



UNIVERSIDAD SIMÓN BOLÍVAR

Ingeniería de la Computación

Estabilización en tiempo real de imágenes capturadas desde un vehículo aéreo teledirigido

Por

Marynel Vázquez

Proyecto de Grado

Presentado ante la Ilustre Universidad Simón Bolívar

como Requerimiento Parcial para Optar el Título de

Ingeniero en Computación

Sartenejas, Octubre de 2008



UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE INGENIERÍA DE LA COMPUTACIÓN

ACTA FINAL DEL PROYECTO DE GRADO

ESTABILIZACIÓN EN TIEMPO REAL DE IMÁGENES CAPTURADAS DESDE UN
VEHÍCULO AÉREO TELEDIRIGIDO

Presentado Por:
MARYNEL VÁZQUEZ

Este proyecto de Grado ha sido aprobado por el siguiente jurado examinador:

Blai Bonet

Prof. Blai Bonet

Alexandra La Cruz

Prof. Alexandra La Cruz

Carolina Chang

Prof. Carolina Chang (Tutor Académico)

SARTENEJAS, 25 de septiembre de 2008

Estabilización en tiempo real de imágenes capturadas desde un vehículo aéreo teledirigido

Por

Marynel Vázquez

RESUMEN

La utilización de vehículos aéreos en el campo de la robótica es sumamente popular. En diversas actividades se puede aprovechar el mejoramiento de la calidad de las imágenes que desde ellos se registran. Con este fin, en el presente proyecto se plantea la estabilización en tiempo real de imágenes capturadas desde una plataforma aérea. Para ello, se considera la información que puede extraerse de los videos grabados desde una cámara adaptada rígidamente a un helicóptero a control remoto.

Se implementó una aplicación para estabilizar, configurable según las necesidades del usuario. A pesar de implicar un pequeño retraso en la reproducción del video final, el procesamiento *online* fácilmente llega a alcanzar una velocidad promedio entre 20 y 28 imágenes por segundo. Se llevaron a cabo varios experimentos para orientar sobre los valores de las variables que influencian el resultado obtenido.

La estabilización puede resumirse en suprimir la componente no intencional del movimiento global percibido entre imágenes, considerando rotación y desplazamientos a lo largo de los ejes cartesianos. Primero se estima este movimiento global por medio de un procedimiento basado en el cálculo de vectores de flujo óptico. Posteriormente, se filtra un modelo “rígido”, producto de la descomposición de la transformación que modela el movimiento, para obtener la cantidad de compensación que debe aplicársele a las imágenes para estabilizarlas.

Al modelar el movimiento global se consideraron una transformación afín, una similar y otra bilineal. Con ellas se lleva a cabo una regresión por mínimos cuadrados para estimar los valores que las componen. Para el caso particular de la primera, también se puso a prueba la utilización de una regresión por mínimos cuadrados totales. Todas las combinaciones de transformación y método de estimación antes mencionadas resultaron útiles para llevar a cabo la estabilización. Sin embargo, con el uso de una similar, bajo la presencia reducida de ruido en las imágenes, se evidenció una menor efectividad al buscar compensar totalmente el movimiento percibido de manera general.

Agradecimientos

En primer lugar doy gracias a Dios.

A mi familia y amigos les agradezco la confianza que siempre tuvieron en mí. En particular, gracias a mis padres, Marisol Ugarte y Nelson Vázquez, a mi abuela Graciela Muñoz y a mi prima Arianne Chang por su ayuda incondicional. A Celso Gorrín, Ana Balliache, Diego Guerrero, Pedro Piñango, Fabiola Di Bartolo, Tomás Lampo y a todos aquellos que me dieron ánimos por alegrarme los días más complicados. A Francelice Sardá, además, por su paciencia infinita y acertados consejos.

Gracias a los integrantes del Grupo de Inteligencia Artificial de la Universidad y a los de Aulas Computarizadas por su compañía y colaboración constante.

Este trabajo no pudo haberse realizado sin el apoyo de la Fundación Carolina Chang para el Desarrollo de la Ciencia. Gracias, especialmente, a mi tutora por haber hecho divertido el desarrollo del proyecto ante cualquier circunstancia y haberme dado fortaleza y motivación para seguir adelante en los momentos más difíciles.

Índice general

Agredecimientos	III
Índice general	IV
Índice de cuadros	VII
Índice de figuras	X
Capítulo 1. Introducción	1
Capítulo 2. Marco Teórico	4
2.1. Descripción general del proceso de estabilización de video	5
2.2. Detección y estimación del movimiento de “buenos” rasgos	8
2.2.1. Interpretación geométrica	10
2.2.2. Refinamiento iterativo	12
2.2.3. Campo de movimiento afín	13
2.2.4. Selección de rasgos	13
2.2.5. Método Piramidal	15
2.3. Estimación del movimiento global	16
2.3.1. Transformaciones	16
2.3.2. Estimación de la transformación	18
2.4. Estabilización del movimiento percibido entre imágenes	21
Capítulo 3. Implementación	24
3.1. Captura de imágenes	26
3.2. Estimación del flujo óptico y movimiento global	27
3.2.1. Selección de “buenos” rasgos	27
3.2.2. Estimación del campo de movimiento	29
3.2.3. Estimación de la transformación que modela el movimiento global	32
3.3. Estimación del movimiento no intencional	34
3.3.1. Simplificación de la transformación que modela el movimiento global	35

3.3.2. Estimación de la componente intencional del movimiento global	37
3.3.3. Detección de movimientos impulsivos	38
3.4. Reducción de vibraciones	39
3.5. Visualización de la secuencia estabilizada	39
3.6. Consideraciones especiales ante la presencia de ruido	39
Capítulo 4. Experimentos y Resultados	41
4.1. Evaluación del proceso de estimación del flujo óptico	41
4.1.1. Experimento I: Influencia del tamaño de la vecindad	44
4.1.2. Experimento II: Influencia del máx. nivel de profundidad de la pirámide . . .	52
4.1.3. Experimento III: Influencia de la cota mínima β como criterio de parada . .	54
4.2. Evaluación del proceso de compensación	55
4.2.1. Experimento IV: Evaluación del modelo y del método de estimación	55
4.2.2. Experimento V: Evaluación de la velocidad de procesamiento	60
4.2.3. Experimento VI: Desempeño final de la estabilización	62
Capítulo 5. Conclusiones y Recomendaciones	66
Bibliografía	72
Apéndice A. Definiciones de interés	76
Apéndice B. ChocoLate y la metodología de trabajo	80
Apéndice C. Detalles sobre la aplicación implementada	82
Apéndice D. Estimación del gradiente	84
Apéndice E. Ejemplos sobre la selección de rasgos	86
Apéndice F. Construcción de pirámides	90
Apéndice G. Cálculos a nivel de subpixeles	92
Apéndice H. Regresión por mínimos cuadrados totales	93

Apéndice I. Descomposición de una transformación lineal	97
Apéndice J. Aplicación de una transformación geométrica	99
Apéndice K. Listado de videos extraídos de <i>YouTube.com</i>	100
Apéndice L. Detalles sobre los Experimentos I, II y III	102
L.1. Detalles sobre resultados del Experimento I	102
L.2. Detalles sobre resultados del Experimento II	105
L.3. Detalles sobre resultados del Experimento III	106
Apéndice M. Detalles sobre los Experimentos IV,V,VI	107
M.1. Detalles sobre resultados del Experimento IV	107
M.1.1. Ganancia con respecto a la diferencia entre imágenes	107
M.1.2. Número de iteraciones	123
M.1.3. Porcentaje de región válida	127
M.2. Detalles sobre resultados del Experimento V	131
M.3. Detalles sobre resultados del Experimento VI	134
M.3.1. Detalles sobre secuencia de <i>YouTube.com</i>	134
M.3.2. Detalles sobre secuencia capturada desde ChocoLate	144

Índice de cuadros

1.	Coeficientes de correlación entre el número de rasgos, el valor del máximo y el del mínimo de los mínimos autovalores	48
2.	Mejores ganancias obtenidas en promedio para un máximo de 100 rasgos.	58
3.	Promedios del mínimo valor de los mínimos autovalores y nivel de calidad encontrado según $T(W_s)$	102
4.	Error promedio de estimación de movimiento según $T(W_s)$ y $T(W_e)$	103
5.	Error estándar del promedio de error en la estimación de movimiento según $T(W_s)$ y $T(W_e)$	103
6.	Porcentaje de rasgos “perdidos” según $T(W_s)$ y $T(W_e)$	104
7.	Error promedio de estimación de movimiento obtenido en el Experimento II	105
8.	Error estándar del promedio de error obtenido en la estimación de movimiento del Experimento II	105
9.	Proporción de rasgos “perdidos” considerando diferentes profundidades para la pirámide utilizada	105
10.	Error promedio de estimación de movimiento obtenido en el Experimento III	106
11.	Error estándar del promedio de error obtenido en la estimación de movimiento del Experimento III	106
12.	Proporción de rasgos “perdidos” obtenido en el Experimento III	106
13.	Ganancia promedio compensando con un modelo afín por m.c. y máximo 500 rasgos. 107	107
14.	Ganancia promedio compensando con un modelo afín por m.c. y máximo 300 rasgos. 108	108
15.	Ganancia promedio compensando con un modelo afín por m.c. y máximo 100 rasgos. 108	108
16.	Ganancia promedio compensando con un modelo afín por m.c.t. y máximo 500 rasgos.109	109
17.	Ganancia promedio compensando con un modelo afín por m.c.t. y máximo 300 rasgos.109	109
18.	Ganancia promedio compensando con un modelo afín por m.c.t. y máximo 100 rasgos.110	110
19.	Ganancia promedio compensando con un modelo similar por m.c. y máximo 500 rasgos.110	110
20.	Ganancia promedio compensando con un modelo similar por m.c. y máximo 300 rasgos.111	111
21.	Ganancia promedio compensando con un modelo similar por m.c. y máximo 100 rasgos.111	111

