parámetro que funcionaría como una constante de regularización ( $\tau$ ) con el fin de ofrecer la oportunidad de afinar el algoritmo para alguna aplicación en específico. De esta manera, (3.1) quedaría finalmente como:

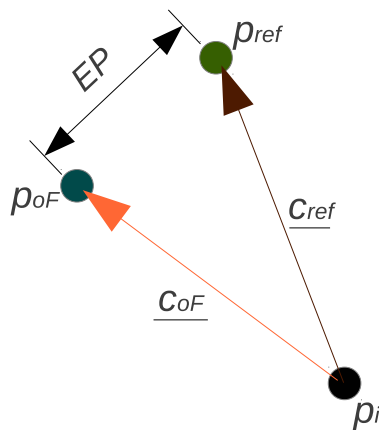
$$\begin{bmatrix} u_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} u_k \\ v_k \end{bmatrix} + \tau \frac{\sum_{\Omega} \begin{bmatrix} I_x I_t - I_x I_x u_k - I_x I_y v_k \\ I_y I_t - I_y I_x u_k - I_y I_y v_k \end{bmatrix}}{\sum_{\Omega} I_x I_x + I_y I_y} \quad (3.2)$$



# Capítulo 4

## Resultados y Análisis

Para comprobar el funcionamiento correcto del sistema se realizarán evaluaciones comparativas del sistema diseñado con un algoritmo de búsqueda exhaustiva por correspondencias. A partir de un conjunto de puntos característicos se utilizan las clases “location-Match” y “ransacEstimation” de la LTI-Lib 2 para obtener una matriz de transformación que se utiliza para generar los vectores de flujo óptico a considerar como “verdad absoluta” y por lo tanto se utilizarán como referencia. En la figura 4.1 se muestra un diagrama del error EP para un punto de interés  $p_i$  con punto correspondiente  $p_{ref}$  de acuerdo con el vector de referencia  $\underline{c_{ref}}$  calculado con dicho algoritmo, y con punto correspondiente  $p_{oF}$  según el resultado del vector de flujo óptico obtenido,  $\underline{c_{oF}}$ .



**Figura 4.1:** Diagrama del error de punto final EP para un punto de interés  $p_i$

Además se utilizó un algoritmo de optimización genético donde los parámetros de los algoritmos de flujo óptico implementados son codificados en una cadena de valores binarios llamados “cromosomas”. A partir de la selección aleatoria entre mutaciones y cruces entre pares de cromosomas se generan nuevos elementos que representan un nuevo conjunto de parámetros que una vez obtenidos se evalúan en el algoritmo. La evaluación utiliza

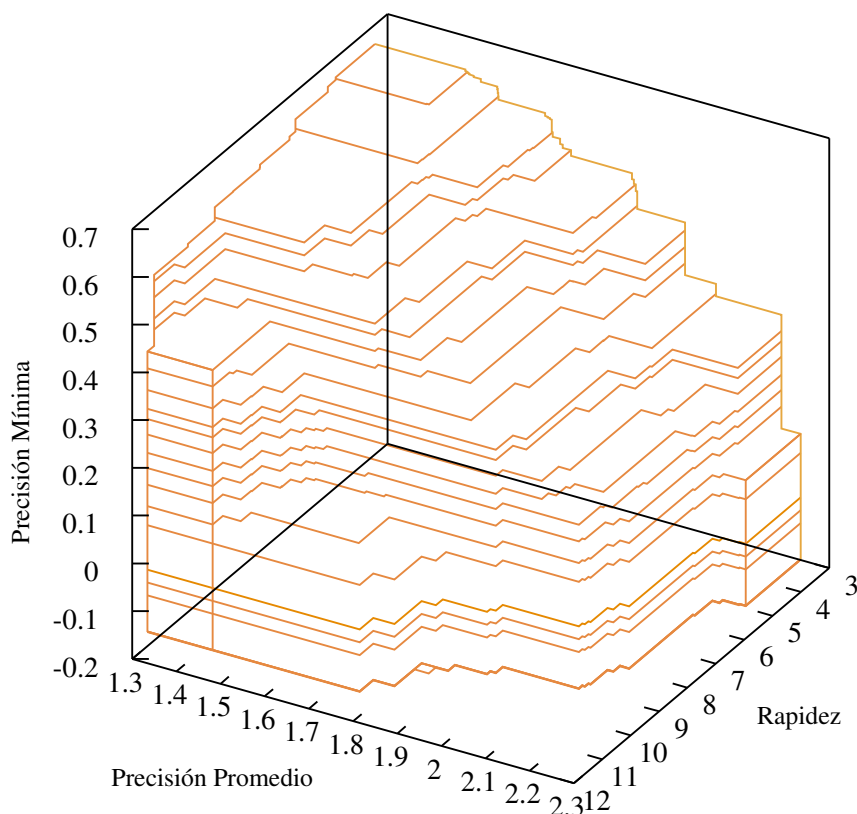
un conjunto de secuencias de imágenes con diferentes características de manera que a partir de los resultados estadísticos del error de punto final o absoluto tales como el promedio, la varianza, la media y los puntos mínimos o máximos se toman decisiones de si el cromosoma generado es un candidato fuerte para ser parte de un frente de Pareto que indicará cuáles son las mejores combinaciones de parámetros que logran las mejores características en precisión y tiempo. De esta manera, este proceso de generación de cromosomas, su evaluación y la selección permite la evolución de las combinaciones de parámetros que optimizan el algoritmo de flujo óptico. En este trabajo se tomaron en cuenta 1000 generaciones para la evolución de cada algoritmo de flujo óptico, resultando en aproximadamente 23 horas de ejecución del algoritmo de optimización.

En este capítulo se presentan los resultados estadísticos ofrecidos a partir del algoritmo de optimización, de manera que inicialmente se presentarán de manera resumida los frentes de Pareto obtenidos para cada uno de los algoritmos de estimación de flujo óptico implementados. Una vez analizados los parámetros con mejores características tanto de tiempo como de error absoluto se realizará una comparación entre dichos métodos y siempre tomando como referencia el método de búsqueda exhaustiva de correspondencias.

## 4.1 Evaluación del algoritmo de flujo óptico propuesto

En la figura 4.2, se muestra el frente de Pareto obtenido de la optimización del algoritmo de flujo óptico propuesto con imágenes integrales como acelerador y estructuras piramidales para mejorar el alcance en términos longitudinales. Para esto se utilizó la secuencia de imágenes de la figura A.1 que comprende rotaciones de uno y tres grados, escalados de 2% y -5%, así como traslaciones horizontales y verticales de hasta 7 píxeles. En este caso se obtuvo un frente de Pareto de 145 individuos que fueron seleccionados entre diez mil posibles combinaciones de los parámetros del algoritmo de flujo óptico aplicadas a dicha secuencia. Los valores numéricos de la figura son aptitudes, es decir entre mayor es el valor mejor es la parametrización asociada. Las aptitudes tomadas en cuenta son la rapidez y la precisión. En el caso de la rapidez  $r(t)$  se calcula a partir del tiempo de CPU  $t$  en  $\mu s$  que se tarda en ejecutar el algoritmo de flujo óptico, la expresión utilizada es  $r(t) = \frac{5 \times 10^5}{4t + 1} - \frac{1}{2}$ . La multiplicación por cuatro del tiempo  $t$  es porque se midieron cuatro ejecuciones completas del algoritmo debido a que la resolución del reloj del sistema operativo es de 1/60Hz, por lo que las mediciones cercanas a 16ms son de baja confiabilidad. Las demás operaciones son meramente ajustes de escala y para evitar la división entre cero. En el caso de las mediciones estadísticas de precisión  $p$ , se obtuvieron del error de punto final EP mediante la función  $p(EP) = \frac{5}{EP + 0.1} - \frac{1}{2}$  que tiene la forma similar a la relación utilizada para la rapidez.