22. Ganancia promedio compensando con un modelo bilineal por m.c. y máximo 500 rasgos.	112
23. Ganancia promedio compensando con un modelo bilineal por m.c. y máximo 300 rasgos.	112
24. Ganancia promedio compensando con un modelo bilineal por m.c. y máximo 100 rasgos.	113
25. Error estándar de la ganancia promedio compensando con un modelo afín por m.c. y máximo 500 rasgos.	114
26. Error estándar de la ganancia promedio compensando con un modelo afín por m.c. y máximo 300 rasgos.	115
27. Error estándar de la ganancia promedio compensando con un modelo afín por m.c. y máximo 100 rasgos.	115
28. Error estándar de la ganancia promedio compensando con un modelo afín por m.c.t. y máximo 500 rasgos.	116
29. Error estándar de la ganancia promedio compensando con un modelo afín por m.c.t. y máximo 300 rasgos.	116
30. Error estándar de la ganancia promedio compensando con un modelo afín por m.c.t. y máximo 100 rasgos.	117
31. Error estándar de la ganancia promedio compensando con un modelo similar por m.c. y máximo 500 rasgos.	117
32. Error estándar de la ganancia promedio compensando con un modelo similar por m.c. y máximo 300 rasgos.	118
33. Error estándar de la ganancia promedio compensando con un modelo similar por m.c. y máximo 100 rasgos.	118
34. Error estándar de la ganancia promedio compensando con un modelo bilineal por m.c. y máximo 500 rasgos.	119
35. Error estándar de la ganancia promedio compensando con un modelo bilineal por m.c. y máximo 300 rasgos.	119
36. Error estándar de la ganancia promedio compensando con un modelo bilineal por m.c. y máximo 100 rasgos.	120

37. Mejores ganancias obtenidas en promedio para un máximo de 500 rasgos.	121
38. Mejores ganancias obtenidas en promedio para un máximo de 300 rasgos.	122
39. Promedio de iteraciones para estimar el modelo de transformación global con un máximo de 500 rasgos	123
40. Error estándar del promedio de iteraciones llevadas a cabo para estimar el modelo de transformación global con un máximo de 500 rasgos	124
41. Promedio de iteraciones para estimar el modelo de transformación global con un máximo de 300 rasgos	124
42. Error estándar del promedio de iteraciones llevadas a cabo para estimar el modelo de transformación global con un máximo de 300 rasgos	125
43. Promedio de iteraciones para estimar el modelo de transformación global con un máximo de 100 rasgos	125
44. Error estándar del promedio de iteraciones llevadas a cabo para estimar el modelo de transformación global con un máximo de 100 rasgos	126
45. Porcentaje de región válida promedio con un máximo de 500 rasgos	127
46. Error estándar del promedio del porcentaje de región válida con un máximo de 500 rasgos	128
47. Porcentaje de región válida promedio con un máximo de 300 rasgos	128
48. Error estándar del promedio del porcentaje de región válida con un máximo de 300 rasgos	129
49. Porcentaje de región válida promedio con un máximo de 100 rasgos	129
50. Error estándar del promedio del porcentaje de región válida con un máximo de 100 rasgos	130
51. Valor de ε en el Experimento V	131
52. Promedio del aproximado número de imágenes procesadas por seg. en la secuencia 1	131
53. Promedio del aproximado número de imágenes procesadas por seg. en la secuencia 2	132
54. Promedio del aproximado número de imágenes procesadas por seg. en la secuencia 3	132
55. Promedio del aproximado número de imágenes procesadas por seg. en la secuencia 4	132
56. Promedio del aproximado número de imágenes procesadas por segundo en total . .	133
57. Valor de ε en el Experimento VI	134

Índice de figuras

1.	Relaciones geométricas entre los de parches de intensidad de dos imágenes	10
2.	Esquema de trabajo para el procesamiento en tiempo real	24
3.	Esquema general del funcionamiento de la aplicación implementada	25
4.	Proceso de simplificación de las transformaciones estimadas	35
5.	Ejemplo de la ventana corrediza	38
6.	Ruido en imágenes transmitidas analógicamente	40
7.	Número de “buenos” rasgos seleccionados para estimar el flujo óptico según $T(W_s)$. .	46
8.	Promedios del mínimo valor de los mínimos autovalores y nivel de calidad según $T(W_s)$	47
9.	Error promedio escalado de la estimación de movimiento al variar $T(W_s)$ y $T(W_e)$. .	49
10.	Proporción de rasgos perdidos según $T(W_s)$ y $T(W_e)$	50
11.	Resultados del Experimento II	53
12.	Resultados del Experimento III.	54
13.	Promedio del aproximado número de imágenes procesadas por segundo en total . .	61
14.	ChocoLate	80
15.	Imágenes capturadas desde ChocoLate	80
16.	Ventanas que conforman constantemente la interfaz de la aplicación	82
17.	Proceso de selección de rasgos en par de imágenes número 650 del segmento de video <i>FlyingWinter3</i> , considerando tamaños de ventana de 3×3 y 7×7 píxeles y utilizando una pirámide de 4 niveles de profundidad máximo.	87
18.	Proceso de selección de rasgos en par de imágenes número 223 del segmento de video <i>FlyingWinter3</i> , considerando tamaños de ventana de 3×3 , 7×7 y 13×13 píxeles y utilizando una pirámide de 4 niveles de profundidad máximo.	88
19.	Pirámide de 4 niveles para una imagen inicial de 32×24 píxeles	90
20.	Regresión lineal por mínimos cuadrados y mínimos cuadrados ortogonales para un mismo conjunto de datos en 2D	93
21.	Máximo de los mínimos autovalores encontrado vs. número de rasgos seleccionados para $T(W_s) = 3$	103
22.	Ejemplos de rasgos “perdidos”	104

23.	Secuencia de imágenes sin filtrar las transformaciones en video de <i>YouTube.com</i> . . .	135
24.	Secuencia de imágenes filtrando las transformaciones en video de <i>YouTube.com</i> . . .	135
25.	Estimación del movimiento acumulado y su componente intencional bajo un modelo afín por m.c. en la secuencia de <i>YouTube.com</i>	136
26.	Magnitud de la compensación según el estimado de movimiento global bajo un modelo afín por m.c. en la secuencia de <i>YouTube.com</i>	137
27.	Estimación del movimiento acumulado y su componente intencional bajo un modelo afín por m.c.t. en la secuencia de <i>YouTube.com</i>	138
28.	Magnitud de la compensación según el estimado de movimiento global bajo un modelo afín por m.c.t. en la secuencia de <i>YouTube.com</i>	139
29.	Estimación del movimiento acumulado y su componente intencional bajo un modelo similar por m.c. en la secuencia de <i>YouTube.com</i>	140
30.	Magnitud de la compensación según el estimado de movimiento global bajo un modelo similar por m.c. en la secuencia de <i>YouTube.com</i>	141
31.	Estimación del movimiento acumulado y su componente intencional bajo un modelo bilineal por m.c. en la secuencia de <i>YouTube.com</i>	142
32.	Magnitud de la compensación según el estimado de movimiento global bajo un modelo bilineal por m.c. en la secuencia de <i>YouTube.com</i>	143
33.	Secuencia de imágenes sin filtrar las transformaciones en video capturado desde ChocoLate estimando un modelo afín por M.C.	145
34.	Secuencia de imágenes filtrando las transformaciones en video capturado desde ChocoLate estimando un modelo afín por M.C.	145
35.	Secuencia de imágenes sin filtrar las transformaciones en video capturado desde ChocoLate estimando un modelo afín por M.C.T.	146
36.	Secuencia de imágenes filtrando las transformaciones en video capturado desde ChocoLate estimando un modelo afín por M.C.T.	146
37.	Secuencia de imágenes sin filtrar las transformaciones en video capturado desde ChocoLate estimando un modelo similar por M.C.	147
38.	Secuencia de imágenes filtrando las transformaciones en video capturado desde ChocoLate estimando un modelo similar por M.C.	147

39.	Secuencia de imágenes sin filtrar las transformaciones en video capturado desde ChocoLate estimando un modelo bilineal por M.C.	148
40.	Secuencia de imágenes filtrando las transformaciones en video capturado desde ChocoLate estimando un modelo bilineal por M.C.	148
41.	Estimación del campo de movimiento en secuencia de imágenes de video capturado desde ChocoLate estimando un modelo bilineal por M.C.	149
42.	Diferencia entre imágenes consecutivas luego de compensar totalmente el movimiento estimado en secuencia de video capturado desde ChocoLate, estimando un modelo bilineal por M.C.	149
43.	Estimación del movimiento acumulado y su componente intencional bajo un modelo afín por m.c. en la secuencia capturada desde ChocoLate	150
44.	Magnitud de la compensación según el estimado de movimiento global bajo un modelo afín por m.c. en la secuencia capturada desde ChocoLate	151
45.	Estimación del movimiento acumulado y su componente intencional bajo un modelo afín por m.c.t. en la secuencia capturada desde ChocoLate	152
46.	Magnitud de la compensación según el estimado de movimiento global bajo un modelo afín por m.c.t. en la secuencia capturada desde ChocoLate	153
47.	Estimación del movimiento acumulado y su componente intencional bajo un modelo similar por m.c. en la secuencia capturada desde ChocoLate	154
48.	Magnitud de la compensación según el estimado de movimiento global bajo un modelo similar por m.c. en la secuencia capturada desde ChocoLate	155
49.	Estimación del movimiento acumulado y su componente intencional bajo un modelo bilineal por m.c. en la secuencia capturada desde ChocoLate	156
50.	Magnitud de la compensación según el estimado de movimiento global bajo un modelo bilineal por m.c. en la secuencia capturada desde ChocoLate	157

Capítulo 1

Introducción

La utilización de vehículos aéreos en el campo de la robótica es sumamente popular. Con ellos aumentan las posibilidades de llevar a cabo tareas de búsqueda y rescate, vigilancia, inspección de terrenos e infraestructuras, entre otras.

En los sistemas de percepción de estas máquinas suelen encontrarse dispositivos de captura de video. Tanto la disminución de sus costos como la gran cantidad de información que puede ser extraída de las imágenes que registran, favorecen el desarrollo de la tecnología en este ámbito.

Las secuencias de imágenes capturadas desde una cámara en movimiento suelen presentar vibraciones que degradan su calidad y resultan molestas visualmente. La estabilización de video consiste, justamente, en suprimir el movimiento impulsivo, o de alta frecuencia, que se percibe en el producto del movimiento propio de la cámara con la que se registra.