En la tabla 4.1 se muestran algunos puntos tomados de la figura 4.2. Se puede observar



**Figura 4.2:** Frente de Pareto del algoritmo de flujo óptico propuesto

inicialmente que los primeros tres puntos corresponden a los que tuvieron el valor más alto en la evaluación de rapidez que como es de esperar su costo resulta ser un incremento en los valores promedio y máximos del error, por ejemplo, para una rapidez de 11,64 equivalente a 10.30ms tiene un promedio de error de 2,45 píxeles y un error máximo de 5.28 píxeles. Por el contrario, se nota una disminución en los valores de rapidez cuando se toman los mejores resultados del promedio y el mínimo de la precisión. Además, se observa que no se obtienen buenos resultados en cuanto a valor mínimo de la precisión cuando se consideran los puntos con los mejores valores promedio, en este caso el valor promedio de error más bajo obtenido es 1,7 píxeles o 2,27 como valor de precisión, mientras que el valor máximo de error (precisión mínima) es de 8,10 píxeles que resulta ser bastante alto. También cabe destacar que los mejores resultados de rapidez están asociados con una máscara para el gradiente de tipo Diferencias Finitas que resulta ser una aproximación bastante simple que evidentemente se refleja como un cálculo rápido pero con un nivel de error más elevado. Sin embargo, los mejores resultados en cuanto al promedio y mínimo de la precisión se dan para la máscara tipo OGD o Derivadas de Gaussianas que ofrece una estimación más precisa; otro tipo de máscara para el gradiente es el operador Ando que tan solo se utilizó en cinco individuos del frente de Pareto.

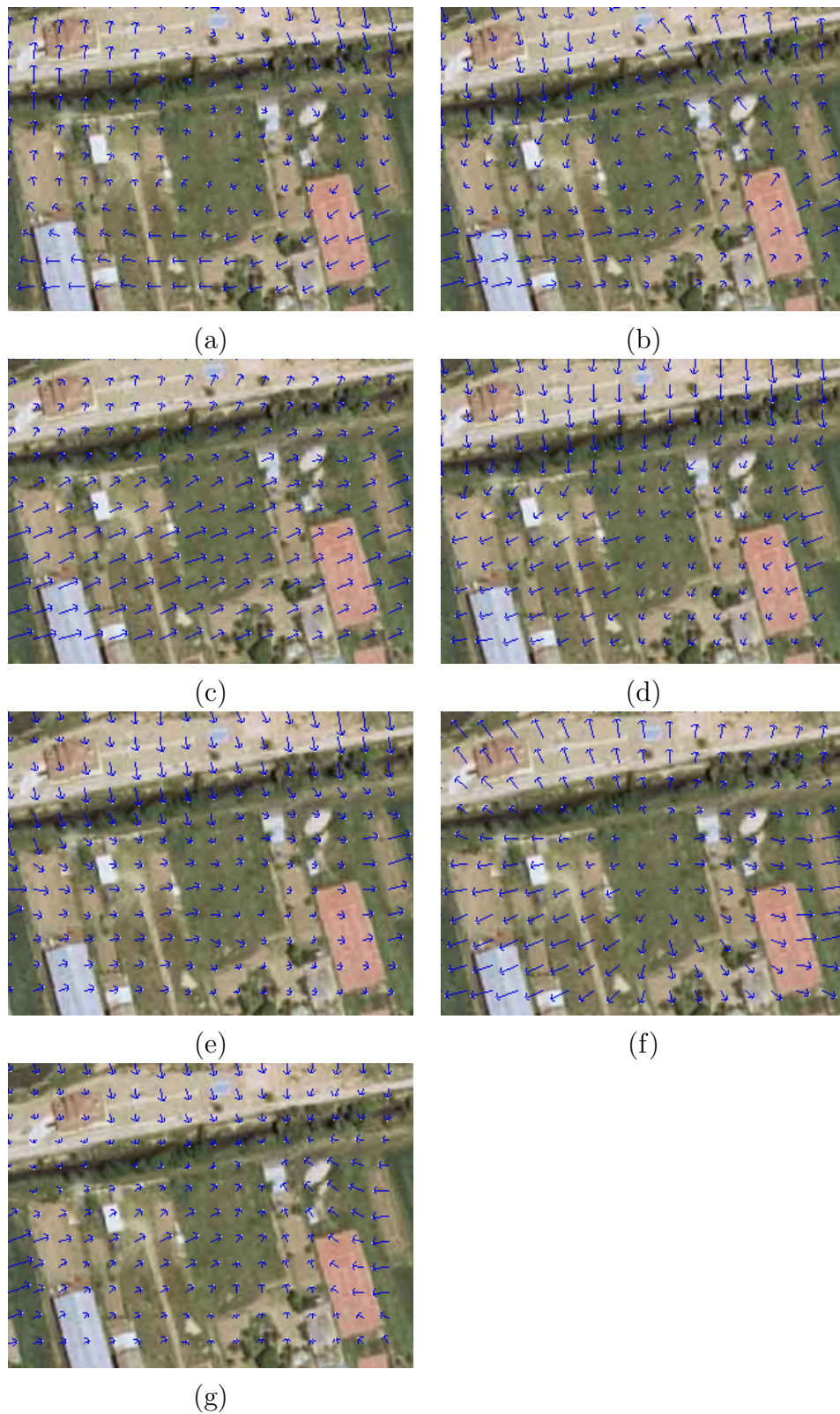
En la figura 4.3 se muestra la estimación de movimiento de cada imagen con respecto a la anterior, para visualizar el comportamiento del flujo óptico. Para esto se utilizó una combinación de parámetros de uno de los puntos de la figura 4.2 que presenta un tiempo promedio de 6,26 o 18,49ms con un error promedio de 2,00 píxeles con un error máximo

<i>Rapidez</i>	<i>Precisión Promedio</i>	<i>Precisión Mínima</i>	<i>Parámetros</i>	
11,4	1,31	0,58	w=58 maxIterations=16 maxError=0,0068	$\tau = 1, 5835$ numLevels=1 gradientKernel=Difference
11,64	1,46	0,43	w=58 maxIterations=51 maxError=0,0048	$\tau = 2, 1114$ numLevels=1 gradientKernel=Difference
11,28	1,50	0,36	w=54 maxIterations=33 maxError=0,0071	$\tau = 2, 2580$ numLevels=1 gradientKernel=Difference
5,55	2,27	0,11	w=60 maxIterations=51 maxError=0,0022	$\tau = 0, 2932$ numLevels=3 gradientKernel=OGD
9,50	2,16	0,1846	w=58 maxIterations=62 maxError=0,0029	$\tau = 1, 407$ numLevels=1 gradientKernel=OGD
9,61	2,01	0,28	w=59 maxIterations=20 maxError=0,0053	$\tau = 1, 20$ numLevels=1 gradientKernel=OGD
5,00	1,52	0,67	w=62 maxIterations=96 maxError=0,002	$\tau = 0, 1466$ numLevels=3 gradientKernel=OGD
6,41	1,59	0,65	w=62 maxIterations=51 maxError=0,00041	$\tau = 0, 2932$ numLevels=2 gradientKernel=OGD
7,66	1,54	0,62	w=63 maxIterations=33 maxError=0,0071	$\tau = 0, 76$ numLevels=1 gradientKernel=OGD

**Tabla 4.1:** Valores destacados del frente de Pareto del algoritmo propuesto

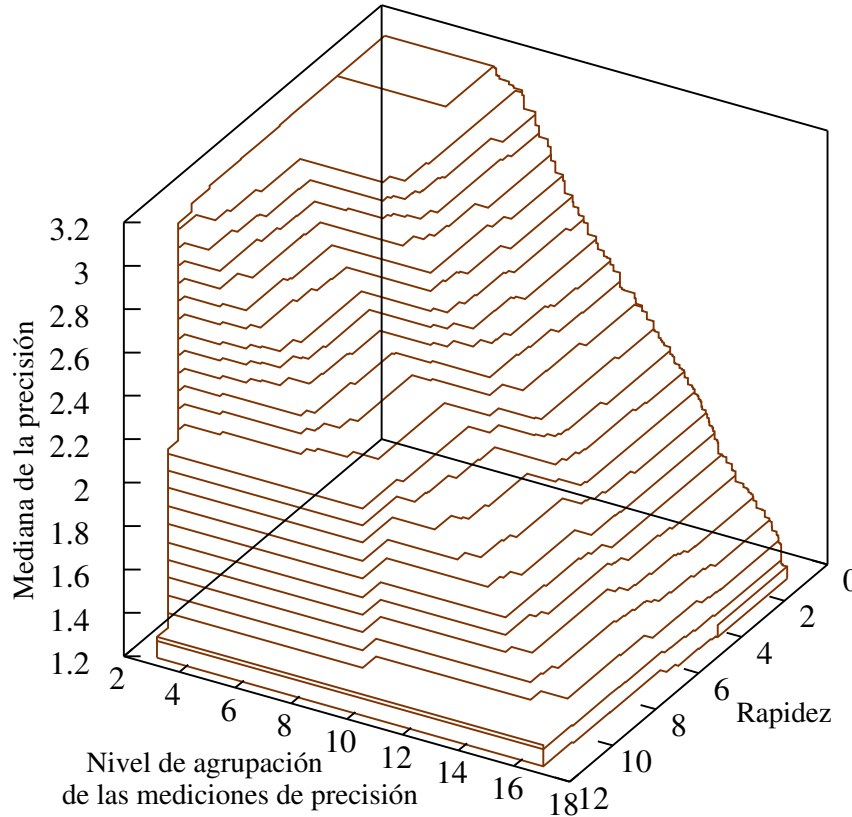
de 5, 23 píxeles.

Para complementar los resultados obtenidos con el frente de Pareto de la figura 4.2 se realizó otro experimento con el algoritmo de optimización en el cual se utilizaron la mediana y el nivel de agrupación de las mediciones de precisión como resultados de un nuevo frente de Pareto mostrado en la figura 4.4. En este caso la población es de 300 combinaciones de los parámetros del algoritmo de flujo óptico propuesto. Se puede observar que los valores de rapidez y la mediana de la precisión tienen rangos similares a la rapidez y precisión promedio de la figura 4.2, respectivamente. Además, se presenta un comportamiento semejante al discutido anteriormente donde los mejores resultados en cuanto a rapidez resultan en un compromiso por sacrificar el nivel de precisión en la estimación del flujo óptico. Por el contrario, mientras mejores son los resultados en cuanto a la mediana de la precisión, la rapidez se ve reducida y la dispersión de las mediciones de error se



**Figura 4.3:** Flujo óptico de la secuencia de imágenes de la figura A.1 con el algoritmo propuesto: (a) de la primera a la segunda, (b) de la segunda a la tercera, (c) de la tercera a la cuarta, (d) de la cuarta a la quinta, (e) de la quinta a la sexta, (f) de la sexta a la séptima, (g) de la séptima a la octava. Parámetros utilizados:  $w = 57$ ,  $\tau = 0,3812$ ,  $maxIterations = 47$ ,  $numLevels = 2$ ,  $maxError = 0,0023$ ,  $gradientKernelType = OGD$ .

eleva.

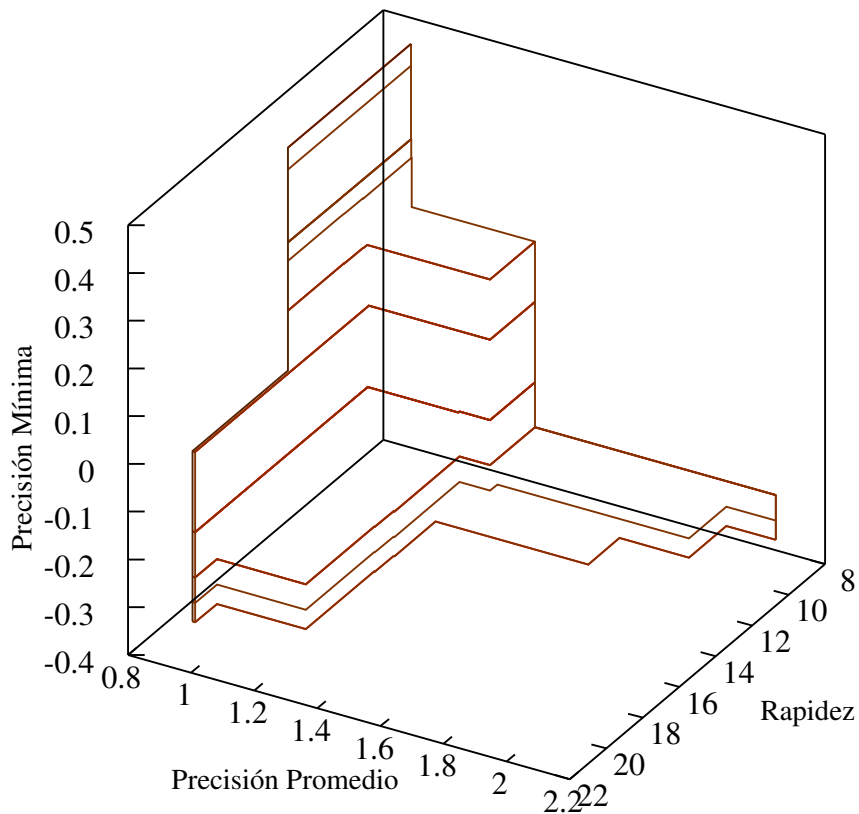


**Figura 4.4:** Frente de Pareto de la mediana y nivel de agrupación de las mediciones de precisión del algoritmo de flujo óptico propuesto

## 4.2 Evaluación del algoritmo Lucas-Kanade

En la figura 4.5 se muestra el frente de Pareto del algoritmo de flujo óptico Lucas-Kanade en el cual los resultados optimizados son la rapidez, la precisión promedio y la precisión mínima. Dentro de las características más importantes de este frente es que se logran mejores valores de rapidez, es decir, el método Lucas-Kanade es potencialmente más veloz que el algoritmo propuesto. Sin embargo, la precisión se ve afectada al obtener como mejor valor de error promedio 1,82 píxeles y un error máximo de 4,97 píxeles. Esto se debe principalmente a que en la secuencia utilizada existen movimientos abruptos entre algunos pares de imágenes consecutivas, por lo que este algoritmo no es capaz de estimar completamente el vector de movimiento principalmente en magnitud. En la figura 4.6 se puede observar el flujo óptico para cada par consecutivo de imágenes en la secuencia, donde se confirma que efectivamente existen problemas de precisión cuando el desplazamiento es mayor a 3 píxeles.

Además, en la figura 4.7 se muestra el frente de Pareto del método Lucas-Kanade para la mediana y el nivel de agrupación de las mediciones de precisión. En este otro experimento

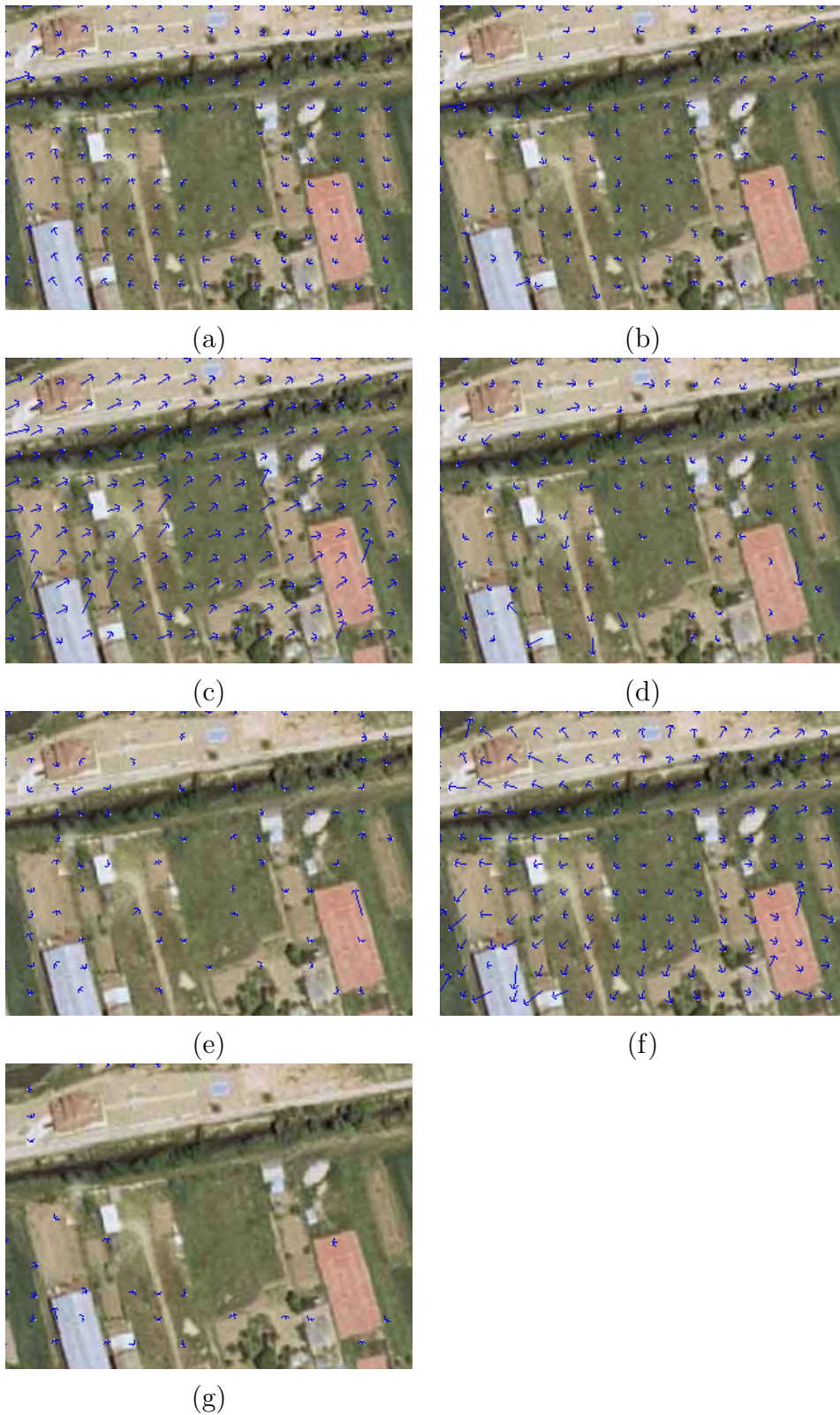


**Figura 4.5:** Frente de Pareto del algoritmo de flujo óptico Lucas-Kanade

se obtienen resultados de la mediana en rangos muy similares a la precisión promedio mostrada en la figura 4.5. Sin embargo, el nivel de agrupación de las mediciones de precisión tiene valores bajos, es decir, para la mayoría de las combinaciones paramétricas tienen gran dispersión en cuanto al error. En cuanto a la rapidez se tienen valores dentro de lo esperado, sin embargo, no se alcanzó un valor mayor a nueve a diferencia de la figura 4.5.