A los vehículos aéreos, y más a los autónomos, suelen incorporárseles diferentes sensores para tener mayor información sobre su movimiento y ubicación en el espacio, tales como unidades de medidas iniciales o sistemas de posicionamiento global. Los datos que se registran con éstos pueden ser utilizados para estabilizar satisfactoriamente videos grabados desde estas naves, sin embargo a un costo adicional que disminuye su accesibilidad en general. Ante esta realidad, surge la necesidad de llevar a cabo la mejora en la calidad de estas secuencias de imágenes por medio del procesamiento de la información que de ellas puede derivarse. Si bien este proceso no resulta tan robusto como lo sería con la incorporación de los datos de otros sensores, pueden obtenerse resultados satisfactorios, tal como lo evidencian Ratakonda [36], Guestrin et ál. [21] y Chen y Lovell [11], entre otros.

Considerando los aspectos anteriores, este trabajo tiene como objetivo principal la creación de una aplicación para estabilizar, en tiempo real, secuencias de imágenes capturadas desde un vehículo aéreo teledirigido. En particular, el enfoque se dirige a la estabilización *online* de imágenes transmitidas desde un helicóptero a control remoto.

La cámara se ubica rígidamente en el robot y éste se desplaza en el aire mientras se proyecta la escena 3D en el plano de las imágenes registradas. El vuelo normalmente se lleva a cabo desde una altura considerable, tal que se percibe un movimiento global entre imágenes consecutivas generado por el del dispositivo de captura. En el tiempo, este movimiento global suele describirse por medio

de dos componentes: movimiento intencional y no intencional. Tal como varios autores indican [14, 20, 29, 43], la primera se supone relativamente lenta y suave, mientras que la segunda representa el movimiento que se desea suprimir.

Es importante la selección de una transformación geométrica que explique lo mejor posible el movimiento global. Una vez encontrado, puede ser filtrado para lograr su descomposición. La estabilización finalmente consiste en compensar su componente no intencional.

La presencia de ruido en la secuencia de imágenes a procesar dificulta la tarea. Puede ocasionar errores significativos en la estimación de movimiento que, luego, degradan la calidad del video estabilizado.

La incorporación de sistemas de estabilización se evidencia hoy en día hasta en cámaras de bajo costo para una gama de consumidores amplia. La teoría que establece los fundamentos de esta tecnología puede ser aplicada al procesamiento de imágenes capturadas desde un vehículo aéreo, tomando en consideración el tipo de movimiento que éstos describen.

El cálculo del flujo óptico, o del movimiento de los píxeles entre imágenes consecutivas en una secuencia [10], forma la base de la aplicación de estabilización que tiene como objetivo este proyecto. Para obtener un aproximado de la transformación global se utilizan diferentes métodos de estimación, así como se consideran varios modelos de transformaciones geométricas en 2D. La estabilización, como fin último, se consigue por medio de una convolución, con una función gaussiana, de los valores en el tiempo de algunos parámetros que conforman la transformación. En particular, son considerados tanto el ángulo de rotación, como los desplazamientos horizontales y verticales que se evidencian de manera general entre las imágenes.

Algunos resultados intermedios del procesamiento pueden ser utilizados para llevar a cabo otras tareas. Por ejemplo, la compensación total del movimiento global facilita la detección de cuerpos u objetos móviles que se perciben a lo largo de la secuencia de imágenes, cuya identificación sin llevar a cabo esta fase puede resultar compleja [28, 7]. Más aún, a partir del seguimiento del desplazamiento de cuerpos independientes, se pueden aprender modelos de actividad terrestres desde una plataforma aérea [30].

De esta manera, la idea de aprovechar la estimación de movimiento con otros fines guía el desarrollo de este proyecto. Algunos autores han impuesto restricciones sobre el modelo de transformación que describe el movimiento, limitándolo por ejemplo a rotación, escalamiento isotrópico

y traslación [34, 49], a consecuencia de una disminución en la calidad de la estimación. En este trabajo se opta por estimar la mejor transformación posible en tiempo real, a la vez que se simplifica para llevar a cabo la estabilización. Este paso consiste en extraer de ella los parámetros que mejor describen la rotación y las traslaciones del movimiento global aproximado.

Poca evidencia existe sobre la influencia de las numerosas variables que determinan el desempeño del algoritmo de estimación de flujo óptico utilizado [31]. En reducidos casos se comparan el desempeño de varios modelos de transformación. Por lo tanto, en este trabajo fue necesario llevar a cabo pruebas que orienten sobre la selección de los valores de estas variables.

En la fase de experimentación se consideraron numerosos pares de imágenes sobre los cuales estimar el flujo óptico. A pesar de no conocer el valor real de los vectores de movimiento de los píxeles, conclusiones relevantes pudieron obtenerse. También se puso a prueba el sistema de estabilización ante diferentes tipos de movimientos capturados desde un helicóptero a control remoto. Las recomendaciones derivadas de esta fase se fundamentan principalmente en la necesidad de procesar rápidamente los videos capturados desde el aire.

Este informe consta de cinco capítulos. El Capítulo 2 primero describe de manera general el proceso de estabilización y, posteriormente, ofrece los fundamentos teóricos sobre las etapas que lo componen. El Capítulo 3 comprende el diseño de la aplicación implementada y detalla sus aspectos más relevantes. El Capítulo 4 describe los experimentos realizados, sus resultados y conclusiones particulares. Por último, en el Capítulo 5 se presentan las conclusiones de este trabajo, las recomendaciones y direcciones futuras.

Capítulo 2

Marco Teórico

En este capítulo se introduce el proceso de estabilización de imágenes capturadas desde una cámara en movimiento y se reseñan implementaciones en el procesamiento de video capturado desde un helicóptero teleoperado a control remoto.

Las imágenes digitales, en blanco y negro, se pueden representar matemáticamente por medio de una función $f(x, y)$ que indica la intensidad de luz en la coordenada (x, y) en el plano de la imagen. Considerando una cámara que posee un sensor que registra niveles de radiación, esta representación de una imagen se basa geométricamente en la proyección de cada punto de la escena en 3 dimensiones (3D), a través del centro del lente, en el plano de la imagen [41].

El **proceso de estabilización** de las imágenes de un video en un instante de tiempo puede resumirse en la estimación global del movimiento de una imagen a su sucesora y la compensación de este movimiento en cierto grado. El movimiento visible en la secuencia de imágenes puede ser causado tanto por el movimiento de la cámara como por el de cuerpos u objetos independientes en la escena. Más aún, el movimiento de la cámara puede ser intencional o no. Precisamente, el objetivo de estabilizar el video es remover la distorsión visual producida por el movimiento no intencional de la cámara en la secuencia. Si bien el movimiento de la cámara en 3D se conoce como *egomotion* [42], su cálculo no es necesario para la estabilización. Por el contrario, la estabilización se fundamenta en el cálculo del movimiento relativo entre imágenes consecutivas.

Pudiese considerarse el movimiento relativo a un marco de referencia estático. Esta referencia pudiese ser, por ejemplo, la primera imagen de la secuencia, dando así la impresión que la cámara no se mueve [20]. Sin embargo, este enfoque resulta impráctico para secuencias capturadas desde un helicóptero a control remoto. Esto se debe a que el movimiento de la cámara en el vehículo aéreo es considerable y el marco de referencia fácilmente no tendría relación con las imágenes capturadas transcurrido un corto tiempo.

El vector de movimiento para cada uno de los píxeles de una imagen se conoce como **flujo óptico** [10]. Verri y Poggio [48] lo definen como el **campo de movimiento** que se percibe gracias a variaciones temporales en los patrones de intensidad, que no siempre coincide con el movimiento real en 3D de la escena. Un objeto que se mueve en el mundo real podría reflejarse en una imagen

bajo un patrón de intensidad constante [23]. Por ejemplo, al rotar una esfera uniforme, el campo de movimiento percibido en ella es cero.

Únicamente las componentes del movimiento, horizontal o vertical, en la dirección del gradiente¹ local de la función de intensidad de la imagen pueden ser estimadas [6]. Esto implica que ambas a veces no pueden calcularse. Así se describe el **problema de apertura** en el cálculo de flujo óptico, puesto que sólo en aquellas regiones de una imagen donde existe suficiente estructura, el movimiento puede ser estimado completamente. De esta manera, por ejemplo, si se considera una región en una imagen con un patrón de intensidad que varía como función de “x” o “y”, pero no ambas; el movimiento del patrón en una dirección alterará la intensidad de la imagen en un punto particular, sin embargo, no se percibirá movimiento en la otra. Es decir, solo la componente vertical del movimiento se puede determinar dado un borde de intensidad horizontal [38].

Otro problema surge cuando algunos cuerpos u objetos proyectados en el plano de la imagen pudiesen también ocultar parcial o totalmente a otros. En esta situación y ante la presencia de movimiento en las imágenes, fácilmente algunos píxeles desaparecen mientras otros, quizás nuevos, ocupan su posición. Entonces, al buscar la correspondencia inexistente de estos puntos entre imágenes consecutivas, el vector de flujo óptico que describe su movimiento fácilmente es erróneo.

2.1. Descripción general del proceso de estabilización de video

El objetivo principal de este trabajo es la estabilización, en tiempo real, de imágenes capturadas desde un helicóptero teleoperado a control remoto. Con este propósito, dada una secuencia de imágenes, el primer paso para estabilizarla es estimar para cada instante de tiempo t el movimiento global entre las imágenes I^{t-1} e I^t . Se supone que este movimiento se debe principalmente al de la cámara y su estimación puede llevarse a cabo encontrando la mejor correspondencia geométrica posible entre las imágenes. La estimación de esta correspondencia entre una imagen y otra que sirve de base, o referencia, se conoce como *image registration* [20, 32]. Si consideramos I^t la imagen de referencia, entonces deseamos encontrar una transformación, basada en el modelo de movimiento global que se percibe, que aplicada a I^{t-1} minimice su diferencia con I^t . Un planteamiento equivalente, aunque menos popular, es considerar como referencia la imagen previa, I^{t-1} , y buscar transformar I^t [35, 33].

¹La definición matemática del gradiente se presenta en el Apéndice A.