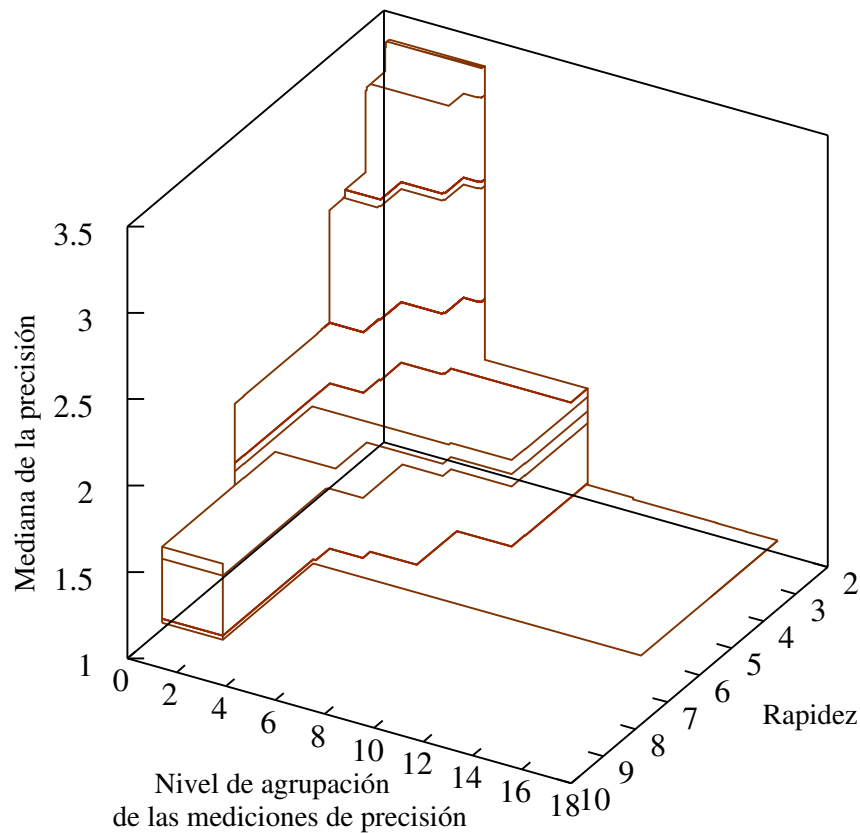
### 4.3 Evaluación del algoritmo Horn-Schunck

Para el método global de flujo óptico Horn-Schunck se obtuvo el frente de Pareto para la rapidez, precisión promedio y precisión mínima mostrado en la figura 4.8. Se puede observar que los valores de rapidez resultaron mejor que los del método Lucas-Kanade. Sin embargo, revisando los parámetros ofrecidos por el optimizador el número de iteraciones está por debajo de cinco y en su mayoría es solo una iteración. El problema de esto es que como se observa, tanto la precisión promedio como los valores mínimos de precisión tienen valores bajos que evidencian un alto nivel de error. Se debe recordar que este algoritmo parte del supuesto de que los desplazamientos son en principio diferenciales y estos experimentos están basados en la secuencia de imágenes de la figura A.1 que contempla saltos bastante largos de hasta siete píxeles. Por lo tanto, para obtener mejores resultados es necesario considerar secuencias con saltos pequeños de no más de 2 píxeles.



**Figura 4.6:** Flujo óptico de la secuencia de imágenes de la figura A.1 con el algoritmo Lucas-Kanade: (a) de la primera a la segunda, (b) de la segunda a la tercera, (c) de la tercera a la cuarta, (d) de la cuarta a la quinta, (e) de la quinta a la sexta, (f) de la sexta a la séptima, (g) de la séptima a la octava. Parámetros utilizados:  $variance = 9.98924732$ ,  $gradientKernelSize = 5$ ,  $kernelSize = 9$ ,  $gradientKernelType = OGD$ .





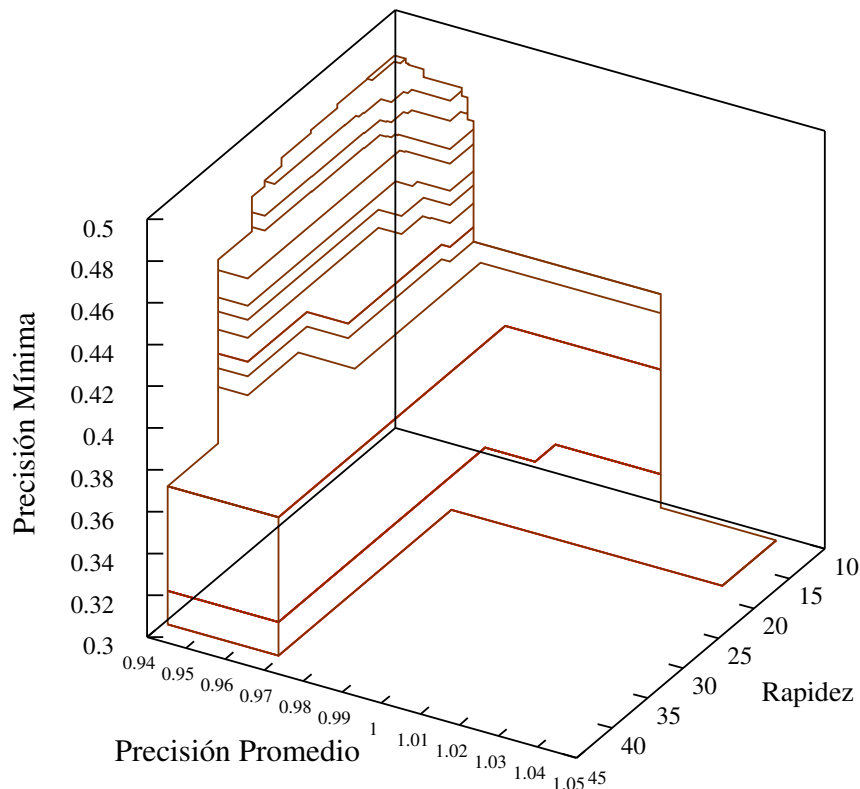
**Figura 4.7:** Frente de Pareto de la mediana y nivel de agrupación de las mediciones de precisión del algoritmo de flujo óptico Lucas-Kanade

En la figura 4.9 se puede observar el flujo óptico para cada par consecutivo de imágenes en la secuencia, donde se confirma que efectivamente existen problemas de precisión para desplazamientos largos.

En la figura 4.10 se muestra un comportamiento similar al obtenido en el frente de Pareto de la figura 4.8. Aunque estos resultados aparentemente tienen los mejores valores de rapidez, sus errores son inaceptables; sin embargo de los 295 individuos en el frente, una gran mayoría tienen valores de rapidez menores a uno, es decir, con tiempos mayores a 83.3ms con errores alrededor de 4 píxeles en promedio, que por resolución no se dibujan del frente de Pareto. Los individuos con valores de rapidez mayores a 15 presentan en común parámetros como: número de iteraciones no mayores a tres, errores superiores al 50% y utilizan el método de cálculo Diferencias Finitas para el gradiente.

## 4.4 Evaluación general

En la tabla 4.2 se muestran los resultados estadísticos con respecto al algoritmo de búsqueda exhaustiva de correspondencias para la secuencia de la figura A.1 que recorre la secuencia en 2.38s. Se observa que el método propuesto tiene las estadísticas del error más bajas y además resultó ser la de menor duración. El método clásico Lucas-



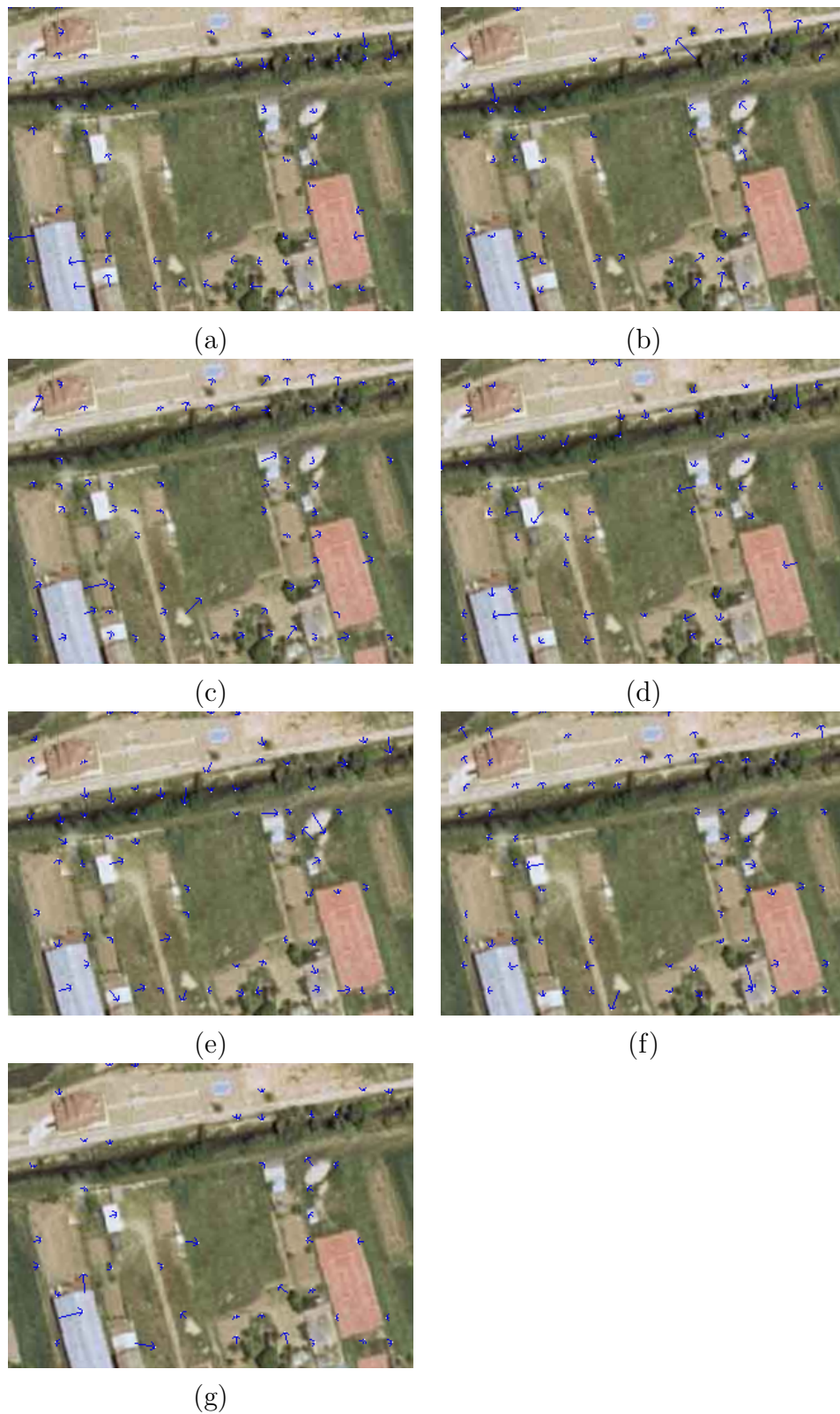
**Figura 4.8:** Frente de Pareto del algoritmo de flujo óptico Horn-Schunck

Kanade y el Horn-Schunck por el contrario ofrecen menor precisión y requieren mayor tiempo de ejecución.

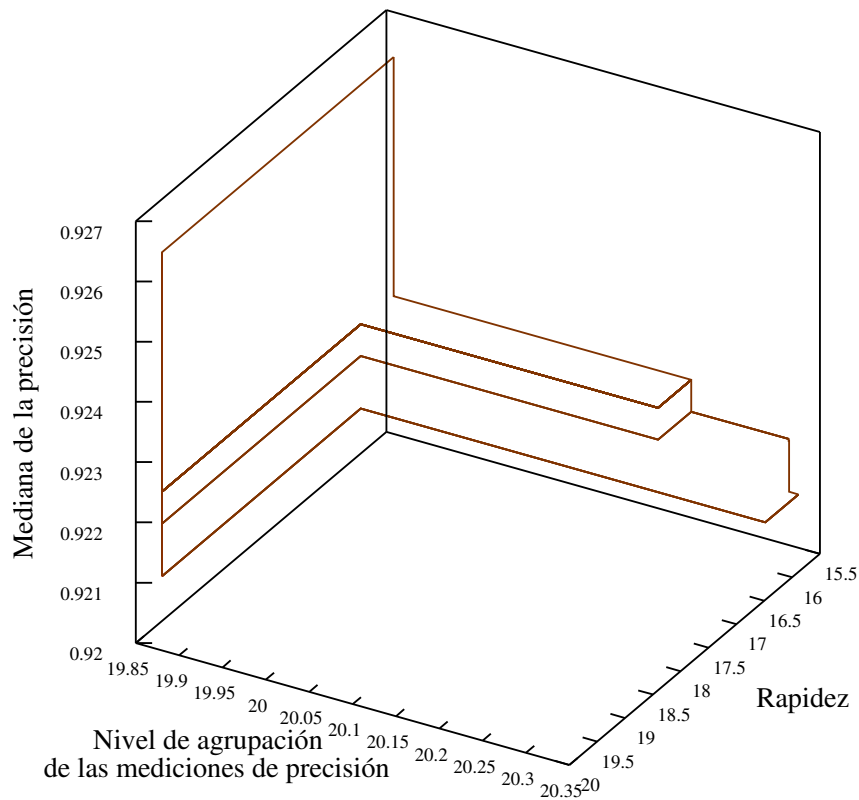
En la tabla 4.3 se muestran los resultados estadísticos del error de cada uno de los algoritmos de flujo óptico para la secuencia de la figura A.3 que se caracteriza por desplazamientos cortos no mayores a dos píxeles. Se observa un comportamiento similar al de la tabla 4.2 donde todavía el método propuesto presenta la mejor precisión y rapidez.

Por otra parte, en la tabla 4.4 se muestran los resultados estadísticos del algoritmo de flujo óptico propuesto para dos conjuntos de parámetros, el de la izquierda es el mismo que se utilizó en la tabla 4.2 y el conjunto de la derecha se obtuvo de otra optimización realizada para la secuencia de la figura A.2 que posee únicamente rotaciones de  $2^\circ$ ,  $5^\circ$  y  $3^\circ$ ,  $5^\circ$  y escalados de 4% y 5%, es decir los desplazamientos son largos pero se encuentran en un rango de valores cercanos; a diferencia de la secuencia de la figura A.1 que utiliza rangos más amplios de traslaciones, rotaciones y escalados.

Por lo tanto, los parámetros obtenidos de la optimización para la secuencia de la figura A.1 deben cubrir un rango más amplio sobre el cual ofrecer las mejores aptitudes posibles. Entonces, se observa que los tiempos de ejecución no varían para cada conjunto de parámetros escogidos, sin embargo se logra reducir el error para la secuencia de la figura A.2 utilizando la parametrización obtenida de la optimización realizada para esta secuencia. Lo que indica que el algoritmo piramidal acelerado con imágenes integrales es optimizable para operar con mejor precisión sobre secuencias con un rango más limitado



**Figura 4.9:** Flujo óptico de la secuencia de imágenes de la figura A.1 con el algoritmo Horn-Schunck: (a) de la primera a la segunda, (b) de la segunda a la tercera, (c) de la tercera a la cuarta, (d) de la cuarta a la quinta, (e) de la quinta a la sexta, (f) de la sexta a la séptima, (g) de la séptima a la octava. Parámetros utilizados:  $\lambda = 0.361504644$ ,  $\text{maxError} = 0.0170940179$ ,  $\text{maxIterations} = 3$ ,  $\text{variance} = 9.60215092$ ,  $\text{kernelSize} = 3$ ,  $\text{gradientKernelType} = \text{OGD}$ .



**Figura 4.10:** Frente de Pareto de la mediana y nivel de agrupación de las mediciones de precisión del algoritmo de flujo óptico Horn-Schunck

de transformaciones geométricas.