Bien es conocido que el cálculo de flujo óptico denso (considerando cada uno de los píxeles) resulta complejo y sensible a ruido [13, 27]. Con el fin de acelerar el proceso de estimación de la transformación, en vez de considerar todos los píxeles posibles, se pueden seleccionar sólo algunos para el procesamiento, quizás utilizando información sobre el contenido de las imágenes.

De esta manera, podrían considerarse únicamente **rasgos sobresalientes**, o distintivos, para los que su correspondencia de I^{t-1} a I^t sirve de base para encontrar una aproximación del campo de movimiento. Adicionalmente, esta selección pudiese tener como objetivo adicional evitar el problema de apertura antes mencionado, tal que se descartan de la selección regiones en la imagen a transformar para las cuales se dificulta estimar ambas componentes de su movimiento. Aunque idealmente estos rasgos deben distribuirse a lo largo de toda la imagen, seleccionando únicamente entre aquellos que no se ubiquen en regiones que pudiesen ser difíciles de correlacionar, se disminuye la probabilidad de obtener una mala aproximación. Algunos rasgos sobresalientes que mencionan Zitova y Flusser [50] incluyen intersecciones de líneas, puntos de alto contraste, puntos de inflexión en curvas, esquinas, entre otros.

Vale destacar que la estrategia de simplificación de las imágenes en el proceso de *image registration* es popular. Algunos autores optan por seleccionar regiones de las imágenes sobre las cuales llevar a cabo el procesamiento basándose en alguna heurística. Tal es el caso, por ejemplo, del sistema de estabilización implementado por Morimoto y Chellappa [35], para el cual resulta conveniente seleccionar puntos en el horizonte con el fin de disminuir las vibraciones en videos capturados desde un vehículo que recorre un terreno irregular. Con la misma meta, Duric y Rosenfeld [14] se guían por el horizonte para seleccionar puntos distantes que facilitan determinar la rotación entre imágenes. Por otro lado, también pueden ser utilizados procedimientos de detección de bordes para descartar regiones de bajo contraste [11, 9].

La selección de rasgos puede llevarse a cabo durante la estabilización de una secuencia cada cierto tiempo. Parece conveniente reutilizar rasgos entre imágenes que no difieren significativamente. Por ejemplo, si se tienen los rasgos sobresalientes de la imagen I^{t-1} , su correspondencia en I^t pudiese considerarse el conjunto de rasgos en la imagen en tiempo t . Dado que el proceso de estabilización supone que la cámara no está estática de manera permanente, eventualmente los rasgos detectados inicialmente no serán útiles para el procesamiento. Esto se debe a que posiblemente irán quedando fuera del plano de las imágenes que se estabilizan y no aportarán ninguna información útil para

el cálculo del movimiento entre éstas. Así mismo, su propiedad de sobresalientes pudiese verse afectada por errores en su correspondencia de una imagen a otra. En consecuencia, la calidad de la estimación pudiese disminuir considerablemente.

Sin embargo, ésta pareciera ser la metodología implementada por Jung y Sukhatme [27], quienes presentan los mismos rasgos correlacionados al principio y al final de una secuencia de 30 imágenes capturadas desde un robot móvil. A pesar de no incluir detalles sobre el mecanismo de selección de nuevos rasgos que debe ocurrir de vez en cuando, sí destacan que los rasgos erróneamente correlacionados en el borde de las imágenes no son considerados en procesamientos posteriores. De manera similar, la correlación de un mismo conjunto de rasgos en [30] se lleva a cabo a lo largo de varias imágenes. En estos casos las secuencias son procesadas directamente desde computadoras embebidas en los robots que poseen la cámara y, por lo tanto, éstas se caracterizan por no estar expuestas a ruido producto de transmisiones de video inalámbrico.

Más conservadora ante la posibilidad de incurrir en errores, resulta la implementación del sistema automático de estabilización de imágenes digitales de Morimoto y Chellappa [35], donde se opta por detectar rasgos de manera alternada. En otras palabras, se seleccionan cada dos imágenes, tal que los rasgos sobresalientes encontrados en un instante de tiempo t son utilizados para estimar el movimiento entre las imágenes I^{t-1} e I^t y entre I^t e I^{t+1} .

El campo de movimiento que se estima usualmente es diverso, en el sentido que puede incluir tanto movimientos de objetos independientes, como aquellos producto del movimiento de la cámara. Por ello, el proceso de estimación de la transformación geométrica debe ser lo menos sensible posible a los movimientos independientes ya mencionados. Afortunadamente, en secuencias de imágenes capturadas desde un helicóptero a control remoto, generalmente el movimiento de la cámara se evidencia ampliamente en el campo de movimiento. Esto se debe a que las imágenes usualmente son capturadas desde cierta altura, ante la cual los movimientos independientes resultan atípicos por el reducido espacio en los planos de la imágenes que ocupan los cuerpos u objetos que los producen. Sin embargo, también es común que estas imágenes posean altos niveles de ruido. El ruido puede ser producto, por ejemplo, de fallas en la transmisión analógica de las imágenes desde una cámara en el vehículo aéreo hasta el lugar de procesamiento. Esta distorsión en las imágenes muchas veces dificulta la selección de rasgos sobresalientes, a la vez que conlleva a errores en su correlación. Ello se evidencia en una mala estimación del campo de movimiento, tal como indica

Johansen [26]. Errores en esta fase del proceso, posteriormente, conllevan a la obtención de una transformación que poco explica el movimiento global entre imágenes. Si el estimado del modelo utilizado no es bueno, posiblemente la estabilización tampoco lo será y se percibirán saltos en el video procesado [29].

Para finalmente estabilizar la secuencia, resulta común filtrar el movimiento global estimado, tal que de él se deriva un movimiento suave y lento que se supone intencional al momento de su captura [14, 20, 29, 43]. De la diferencia entre estos dos estimados, se calcula el movimiento vibratorio, o impulsivo, que debe ser compensado.

2.2. Detección y estimación del movimiento de “buenos” rasgos

Para encontrar una estimación del campo de movimiento en secuencias de imágenes es bien conocido el uso del algoritmo de Lucas y Kanade [31], tal como se evidencia en [7], [10], [27], [33] y [30].

Sean $g(\vec{x})$ y $f(\vec{x})$ las funciones que representan la intensidad de las imágenes I^{t-1} e I^t en escala de grises, respectivamente. Siendo $\vec{x} = (x, y)$, tal que $\vec{x} \in W$ para una cierta región W ; se desea encontrar \vec{h} tal que $f(\vec{x} + \vec{h})$ aproxima a $g(\vec{x})$. \vec{h} representa la cantidad de movimiento de una imagen a otra para los píxeles pertenecientes a W .

De esta manera, el algoritmo de Lucas y Kanade en su caso más simple limita el movimiento de estos píxeles a translación. Así mismo, no considera un único píxel, sino una región W , sobre la cual estimar su correlación en las imágenes. Esto se debe a que si se estimara sólo el movimiento de un píxel, éste debería ser suficientemente distingible con respecto a todos sus vecinos y, más aún, el procedimiento de estimación fácilmente se vería afectado por la presencia de ruido.

Para calcular el error de la aproximación se utiliza la diferencia entre $f(\vec{x} + \vec{h})$ y $g(\vec{x})$ tal que se busca minimizar

$$E = \sum_{\vec{x} \in W} [f(\vec{x} + \vec{h}) - g(\vec{x})]^2 \quad (1)$$

en función de \vec{h} . Para ello se deriva el error E en función de h_x y h_y tal que

$$\frac{\partial E}{\partial \vec{h}} = \begin{bmatrix} \frac{\partial E}{\partial h_x} & \frac{\partial E}{\partial h_y} \end{bmatrix}^T = \begin{bmatrix} 0 & 0 \end{bmatrix}^T \quad (2)$$

El valor de $f(\vec{x} + \vec{h})$ puede ser estimado a partir de la expansión de Taylor de la función de

intensidad², truncada al término lineal, tal que para $|\vec{h}|$ pequeño se tiene

$$f(\vec{x} + \vec{h}) \approx f(\vec{x}) + \nabla f \cdot \vec{h} \quad (3)$$

Esta aproximación puede utilizarse para minimizar el error E en función de \vec{h} . En particular, para $\frac{\partial E}{\partial h_x}$, sustituyendo (3) en (1) y derivando según (2),

$$\begin{aligned} 0 = \frac{\partial E}{\partial h_x} &\approx \frac{\partial}{\partial h_x} \sum_{\vec{x} \in W} [f(\vec{x}) + \nabla f \cdot \vec{h} - g(\vec{x})]^2 = 2 \sum_{\vec{x} \in W} (f(\vec{x}) + \nabla f \cdot \vec{h} - g(\vec{x})) \frac{\partial}{\partial h_x} (\nabla f \cdot \vec{h}) \\ &= 2 \sum_{\vec{x} \in W} (f(\vec{x}) + \nabla f \cdot \vec{h} - g(\vec{x})) \frac{\partial}{\partial h_x} \left(\frac{\partial f}{\partial x} h_x + \frac{\partial f}{\partial y} h_y \right) = 2 \sum_{\vec{x} \in W} (f(\vec{x}) + \nabla f \cdot \vec{h} - g(\vec{x})) \frac{\partial f}{\partial x} \end{aligned}$$

Por lo tanto,

$$\sum_{\vec{x} \in W} \frac{\partial f}{\partial x} \left(\frac{\partial f}{\partial x} h_x + \frac{\partial f}{\partial y} h_y \right) \approx \sum_{\vec{x} \in W} \frac{\partial f}{\partial x} (g(\vec{x}) - f(\vec{x}))$$

De manera similar, si se busca minimizar la derivada parcial en función de h_y se obtiene

$$\sum_{\vec{x} \in W} \frac{\partial f}{\partial y} \left(\frac{\partial f}{\partial x} h_x + \frac{\partial f}{\partial y} h_y \right) \approx \sum_{\vec{x} \in W} \frac{\partial f}{\partial y} (g(\vec{x}) - f(\vec{x}))$$

A partir de las ecuaciones anteriores se puede reescribir el sistema para estimar \vec{h} de la siguiente manera,

$$\begin{bmatrix} \sum_{\vec{x} \in W} \frac{\partial f}{\partial x}^2 & \sum_{\vec{x} \in W} \frac{\partial f}{\partial x} \frac{\partial f}{\partial y} \\ \sum_{\vec{x} \in W} \frac{\partial f}{\partial y} \frac{\partial f}{\partial x} & \sum_{\vec{x} \in W} \frac{\partial f}{\partial y}^2 \end{bmatrix} \begin{bmatrix} h_x \\ h_y \end{bmatrix} \approx \begin{bmatrix} \sum_{\vec{x} \in W} \frac{\partial f}{\partial x} (g(\vec{x}) - f(\vec{x})) \\ \sum_{\vec{x} \in W} \frac{\partial f}{\partial y} (g(\vec{x}) - f(\vec{x})) \end{bmatrix} \quad (4)$$

Entonces, renombrando las matrices que componen el sistema tal que se tiene $Z\vec{h} \approx \vec{e}$, la solución puede aproximarse considerando

$$\vec{h} \approx Z^{-1} \vec{e} \quad (5)$$

Este planteamiento se conoce como la ecuación estándar para el cálculo de flujo óptico de Lucas y Kanade y es válida para \vec{h} pequeño.