De la tabla 4.5 se puede observar las mediciones de error de los algoritmos clásicos de Lucas- Kanade y Horn-Schunck con respecto al algoritmo propuesto para la secuencia de la figura A.1. Se observa que el algoritmo de Horn-Schunck es el que más coincide con el método de flujo óptico propuesto con una diferencia promedio de 1,20 píxeles. Esto indica que a pesar de las ventajas del método de Lucas-Kanade con un enfoque en mínimos cuadrados, las técnicas iterativas ofrecen mejores características de precisión en el cálculo del flujo óptico.

En la figura 4.11 se muestra un frente de Pareto con la proyección en dos dimensiones de las aptitudes de rapidez y precisión promedio de los tres algoritmos de flujo óptico presentados. Se observa como el algoritmo propuesto es el que mejores resultados de precisión ofrece con características de rapidez aceptables. El método Lucas-Kanade se posiciona en termino medio en cuanto a precisión pero con valores de rapidez mayores a los del método propuesto. Por otro lado, el método Horn-Schunck tiene la precisión más baja aunque, como se analizó antes, esto es porque el algoritmo no itera más de tres veces; por lo que aumentando el número de iteraciones a más de veinte, por ejemplo, la precisión mejora y la rapidez disminuye.

<i>Métricas</i> \ <i>Método</i>	<i>Método propuesto</i>	<i>Lucas – Kanade</i>	<i>Horn – Schunk</i>
<i>Error Promedio (píxeles)</i>	3,59	4,28	4,59
<i>Error Máximo (píxeles)</i>	6,11	39,28	6,67
<i>Error Mínimo (píxeles)</i>	1,33	0,12	2,44
<i>Mediana del Error (píxeles)</i>	3,60	3,76	4,63
<i>Desviación estándar (píxeles)</i>	1,38	3,68	1,37
<i>Tiempo (ms)</i>	70,00	99,96	105,0
<i>Parámetros</i>	w=57 $\tau = 0,3812$ maxIterations=47 numLevels=2 maxError=0,0023 gradientKernel=OGD	variance= 9,9892 gradientKernelSize=5 kernelSize=9 gradientKernel=OGD	lambda= 0,3615 maxError=0,0171 maxIterations=3 variance=9,6022 kernelSize=3 gradientKernel=OGD

**Tabla 4.2:** Evaluación de los algoritmos de rastreo de puntos para la secuencia de la figura A.1

<i>Métricas</i> \ <i>Método</i>	<i>Método propuesto</i>	<i>Lucas – Kanade</i>	<i>Horn – Schunk</i>
<i>Error Promedio (píxeles)</i>	0,61	1,07	1,42
<i>Error Máximo (píxeles)</i>	2,27	13,79	2,06
<i>Error Mínimo (píxeles)</i>	0,01	0,08	0,67
<i>Mediana del Error (píxeles)</i>	0,52	0,91	1,42
<i>Desviación estándar (píxeles)</i>	0,36	0,92	0,33
<i>Tiempo (ms)</i>	37,5	52,5	57,0
<i>Parámetros</i>	w=57 $\tau = 0,3812$ maxIterations=47 numLevels=2 maxError=0,0023 gradientKernel=OGD	variance= 9,9892 gradientKernelSize=5 kernelSize=9 gradientKernel=OGD	lambda= 0,3615 maxError=0,0171 maxIterations=3 variance=9,6022 kernelSize=3 gradientKernel=OGD

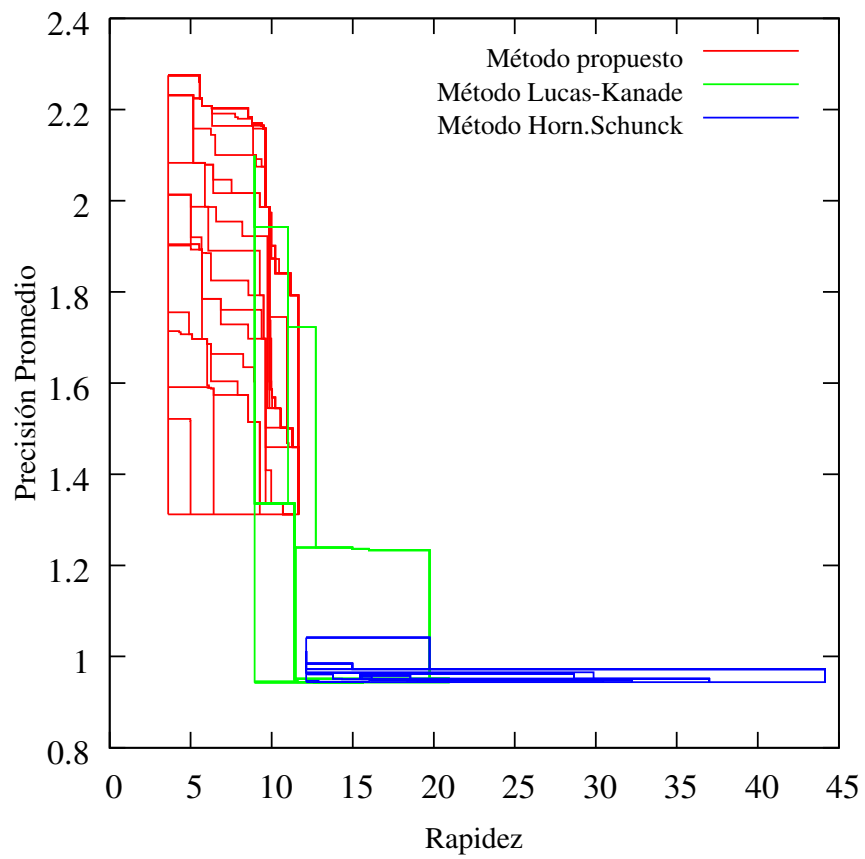
**Tabla 4.3:** Evaluación de los algoritmos de rastreo de puntos para la secuencia de la figura A.3

<i>Métricas</i> \ <i>Secuencias</i>	<i>Secuencia A.1</i>	<i>Secuencia A.2</i>	<i>Secuencia A.1</i>	<i>Secuencia A.2</i>
<i>Error Promedio (píxeles)</i>	3,59	4,09	2,94	3,065
<i>Error Máximo (píxeles)</i>	6,11	8,81	12,64	16,31
<i>Error Mínimo (píxeles)</i>	1,33	0,29	0,21	0,11
<i>Mediana del Error (píxeles)</i>	3,60	3,95	2,804	2,726
<i>Desviación estándar (píxeles)</i>	1,38	1,81	1,54	2,035
<i>Tiempo (ms)</i>	70,00	40,00	70,00	40,00
<i>Parámetros</i>	w=57 $\tau = 0,3812$ maxIterations=47 numLevels=2 maxError=0,0023 gradientKernel=OGD		w=32 $\tau = 0,375921249$ maxIterations=16 numLevels=3 maxError=0,00994140748 gradientKernel=OGD	

**Tabla 4.4:** Evaluación del algoritmo de flujo óptico propuesto para diferentes parametrizaciones y secuencias de imágenes

<i>Métricas</i> \ <i>Método</i>	<i>Lucas – Kanade</i>	<i>Horn – Schunk</i>
<i>Error Promedio (píxeles)</i>	2,31	1,20
<i>Error Máximo (píxeles)</i>	16,96	4,55
<i>Error Mínimo (píxeles)</i>	0,10	0,45
<i>Mediana del Error (píxeles)</i>	2,02	1,17
<i>Desviación estándar (píxeles)</i>	1,80	0,46
<i>Parámetros</i>	variance= 9,98924732 gradientKernelSize=5 kernelSize=32 gradientKernel=OGD	lambda= 0,361504644 maxError=0,0170940179 maxIterations=3 variance=9,60215092 kernelSize=3 gradientKernel=OGD

**Tabla 4.5:** Comparación de resultados entre el algoritmo de flujo óptico propuesto como referencia y los clásicos para la secuencia de la figura A.1



**Figura 4.11:** Comparación de frentes de Pareto de los algoritmos de flujo óptico





# Capítulo 5

## Conclusiones

Se desarrolló un estimador de movimiento basado en flujo óptico que mediante una etapa previa de extracción de puntos de interés implementada con el método “Fast Hessian Detection”, permite reducir los tiempos de ejecución en un 95,5% con respecto al algoritmo de búsqueda exhaustiva de correspondencias utilizado, para el caso de 1048 puntos de interés escogidos en una secuencia de imágenes de  $320 \times 240$  píxeles.