²En el Apéndice A se describe la expansión de Taylor de una función de dos variables.

2.2.1. Interpretación geométrica

Tomasi y Kanade [44] explican una forma más intuitiva para la derivación de este planteamiento. Podemos considerar la función de intensidad de la imagen $g(\vec{x})$ en cierta región W , como se muestra en la Figura 1(a), y una copia superpuesta sobre ella, tal que no hay espacio entre las dos superficies de intensidad. Si desplazamos la copia llevando a cabo una traslación horizontal de pequeña magnitud, entonces se formará una brecha entre estas dos superficies. Esta última que fue desplazada podemos considerarla como $f(\vec{x} + \vec{h})$, tal que \vec{h} representa el desplazamiento.³

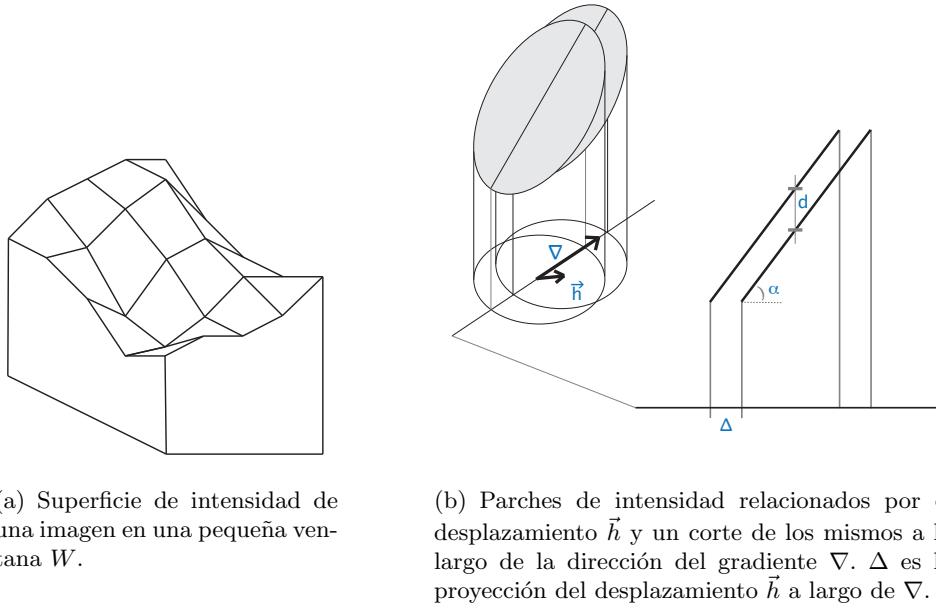


Figura 1: Relaciones geométricas entre los de parches de intensidad de dos imágenes en función de un desplazamiento \vec{h} entre éstos. Figuras tomadas de [44].

El ancho de la brecha, medido horizontalmente, puede expresarse como una función del desplazamiento entre los dos parches de intensidad. Medido verticalmente, el ancho de la brecha es la diferencia entre los valores de intensidad de las superficies.

Siguiendo la Figura 1(b), podemos enfocarnos en un pequeño parche de la superficie de intensidad $g(\vec{x})$, en conjunto al correspondiente de $f(\vec{x} + \vec{h})$ que resulta de la traslación. El vector de desplazamiento \vec{h} generalmente no sigue la misma dirección del gradiente de f , indicado como ∇ en la figura. Utilizando la definición de vector unitario que se presenta en la ecuación (39) en el

³En este caso simple, para \vec{x} definido en la región W , $f(\vec{x} + \vec{h}) = g(\vec{x})$. Los píxeles desplazados en f poseen el mismo valor de intensidad que aquellos en g . En la práctica esto raramente sucede, pero el proceso de estimación busca encontrar la mejor aproximación de \vec{h} tal que se minimiza el error entre $g(\vec{x})$ y $f(\vec{x} + \vec{h})$. En este caso, $g(\vec{x}) = f(\vec{x} + \vec{h}) + n(\vec{x})$, donde n representa ruido o variaciones en la intensidad de las imágenes.

Apéndice A, podemos descomponer este gradiente como $\nabla = |\nabla|\hat{\nabla}$, donde $|\nabla|$ indica su magnitud y $\hat{\nabla}$ su dirección. Luego, el desplazamiento Δ a lo largo de la dirección del gradiente ∇ es la proyección de \vec{h} en $\hat{\nabla}$, tal que

$$\Delta = \vec{h} \cdot \hat{\nabla} \quad (6)$$

Considerando entre los parches

$$d = g(\vec{x}) - f(\vec{x}) \quad (7)$$

también se cumple

$$d = \Delta \tan \alpha \quad (8)$$

siendo α la máxima pendiente de los parches. Utilizando (8) y (6) y puesto que la tangente de α es igual a la magnitud del gradiente, $\tan \alpha = |\nabla|$, se puede expresar el ancho d de la brecha en función del desplazamiento \vec{h} :

$$d = \Delta |\nabla| = (\vec{h} \cdot \hat{\nabla}) |\nabla| = \vec{h} \cdot (\hat{\nabla} |\nabla|) = \vec{h} \cdot \nabla \quad (9)$$

Esta ecuación considera las dos incógnitas h_x y h_y que definen \vec{h} . El gradiente ∇ puede ser estimado de $g(\vec{x})$, mientras que la diferencia d puede computarse considerando la intensidad ambos parches. De manera detallada, $d = [h_x \ h_y] [\frac{\partial f}{\partial x} \ \frac{\partial f}{\partial y}]^T$, tal que el problema de apertura puede observarse claramente. Si sólo consideramos parches para resolver el sistema (9), fácilmente ocurre que no se puede determinar el desplazamiento \vec{h} en su totalidad, sino únicamente una de sus componentes. Por ejemplo, si la intensidad en estos parches correspondientes varía sólo verticalmente, tendremos que $\frac{\partial f}{\partial x} = 0$ y, por lo tanto,

$$d = \begin{bmatrix} h_x & h_y \end{bmatrix} \begin{bmatrix} 0 & \frac{\partial f}{\partial y} \end{bmatrix}^T = h_y \left(\frac{\partial f}{\partial y} \right)$$

De esta manera, h_x no puede ser determinada. Si por el contrario consideramos la región W completa, diferentes parches nos permitirán computar diferentes componentes del desplazamiento, suponiendo que éste es constante en W . Podemos observar que si \vec{h} es estimado erróneamente, existirá diferencia entre los extremos derecho e izquierdo de la ecuación (9). Por lo tanto, la mejor aproximación de \vec{h} en W puede obtenerse minimizando el cuadrado de esa diferencia para los diferentes parches. Matemáticamente, si cada parche corresponde a un píxel, podemos plantear este

error de manera discreta utilizando (7), tal que

$$E = \sum_{\vec{x} \in W} [\vec{h} \cdot \nabla - d]^2 = \sum_{\vec{x} \in W} [\vec{h} \cdot \nabla - (g(\vec{x}) - f(\vec{x}))]^2 = \sum_{\vec{x} \in W} [f(\vec{x}) + \nabla \cdot \vec{h} - g(\vec{x})]^2 \quad (10)$$

De esta manera, puesto que \vec{h} es pequeño, podemos utilizar (3) y reescribir el resultado como

$$E = \sum_{\vec{x} \in W} [f(\vec{x} + \vec{h}) - g(\vec{x})]^2$$

siendo éste el mismo planteamiento inicial de (1).

Vale mencionar qué sucede cuando el desplazamiento \vec{h} se lleva a cabo en dirección opuesta al gradiente. Si éste es el caso, considerando las funciones de intensidad de los parches, se tiene $f(\vec{x}) > g(\vec{x})$. Por ello, siendo d un escalar positivo, debe definirse ahora como $d = f(\vec{x}) - g(\vec{x})$. La proyección de \vec{h} a lo largo de la dirección del gradiente será un número negativo de igual valor absoluto a Δ , por lo que la ecuación (6) debe ser modificada por $\Delta = -\vec{h} \cdot \hat{\nabla}$. Siguiendo el mismo razonamiento, (9) ahora se expresa como

$$d = \Delta |\nabla| = (-\vec{h} \cdot \hat{\nabla}) |\nabla| = -\vec{h} \cdot (\hat{\nabla} |\nabla|) = -\vec{h} \cdot \nabla \quad (11)$$

Finalmente, el equivalente al error de (10) que se desea minimizar nos lleva al mismo resultado:

$$E = \sum_{\vec{x} \in W} [d + \vec{h} \cdot \nabla]^2 = \sum_{\vec{x} \in W} [(f(\vec{x}) - g(\vec{x})) + \vec{h} \cdot \nabla]^2 = \sum_{\vec{x} \in W} [f(\vec{x}) + \nabla \cdot \vec{h} - g(\vec{x})]^2 \quad (12)$$

2.2.2. Refinamiento iterativo

Para obtener una buena aproximación del desplazamiento, se puede iterar de manera tal que se desplaza $f(\vec{x})$ en función de \vec{h} y se repite el procedimiento de estimación. De esta manera, considerando el sistema (4),

$$\vec{h}_o = 0, \quad (13)$$

$$\vec{h}_{k+1} = h_k + \begin{bmatrix} \sum_{\vec{x} \in W_k} \frac{\partial f}{\partial x}^2 & \sum_{\vec{x} \in W_k} \frac{\partial f}{\partial x} \frac{\partial f}{\partial y} \\ \sum_{\vec{x} \in W_k} \frac{\partial f}{\partial y} \frac{\partial f}{\partial x} & \sum_{\vec{x} \in W_k} \frac{\partial f}{\partial y}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{\vec{x} \in W_k} \frac{\partial f}{\partial x} (g(\vec{x}) - f(\vec{x})) \\ \sum_{\vec{x} \in W_k} \frac{\partial f}{\partial y} (g(\vec{x}) - f(\vec{x})) \end{bmatrix} \quad (14)$$

donde la región W_k cumple ($\forall \vec{x}, k : \vec{x} \in W_0 \wedge 0 < k : \vec{x} + \vec{h}_{k-1} \in W_k$). Idealmente, la secuencia de estimados de \vec{h} convergerá al mejor valor.