Se implementaron dos algoritmos de flujo óptico basados en las propuestas clásicas de Lucas-Kanade y Horn-Schunck; además se propuso un tercer método de flujo óptico local multi-escala, donde la ecuación que minimiza el error se resuelve con el método iterativo de Newton-Rhapson y se acelera el cálculo mediante la utilización de imágenes integrales.

La implementación multi-escala del algoritmo propuesto mediante estructuras piramidales le permite lidiar con desplazamientos largos y por lo tanto ser más preciso que los métodos clásicos. Para la secuencia A.1 se estimó un error promedio de 3,59 píxeles, mientras que para el método Lucas-Kanade se obtuvo un error promedio de 4,28 píxeles y para el método de Horn-Schunck el error promedio fue de 4,59 píxeles en promedio. Además, se estimó que la coincidencia del método de Lucas-Kanade difiere en promedio 2,31 píxeles con respecto al método propuesto y que el método clásico Horn-Schunck difiere del mismo en 1,20 píxeles en promedio.

Se utilizó un algoritmo genético de optimización con el cual se obtuvieron frentes de Pareto que mostraron el comportamiento de los métodos de flujo óptico en cuanto a rapidez y precisión. El método propuesto resultó ser el que alcanzó niveles de precisión más altos, con errores de 1,68 píxeles en promedio como mínimo para la secuencia A.1. Además se obtuvo que para una optimización de este método a partir de una secuencia con un rango más limitado de transformaciones geométricas que la secuencia A.1 ofrece mejores precisiones que para dicha secuencia.

## 5.1 Recomendaciones

En futuros trabajos relacionados con este proyecto se recomienda investigar sobre algún algoritmo de flujo óptico menos sensible a desplazamientos abruptos, ya sea de tipo diferencial o que se enfoque en otras propiedades de la imagen distintas del brillo. Sin embargo, en caso de que se utilice el estimador de movimiento de este proyecto se recomienda realizar nuevamente la optimización específicamente para la aplicación de interés. Es decir, se debe tener claro el tipo de transformaciones geométricas y los rangos de magnitud de los desplazamientos predominantes de la aplicación. Esto para diseñar una secuencia a optimizar lo más representativa posible, recordando no excederse con los rangos pues entre más angostos sean, mejor precisión se obtiene.

Por otra parte, al optimizador se le indican los intervalos dentro de los cuales puede tomar valores aleatorios para cada parámetro del algoritmo de flujo óptico a optimizar, por lo que debe tenerse especial cuidado con los intervalos del número máximo de iteraciones y del error máximo de convergencia; esto porque al optimizar rapidez es probable que se escoja un número máximo de iteraciones bajo o un error máximo alto que causará un valor de rapidez elevado pero con una precisión deficiente.

Además, es recomendable agregar nuevas métricas de optimización en caso de que resulten indispensables para alguna aplicación en específico, pues esto producirá un ajuste de parámetros más acordes con la aplicación de interés.

# Bibliografía

- [1] Optical flow evaluation results. URL <http://vision.middlebury.edu/flow/eval/results/results-e1.php>.
- [2] Alfonso Alba, Edgar Arce-Santana, and Mariano Rivera. Optical flow estimation with prior models obtained from phase correlation. In *Proceedings of the 6th international conference on Advances in visual computing - Volume Part I*, ISVC'10, pages 417–426, Berlin, Heidelberg, 2010. Springer-Verlag. URL <http://dl.acm.org/citation.cfm?id=1939921.1939968>.
- [3] Shigeru Ando. Consistent Gradient Operators. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(3):252–265, March 2000. URL <http://dx.doi.org/10.1109/34.841757>.
- [4] S. Baker, D. Scharstein, J.P. Lewis, S. Roth, M.J. Black, and R. Szeliski. A Database and Evaluation Methodology for Optical Flow. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, oct. 2007.
- [5] Simon Baker and Iain Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *Int. J. Comput. Vision*, 56(3):221–255, February 2004. URL <http://dx.doi.org/10.1023/B:VISI.0000011205.11775.fd>.
- [6] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008. URL <http://dx.doi.org/10.1016/j.cviu.2007.09.014>.
- [7] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM Comput. Surv.*, 27(3):433–466, September 1995. URL <http://doi.acm.org/10.1145/212094.212141>.
- [8] Jean-Yves Bouguet. Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm, 2000. URL [http://robots.stanford.edu/cs223b04/algo\\_tracking.pdf](http://robots.stanford.edu/cs223b04/algo_tracking.pdf).
- [9] A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schnorr. Variational optical flow computation in real time. *Image Processing, IEEE Transactions on*, 14(5):608–615, may 2005.

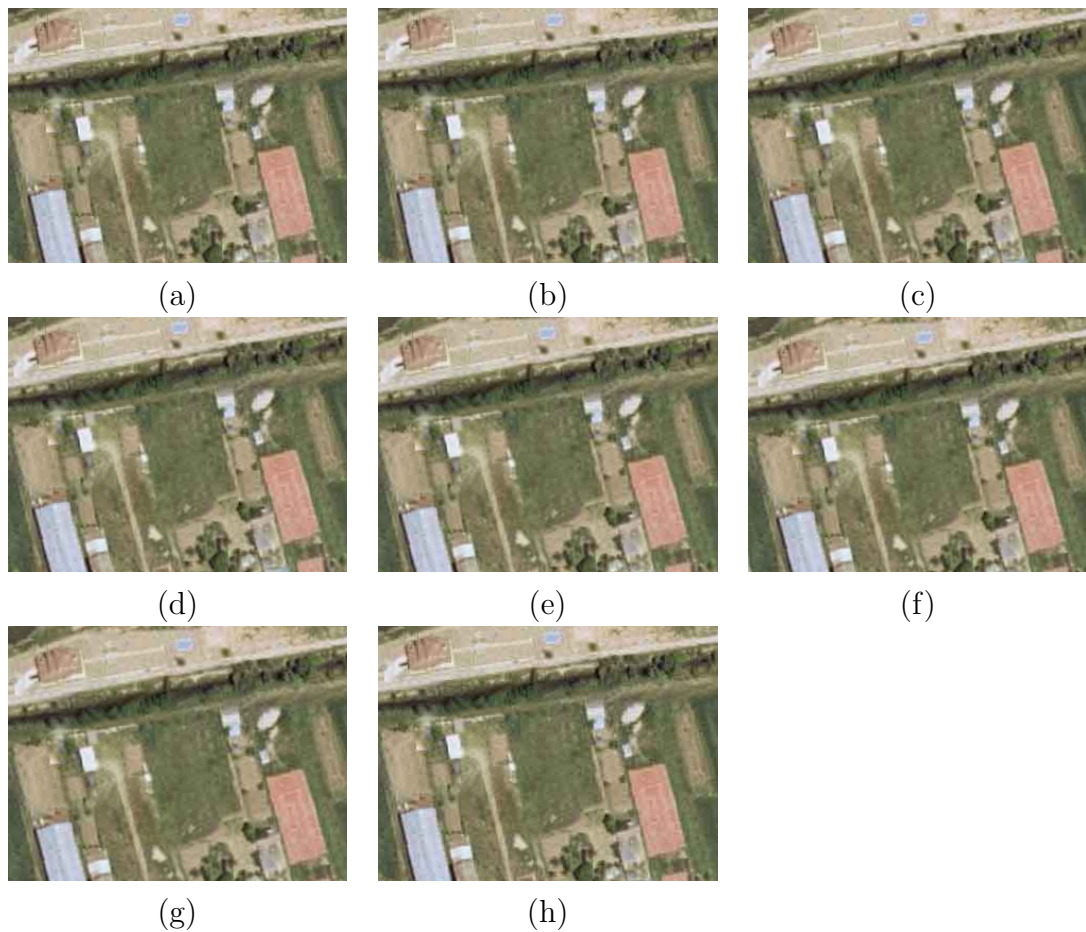
- [10] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods. *Int. J. Comput. Vision*, 61(3):211–231, February 2005. URL <http://dl.acm.org/citation.cfm?id=1028916.1035417>.
- [11] David Corne, Joshua D. Knowles, and Martin J. Oates. The Pareto Envelope-Based Selection Algorithm for Multi-objective Optimisation. In *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, PPSN VI, pages 839–848, London, UK, UK, 2000. Springer-Verlag. URL <http://dl.acm.org/citation.cfm?id=645825.669102>.
- [12] Franklin C. Crow. Summed-area tables for texture mapping. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '84, pages 207–212, New York, NY, USA, 1984. ACM. URL <http://doi.acm.org/10.1145/800031.808600>.
- [13] P. Doerfler and P. Alvarado. LTI-Lib -A C++ Open Source Computer Vision Library. In K.-F. Kraiss, editor, *Advanced Man-Machine Interaction. Fundamentals and Implementation.*, pages 399–421. Springer Verlag, 2006.
- [14] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. URL <http://doi.acm.org/10.1145/358669.358692>.
- [15] David J. Fleet and A. D. Jepson. Computation of component image velocity from local phase information. *Int. J. Comput. Vision*, 5(1):77–104, September 1990. URL <http://dx.doi.org/10.1007/BF00056772>.
- [16] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall, 3 edition, 2008.
- [17] Xiaoxin Guo, Zhiwen Xu, Yueping Feng, Yunxiao Wang, and Zhengxuan Wang. Optical flow computation with fourth order partial differential equations. In *Proceedings of the 2006 joint IAPR international conference on Structural, Syntactic, and Statistical Pattern Recognition*, SSPR'06/SPR'06, pages 279–286, Berlin, Heidelberg, 2006. Springer-Verlag. URL [http://dx.doi.org/10.1007/11815921\\_30](http://dx.doi.org/10.1007/11815921_30).
- [18] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.
- [19] Berthold K. P. Horn and Brian G. Schunck. Determining Optical Flow. *ARTIFICIAL INTELLIGENCE*, 17:185–203, 1981.
- [20] Bruce D. Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. pages 674–679, 1981.

- [21] Luis Merino, Fernando Caballero, J.Ramiro Martínez-de Dios, Iván Maza, and Aníbal Ollero. An Unmanned Aircraft System for Automatic Forest Fire Monitoring and Measurement. *Journal of Intelligent & Robotic Systems*, 65:533–548, 2012. URL <http://dx.doi.org/10.1007/s10846-011-9560-x>.
- [22] T. Senst, V. Eiselein, M. Patzold, and T. Sikora. Efficient real-time local optical flow estimation by means of integral projections. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 2345–2348, sept. 2011.
- [23] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis, and Machine Vision*, volume 3. Thomson-Engineering, 3 edition, 2007.
- [24] Soujanya Vendra, Giampiero Campa, Marcello Napolitano, Marco Mammarella, Mario Fravolini, and Mario Perhinschi. Addressing corner detection issues for machine vision based UAV aerial refueling. *Machine Vision and Applications*, 18:261–273, 2007. URL <http://dx.doi.org/10.1007/s00138-006-0056-9>. 10.1007/s00138-006-0056-9.
- [25] Yi-Sheng Yao and R. Chellappa. Tracking a dynamic set of feature points. *Trans. Img. Proc.*, 4(10):1382–1395, October 1995. URL <http://dx.doi.org/10.1109/83.465103>.

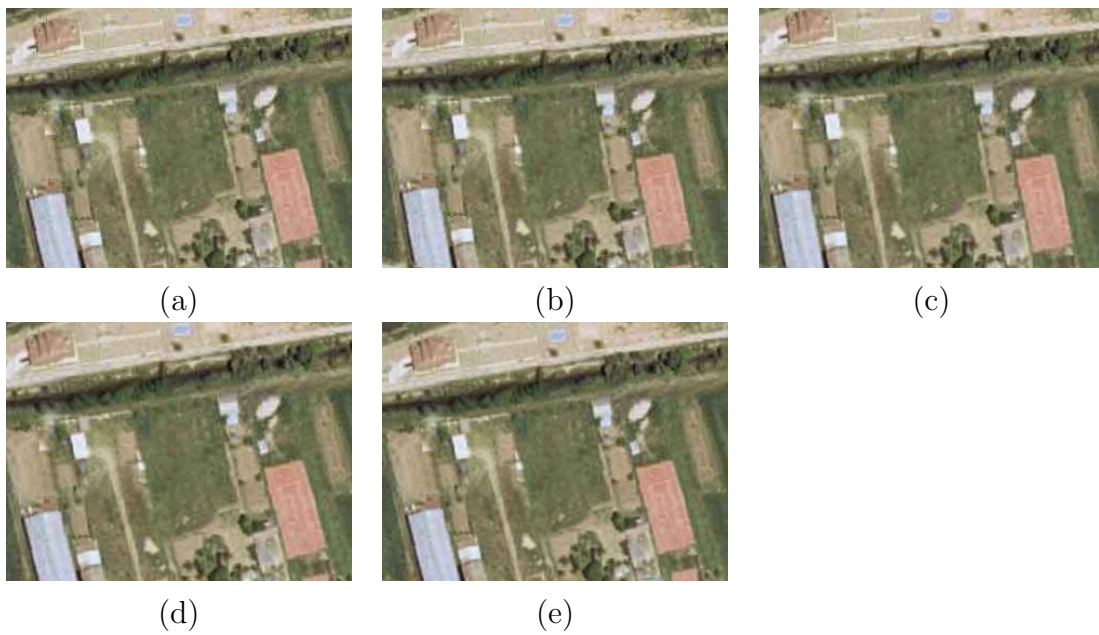


# Apéndice A

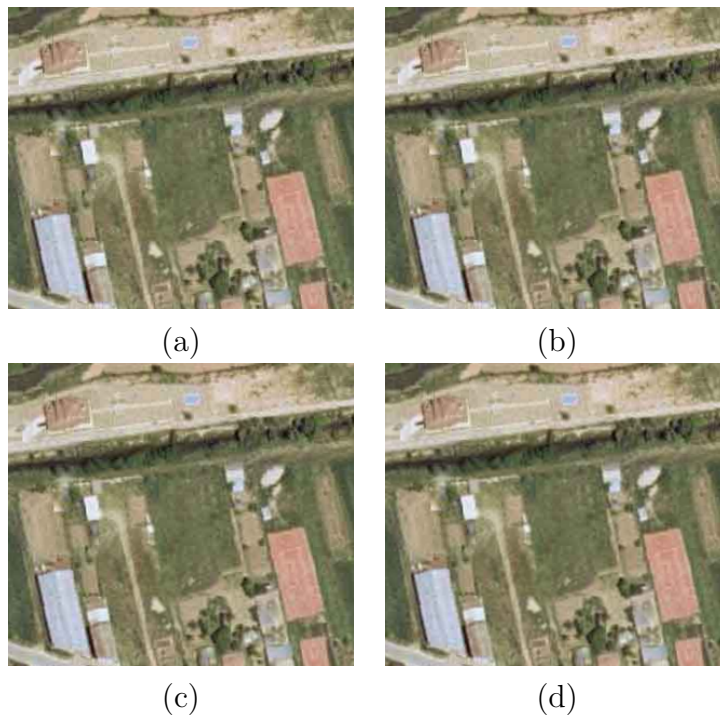
## Secuencias de imágenes



**Figura A.1:** Secuencia de imágenes: (a) es la imagen inicial, (b) es una rotación de  $1^\circ$  en el sentido de las manecillas del reloj, (c) es una rotación de  $3^\circ$  en el sentido contrario de las manecillas del reloj, (d) es una traslación en  $(x,y)$  de  $(2,-1)$  píxeles, (e) es otra traslación de  $(-3,5)$  píxeles, (f) es una traslación larga de  $(7,4)$  píxeles, (g) es un escalado de  $1,02$  (2%) y (h) es otro escalado de  $0,95$  (-5%). Todas las transformaciones son respecto a la imagen anterior y se utilizó un interpolador cúbico.



**Figura A.2:** Secuencia de imágenes: (a) es la imagen inicial, (b) es una rotación de  $2,5^\circ$  en el sentido de las manecillas del reloj, (c) es un escalado de 1,04 (4%), (d) es una rotación de  $3,5^\circ$  en el sentido contrario de las manecillas del reloj, (e) es otro escalado de 0,95 (-5%). Todas las transformaciones son respecto a la imagen anterior y se utilizó un interpolador cúbico.



**Figura A.3:** Secuencia de imágenes: (a) es la imagen inicial, (b) es una traslación de (2,-1) píxeles, (c) es un escalado de 1,01 (1%), (d) es una rotación de  $0,5^\circ$  en el sentido de las manecillas del reloj. Todas las transformaciones son respecto a la imagen anterior y se utilizó un interpolador cúbico.



# Índice alfabético

búsqueda exhaustiva de correspondencias,  
11

detector de Harris, 10

ecuaciones de Euler-Lagrange, 7  
estructuras piramidales, 17

filtro gaussiano, 7

flujo óptico, 5

frente de Pareto, 12

gradiente, 11

Horn-Schunck, 5

imágenes integrales, 10

Lucas-Kanade, 5

método Gauss-Seidel, 7

matriz Hessiana, 10

objetivos, 3

puntos característicos, 10

transformación geométrica, 11

transformaciones de similitud, 15