2.2.3. Campo de movimiento afín

El planteamiento anterior puede extenderse para la correlación de bloques cuyo movimiento se modela no sólo por traslación, sino por una transformación lineal arbitraria [31, 38]. En este caso, la aproximación entre f y g está dada por $f(A\vec{x} + \vec{h}) \approx g(\vec{x})$, donde \vec{h} sigue representando el desplazamiento y $A = 1 + D$, con 1 la matriz identidad de tamaño 2×2 y D la matriz de deformación

$$D = \begin{bmatrix} d_{xx} & d_{xy} \\ d_{yx} & d_{yy} \end{bmatrix} \quad (15)$$

En otras palabras, ahora la descripción del movimiento está dada por un campo de movimiento afín⁴ en W , tal que $\vec{u} = D\vec{x} + \vec{h}$. En este caso se tienen como incógnitas los 6 parámetros que aparecen en la matriz de deformación D y el vector de desplazamiento \vec{h} . De esta manera,

$$f(A\vec{x} + \vec{h}) = f((1 + D)\vec{x} + \vec{h}) = f(\vec{x} + (D\vec{x} + \vec{h})) = f(\vec{x} + \vec{u}) \quad (16)$$

El modelo antes expuesto que restringe el movimiento únicamente a traslación puede verse como una particularización del modelo afín. Esto equivale a que la matriz A sea la identidad, de tamaño 2×2 , de manera tal que $A\vec{x} + \vec{h} = \vec{x} + \vec{h}$.

2.2.4. Selección de rasgos

La calidad del movimiento estimado depende en parte del tamaño de la región W , así como de la cantidad de movimiento entre las imágenes. Para una región pequeña es difícil estimar los 4 parámetros de D , porque las variaciones de movimiento de los puntos en W también son pequeñas y, por lo tanto, menos confiables. Sin embargo, ventanas más pequeñas en general son preferibles a otras grandes, porque es probable que los puntos que la conforman estén a la misma profundidad en la escena. En [38] los experimentos indican que la mejor combinación de los dos modelos para estimar el movimiento es la utilización de traslación para seguir los puntos de una imagen a su

⁴Se considera un campo de movimiento afín como aquel representado por una transformación afín. En el Apéndice A se incluye la definición de este tipo de transformación.

sucesora, y la aplicación de un modelo afín para comparar los rasgos seguidos entre imágenes distantes en la secuencia.

Aquellas regiones W que resultan apropiadas para llevar a cabo la estimación de movimiento serán consideradas “**buenos**” rasgos. Dado que el algoritmo de Lucas y Kanade [31] se basa en resolver el sistema de la ecuación (4), así como su equivalente en el caso de un campo de movimiento afín, la matriz simétrica

$$Z = \begin{bmatrix} \sum_{\vec{x} \in W} \frac{\partial f^2}{\partial x} & \sum_{\vec{x} \in W} \frac{\partial f}{\partial x} \frac{\partial f}{\partial y} \\ \sum_{\vec{x} \in W} \frac{\partial f}{\partial y} \frac{\partial f}{\partial x} & \sum_{\vec{x} \in W} \frac{\partial f^2}{\partial y} \end{bmatrix} \quad (17)$$

debe ser invertible y estar bien condicionada.

Una matriz está bien condicionada si su número de condición es pequeño. Siguiendo la definición de número de condición que se presenta en el Apéndice A, si se utiliza la norma l_2 , entonces el número de condición de la matriz simétrica e invertible Z puede calcularse como

$$\text{num_cond}(Z) = \|Z\|_2 \|Z^{-1}\|_2 = \frac{\lambda_{\max}(Z)}{\lambda_{\min}(Z)} \quad (18)$$

siendo $\lambda_{\max}(Z)$ y $\lambda_{\min}(Z)$ los dos autovalores asociados a Z .⁵

El número de condición de Z cumple $\text{num_cond}(Z) \geq 1$, por lo cual esta matriz está bien condicionada si éste se acerca a 1. De esta manera, el requerimiento sobre el número de condición asociado a la matriz Z de un “buen” rasgo, exige que sus autovalores no difieran en varios órdenes de magnitud [38].

Adicionalmente, Z debe sobreponerse al nivel de ruido presente en la imagen a transformar, lo cual implica que ambos autovalores deben poseer un valor alto. Esto se debe a que dos autovalores pequeños indican un perfil de intensidad relativamente constante en W . Uno grande y otro pequeño corresponden a patrones de textura unidireccionales. Finalmente, dos grandes representan esquinas o patrones distintivos que facilitan la estimación del movimiento.

Bouget [8] propone un método automatizado para la selección de los “buenos” rasgos. Considerando un píxel con posición (i, j) en una imagen, se define su **nivel de calidad** como el mínimo autovalor de la matriz Z , según (17), en la vecindad $W_{(i,j)}$ centrada en su coordenada. Dada una imagen, el procedimiento primero computa el nivel de calidad de todos los píxeles que la conforman.

⁵En el Apéndice A se ofrece la definición de un autovalor tal como se utiliza en estos planteamientos.

De esta manera, se registra el mínimo autovalor $\lambda_{(i,j)}$, dado $W_{(i,j)}$. Se determina entre todos éstos el máximo de los mínimos autovalores, definiendo $\lambda_{\max} = \max(\lambda_{(i,j)})$. Se establece como mínimo nivel de calidad aceptado para los rasgos sobresalientes un porcentaje de este máximo, tal que se considera un “buen” rasgo aquel píxel para el cual se cumple $\lambda_{(i,j)} > k\lambda_{\max}$, con $0 \leq k \leq 1$. De aquellos rasgos que poseen un nivel de calidad aceptado, se retienen únicamente los máximos locales. Estos máximos locales se identifican como aquellos píxeles (i,j) que satisfacen $\lambda_{(i,j)} > \lambda_{(k,p)}$, tal que $i-1 \leq k \leq i+1$ con $k \neq i$ y $j-1 \leq p \leq j+1$ con $p \neq j$. En otras palabras, un píxel se retiene si su mínimo autovalor asociado es mayor al de cualquier otro píxel en su vecindad de 3×3 . Finalmente, se crea el conjunto de “buenos” rasgos seleccionados a partir de los que fueron retenidos, tal que se cumple que la mínima distancia entre cualquier par de píxeles que le pertenezcan es mayor a un valor predeterminado.

Previamente se había utilizado un razonamiento similar en un proceso de segmentación de movimiento. Irani et ál. [25] definen un nivel de “fiabilidad” sobre un píxel, considerando una vecindad dada y la ecuación para el cálculo de flujo óptico de Lucas y Kanade [31]. Este nivel de fiabilidad, $R(x,y)$, lo expresan como el inverso del número de condición de la matriz de coeficientes de (4), tal que

$$R(x,y) = \frac{\lambda_{\min}}{\lambda_{\max}} \quad (19)$$

siendo λ_{\min} y λ_{\max} el mínimo y el máximo autovalor de Z según (17), respectivamente.

Si definimos un nivel de fiabilidad a partir de cierto umbral k , tal que se considera que un píxel (x,y) posee un nivel de fiabilidad alto si $R(x,y) > k$; entonces, este nivel de fiabilidad es equivalente al nivel de calidad propuesto por Bouget [8]: $(\lambda_{\min}/\lambda_{\max}) > k \equiv \lambda_{\min} > k\lambda_{\max}$.

2.2.5. Método Piramidal

Una **pirámide** puede definirse como una colección de representaciones de una imagen [17]. En particular podemos considerar representaciones multiescalas, tal como propone utilizar Bouget [8] para mejorar la robustez del algoritmo de Lucas y Kanade [31].

El proceso de estimación de flujo óptico, considerando la ecuación (4) y su posible refinamiento iterativo, se beneficia de la utilización de regiones W de mayor tamaño a la magnitud del desplazamiento. Si se reduce el tamaño de las imágenes a procesar, manteniendo constante el tamaño de W , entonces se puede obtener una primera aproximación del movimiento de cierto rasgo. Esta estima-

ción sirve de base para su posterior refinamiento considerando las imágenes en una escala mayor. Suponiendo que un máximo desplazamiento de h_{\max}^0 píxeles puede ser estimado por el algoritmo en su implementación simple para las imágenes en sus dimensiones originales, según [8], el máximo desplazamiento h_{\max}^l que se puede estimar en el nivel l de una pirámide, tal que $0 \leq l$, puede calcularse por $h_{\max}^l = (2^{l+1} - 1)h_{\max}^0$. De esta manera se puede utilizar una región W pequeña y, a la vez, estimar movimientos de magnitud considerable. Por ejemplo, para una pirámide compuesta por 4 niveles se tiene una ganancia de 15 veces la magnitud del máximo desplazamiento que puede ser estimado.

2.3. Estimación del movimiento global

Los aspectos más relevantes en el proceso de estimación del movimiento global que se percibe en una secuencia de imágenes son el tipo de transformación utilizada y el método de estimación.

2.3.1. Transformaciones

La transformación en 2D que modela el movimiento de la cámara debe estar sujeta al tipo de deformación esperada en la secuencia. Ante la presencia de movimientos atípicos o errores en la estimación del flujo óptico, la utilización de transformaciones más flexibles tiende a disminuir la robustez del proceso de estimación del movimiento global. Esto se debe a que las transformaciones pueden ajustarse fácilmente a los movimientos detectados en las imágenes que no son producto del movimiento de la cámara [27].

Transformación afín

En su forma más general, esta transformación permite representar rotaciones respecto al centro de coordenadas, escalamientos, traslaciones, inclinaciones y hasta reflexiones a lo largo de los ejes x e y . Cuando el tiempo que transcurre entre la captura de imágenes consecutivas es pequeño, parece razonable utilizar este tipo de transformación para modelar el movimiento percibido desde un helicóptero [30, 27].

Matemáticamente⁶, dado un punto $\mathbf{p} = [x \ y]^T$,

$$T_a(\mathbf{p}) = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_5 \\ a_6 \end{bmatrix} \quad (20)$$

La transformación afín T_a se caracteriza por preservar la colinealidad de puntos y la proporción de distancia a lo largo de una línea. En otras palabras, todos los puntos pertenecientes a una recta inicialmente lo seguirán haciendo posterior a la transformación; y el punto medio de un segmento de recta lo seguirá siendo luego de la aplicación de T_a .

La transformación afín podría restringirse a traslación, rotación y un mismo escalamiento horizontal y vertical. De esta manera, considerando de nuevo $\mathbf{p} = [x \ y]^T$, la transformación reduce su número de parámetros a 4 y puede expresarse por

$$T_s(\mathbf{p}) = \begin{bmatrix} a_1 & -a_2 \\ a_2 & a_1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_3 \\ a_4 \end{bmatrix} = s \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (21)$$

Renombrando los parámetros de T_s en la ecuación anterior, podemos identificar s como aquel responsable del escalamiento, t_x y t_y los encargados de la traslación horizontal y vertical, respectivamente, y θ el ángulo de rotación relativo al origen de coordenadas.

El uso de este tipo de transformación parece ser popular. Guestrin et ál. [20] la consideran bajo un modelo de movimiento “similar”. Morimoto y Chellappa [35] la utilizan para la implementación de un sistema de estabilización que soporta desplazamientos hasta de 21 píxeles entre imágenes consecutivas. Johansen [26] pone a prueba este modelo para estabilizar video capturado desde un helicóptero autónomo. También es el tipo de transformación empleado por Chang et ál. [10] y Estalayo et ál. [15].

Si se considera $s = 1$, entonces se tiene un modelo “rígido”. Su utilización se evidencia en sistemas de estabilización que suponen como movimiento principal de la cámara traslación y, adicionalmente, buscan poder representar rotaciones [20, 11].

El modelo de transformación podría considerar sólo traslación, tal como se pone en práctica

⁶Esta formulación de una transformación afín resulta de la expansión detallada de la de la ecuación (42) en el Apéndice A.

en [36], para reducir vibraciones que se perciben en el plano de las imágenes de una secuencia de video. Más aún, podría considerar únicamente traslaciones verticales [9]. Sin embargo, existe evidencia que indica que con un modelo tan restringido se dificulta el proceso de detección de vectores de movimiento atípicos en secuencias capturadas desde un helicóptero a control remoto, utilizando mínimos cuadrados o RANSAC como método de estimación iterativo [26].⁷

Transformación bilineal

Cuando la diferencia entre las imágenes consecutivas crece, posiblemente por una limitada velocidad de procesamiento en ciertas circunstancias, una transformación afín parece no ser suficiente. Este es el caso de [27], donde se utiliza una transformación bilineal para modelar el movimiento percibido desde una cámara en varios robots móviles, incluyendo un helicóptero autónomo. De manera similar, dado $\mathbf{p} = [x \ y]^T$, esta transformación se define por

$$T_b(\mathbf{p}) = \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_5 \\ b_6 \end{bmatrix} + \begin{bmatrix} b_7 * x * y \\ b_8 * x * y \end{bmatrix} \quad (22)$$

A diferencia de las transformaciones afines, T_b no posee la propiedad de preservar la colinealidad de puntos.

2.3.2. Estimación de la transformación

La estimación de los parámetros de la transformación seleccionada debe tener como fin encontrar una representación del movimiento global. Para ello se puede utilizar un **enfoque diagnóstico** de rasgos atípicos o *outliers* [22], tal que, una vez detectados, estos rasgos se eliminan de procesamientos posteriores. Un método para identificar los vectores de movimiento infrecuentes del flujo óptico puede fundamentarse en la comparación del nivel de influencia de éstos en la estimación de la transformación. Si se logran identificar estos vectores, la transformación pudiese ser estimada de manera iterativa sin considerarlos, tal como se plantea en el proceso de detección de *outliers* de Torr y Murray [45]. Sin embargo, este procedimiento resulta costoso puesto que su base teórica exige que los rasgos atípicos sean detectados uno por uno.

Es popular el uso de iterativo de mínimos cuadrados para estimar la transformación. Se trata

⁷Detalles sobre mínimos cuadrados y RANSAC, como métodos de estimación de la transformación que modela el movimiento global, se ofrecen en la sección 2.3.2.

de obtener una primera aproximación utilizando todos los vectores que representan la velocidad de los rasgos sobresalientes y, posteriormente, ir refinando el modelo eliminando del procesamiento aquellos rasgos que poseen un error mayor a un valor preestablecido. Así como expresan Golub y Loan [19], si se considera un conjunto de datos en una matriz A de tamaño $m \times n$ y un vector de observaciones b de m componentes, el problema de **regresión lineal múltiple de mínimos cuadrados** se basa en encontrar \hat{x} , de dimensiones $n \times 1$, para solucionar el sistema sobre determinado $Ax = b$, tal que

$$\hat{x} = (\min x : x \in \mathbb{R}^n : \|b - Ax\|_2) \quad (23)$$

En este caso $\|\cdot\|_2$ denota la norma Euclídea o l_2 ⁸.

Uno de los métodos más populares para encontrar la solución aproximada de un problema de regresión lineal múltiple por mínimos cuadrados, consiste en derivar el error de la solución según (23) en función de \hat{x} e igualar a cero para obtener la ecuación normal $\hat{x} = (A^T A)^{-1} A^T b$.

En general, mínimos cuadrados ha sido utilizado por ejemplo, por Morimoto y Chellappa [34], para estabilizar imágenes capturadas desde un vehículo. Más aún, según Guestrin et ál. [20], también se ha utilizado para minimizar la diferencia de intensidad entre las imágenes, encontrando una solución al problema de superposición sin estimar el campo de movimiento. Este planteamiento se basa en intentar evitar errores producto de la estimación del flujo óptico, sin embargo reportan un mayor tiempo de cómputo comparado con la implementación de Morimoto y Chellappa.

Chang et ál. [10] utilizan mínimos cuadrados de manera iterativa. En particular, la detección de *outliers* se basa en la desviación estándar⁹ del error del ajuste de una primera estimación.

La regresión lineal múltiple antes mencionada se ha puesto en práctica para compensar videos capturados desde un vehículo aéreo. Tal es el caso de Jung y Sukhatme [27], quienes la utilizan para obtener la transformación de una imagen a su sucesora con el fin de facilitar el proceso de detección y seguimiento de cuerpos u objetos que se mueven independientemente en las secuencias. En este caso, con el uso de un umbral predeterminado para los errores que se derivan de la diferencia entre aplicar la transformación a cada rasgo y su posición según el vector de flujo óptico en la imagen que sirve de referencia, se determinan aquellos que son considerados atípicos.

Otra opción ante la sensibilidad de mínimos cuadrados parece ser la propuesta por Bell et ál.

⁸La formulación matemática de esta norma se presenta en el Apéndice A.

⁹La formulación matemática de la desviación estándar puede derivarse fácilmente de la definición de varianza que se presenta en el Apéndice A.

[7], quienes plantean utilizar mínimos cuadrados ponderados iterativamente¹⁰. Este proceso de minimización consiste en agregar un valor de peso a los errores, tal que

$$\hat{x} = (\min x : x \in \mathbb{R}^n : \|D(b - Ax)\|_2) \quad (24)$$

D sirve para ponderar y posee dimensiones $m \times m$, siendo una matriz diagonal y no singular.¹¹

Otro método para estimar la transformación global a partir del conjunto de vectores que representan el campo de movimiento se conoce como **RANSAC**, diminutivo para *Random Sample Consensus* [16]. Dado un modelo que se desea ajustar a un conjunto P de puntos, se selecciona de manera aleatoria, o siguiendo alguna heurística aplicable al problema, un subconjunto $S1$ de datos. La cardinalidad de este subconjunto es mínima, tal que con los puntos que le pertenecen se puede instanciar el modelo. Usando una primera solución, se determina un conjunto $S1^*$ de datos en P para los cuales su error con respecto al modelo instanciado es menor a un cierto valor de tolerancia.

Si la cantidad de puntos de $S1^*$ es mayor a un mínimo umbral u preestablecido, entonces $S1^*$ se utiliza para estimar el modelo final. Esto puede llevarse a cabo, por ejemplo, utilizando mínimos cuadrados.

Si la cardinalidad de $S1^*$ es menor al umbral, entonces se selecciona un nuevo subconjunto de P , $S2$, para el cual se repite el proceso. Si después de un número máximo de intentos no se obtiene un subconjunto $S2^*$ de por lo menos u elementos para estimar el modelo, bien se utiliza el de mayor cardinalidad encontrado o se termina el procedimiento sin reportar una solución.

Bajo un modelo de transformación similar y utilizando una cámara posicionada rígidamente en un helicóptero, Johansen [26] no encontró ventajas significativas con el uso de este paradigma, en comparación a mínimos cuadrados iterativos. Adicionalmente indica que los tiempos de cómputo obtenidos al utilizar RANSAC superan ampliamente a los obtenidos con mínimos cuadrados.

¹⁰Bell et ál. [7] no ofrecen pruebas que den soporte a su argumento sobre la mayor robustez de utilizar mínimos cuadrados ponderados con respecto al método más simple, que asigna el mismo peso a todos los datos. Sin embargo, basando su planteamiento en la utilización de la función de Huber, indican que los cálculos no resultan excesivamente costosos para estimar el movimiento global entre imágenes consecutivas de un video, utilizando el método de Lucas y Kanade [31] que posteriormente complementan Shi y Tomasi [38].

¹¹Todo d_{ij} perteneciente a D es cero si $i \neq j$. Adicionalmente, el determinante de D no es nulo. En consecuencia, para cualquier d_{ii} en D se cumple $d_{ii} \neq 0$.

2.4. Estabilización del movimiento percibido entre imágenes

Para disminuir las vibraciones que resultan molestas en un video, el método más simple es promediar el valor de los parámetros que determinan el modelo utilizado. Ratakonda [36] implementa esta estrategia considerando el desplazamiento en 5 imágenes antes y después de la que se estabiliza.

Algunos autores proponen aplicar filtros típicos del procesamiento de señales digitales a los parámetros que componen la transformación que representa el movimiento global [20, 11]. De la aplicación de este filtro se obtiene la componente suave del movimiento estimado, tal que de su diferencia con el movimiento original puede calcularse la cantidad de movimiento a compensar.

Para llevar a cabo este proceso es necesario conocer los valores de los parámetros que conforman la transformación en el tiempo. Guestrin et ál. [20] indican que es común que la cámara se desplace sobre los ejes “x” e “y”, por lo cual primero refinan un modelo traslacional y, posteriormente, estiman un modelo “rígido”. Con este método secuencial reportan poder procesar hasta 5 imágenes por segundo, considerando el color en éstas para aumentar la robustez al estimar el movimiento.

Utilizando también un modelo “rígido”, Chen y Lovell [11] obtienen inicialmente un vector de traslación promedio a partir del campo de movimiento que se percibe entre imágenes consecutivas. Gracias a este resultado, luego encuentran el ángulo de rotación. Por medio de una implementación específica para cierto tipo de *hardware*, consiguen procesar hasta 17 imágenes por segundo.

Para estabilizar video también puede utilizarse un filtro de Kalman, según la propuesta de Litvin et ál. [29]. Éste consiste en estimar el estado de un sistema discreto y dinámico a partir de un conjunto de observaciones ruidosas. Las variables de estado del sistema de estabilización son los parámetros que conforman la transformación que representa la componente intencional del movimiento acumulado en el tiempo.

Para llevar a cabo la estimación, primero se calcula el movimiento global. Se define una transformación acumulativa, utilizando un modelo afín, que expresa el movimiento desde la primera imagen capturada, I^0 , hasta la actual, I^n . Esta transformación se obtiene aplicando, en cadena, cada una de las que se estiman entre imágenes consecutivas. Sea \mathbf{x}_n un punto en la imagen I^n y expresando el modelo afín por medio de una matriz A de 2×2 , que representa la transformación lineal conformada por a_1, a_2, a_3 y a_4 según (20), y un vector \mathbf{b} que posee los parámetros de traslación,

$$\mathbf{x}_n = A_n \mathbf{x}_{n-1} + \mathbf{b}_n = \left(\prod_{i=n}^1 A_i \right) \mathbf{x}_0 + \sum_{i=1}^n \left(\prod_{j=n}^{i+1} A_j \right) b_i = \hat{A}_n \mathbf{x}_0 + \hat{\mathbf{b}}_n \quad (25)$$

donde A_n y \mathbf{b}_n componen la transformación encontrada entre I^{n-1} e I^n y \widehat{A}_n y $\widehat{\mathbf{b}}_n$ conforman la acumulativa desde la primera imagen hasta la última.

Siendo \check{A}_n y $\check{\mathbf{b}}_n$ las matrices que componen la transformación que representa la componente intencional del movimiento acumulado, se atribuye el movimiento impulsivo a la diferencia entre $(\widehat{A}_n, \widehat{\mathbf{b}}_n)$ y $(\check{A}_n, \check{\mathbf{b}}_n)$. De esta manera, para encontrar $(\check{A}_n, \check{\mathbf{b}}_n)$ se utiliza el filtro antes mencionado, tal que se considera $(\widehat{A}_n, \widehat{\mathbf{b}}_n)$ como observaciones con ruido de la componente intencional del movimiento acumulada en el tiempo. Para estimar los valores de \check{a}_1 , \check{a}_4 y \check{b}_1 y \check{b}_2 se suponen velocidades constantes sobre los movimientos globales de zoom y traslación que se evidencian en la secuencia. Para los parámetros \check{a}_2 y \check{a}_3 se asume rotación constante, tal que sólo se modifican por ruido en las observaciones encontradas. Este marco de trabajo permite ajustar ciertos parámetros que determinan la cantidad de movimiento a compensar, siendo posible tanto estabilizar como eliminar totalmente el movimiento global percibido en la secuencia.

Una suposición importante se lleva a cabo para aplicar el filtro de Kalman. Los modelos de ruido que se emplean para los parámetros siguen una distribución gausiana, de media cero, y se consideran independientes, a pesar de no serlo en la realidad. Las variables \check{a}_2 y \check{a}_3 , por ejemplo, están relacionadas dado que ambas describen rotación. A pesar de este detalle, en [29] se reporta que el uso del filtro permite superar el desempeño de un producto comercial de estabilización.

Tico y Vehvilainen [43] complementan la metodología agregando restricciones sobre los parámetros, de modo que el video estabilizado puede ser enmarcado y no se observan regiones sin contenido en las imágenes.

Siguiendo los planteamientos de [29] para estimar la componente intencional del movimiento, pero utilizando un filtro de partículas para encontrar la transformación global, en [49] se lleva a cabo un proceso de estabilización *online* en función a un marco de referencia estático. Bajo un modelo “rígido”, el proceso de estimación del movimiento global consiste en aproximar recursivamente la distribución posterior del ángulo de rotación y los desplazamientos horizontal y vertical, por medio de un conjunto de partículas que representan estados hipotéticos. Estas partículas se generan a partir de una función denominada *importance density*. Particularmente, se proponen considerando su estado anterior y un factor de ruido que sigue una distribución gausiana. La media de la distribución se obtiene del estimado de movimiento global entre dos imágenes consecutivas, tal como se ha calculado en casos anteriores utilizando mínimos cuadrados. A cada uno de estos estados

hipotéticos se les asigna un peso, que depende de un proceso de comparación entre la imagen transformada con los valores que se proponen y la imagen inicial que sirve de referencia. El estimado de las variables de interés en un instante de tiempo, finalmente, se obtiene por medio de la suma ponderada de cada uno de los valores de estos parámetros según las propuestas de las partículas. A pesar de implicar un costo adicional, la utilización de este filtro busca obtener robustez ante estimados de baja calidad del movimiento global.

Con el objetivo de no considerar la cadena de movimiento global acumulado en el tiempo, puesto que en ella los errores influencian estimados posteriores, en [33] se considera un conjunto de imágenes vecinas, tal que los índices que las identifican en la secuencia se definen por $N_i = \{j \mid i-t \leq j \leq i+t\}$, siendo t un parámetro ajustable. De esta manera, considerando una imagen I^i y el estimado de movimiento global entre cada par de imágenes, se determinan las transformaciones que representan el movimiento desde I^i a cada I^j , tal que $j \in N_i$. En otras palabras, se calcula la posición de cada imagen en la vecindad con respecto a la que ocupa la posición central.

El conjunto de transformaciones relativas que se obtienen se consideran para llevar a cabo una convolución con una función gausiana, de media cero y desviación estándar \sqrt{t} . El valor que resulta de esta operación representa la compensación que debe aplicarse a la imagen I^i . Si bien en este trabajo se menciona que las transformaciones siguen un modelo afín, no se ofrecen detalles sobre cómo manejar los diferentes parámetros que las componen.

Johansen [26] propone ajustar una parábola a un conjunto de valores que representan el estado estimado antes y después de una imagen a estabilizar, con el fin de obtener la componente de movimiento intencional. Al igual que en los otros casos donde se consideran sólo un grupo pequeño de estimaciones, el video estabilizado muestra cierto retraso durante su reproducción; aunque es posible procesar la secuencia en tiempo real sin que los errores acumulados en el tiempo afecten gravemente el resultado.

Capítulo 3

Implementación

En este capítulo se describe la forma de trabajo empleada para el procesamiento en tiempo real de secuencias de imágenes capturadas desde un helicóptero teleoperado a control remoto. Así mismo, se ofrecen detalles sobre los aspectos más relevantes de la aplicación que se implementó para la estabilización de video.

El vehículo aéreo utilizado, ChocoLate, pertenece al Grupo de Inteligencia Artificial (GIA) de la Universidad. Para la obtención de imágenes en tiempo real, este helicóptero fue adaptado agregándole una pequeña cámara que transmite imágenes a color de manera inalámbrica. La salida de video analógica del receptor fue conectada como entrada a una *HandyCam*, modelo *DCR-TRV730* de *Sony*. Esta cámara ofrece la posibilidad de digitalizar video y, de esta manera, la salida digital de la misma a través del puerto *FireWire 400*¹ se conectó a una computadora donde se ejecuta la aplicación. De manera gráfica, el proceso llevado a cabo para la estabilización en tiempo real de imágenes se presenta en la Figura 2.²

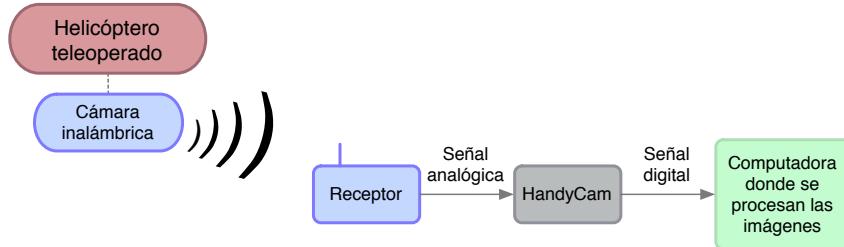


Figura 2: Esquema de trabajo para el procesamiento en tiempo real de imágenes capturadas desde un helicóptero teleoperado.

La cámara *Sony* también permite grabar, en cintas de formato *Digital8*, el contenido que recibe a través de su entrada analógica. De esta manera, se puede replicar el proceso de estabilización en tiempo real sin la necesidad de volar el helicóptero a control remoto. Los videos grabados en la cámara pueden ser transmitidos a la computadora donde corre la aplicación de estabilización a la misma velocidad³ que sucede cuando se transmite el video analógico desde el vehículo aéreo.

¹El puerto *FireWire* utilizado se conoce como la interfaz IEEE-1394 y permite la transmisión de información a una velocidad de hasta 400 Mbits/s.

²Detalles adicionales sobre ChocoLate y la metodología de trabajo empleada se presentan en el Apéndice B.

³La *HandyCam* transmite video a una velocidad de 30 imágenes por segundo a través del puerto *FireWire 400*.

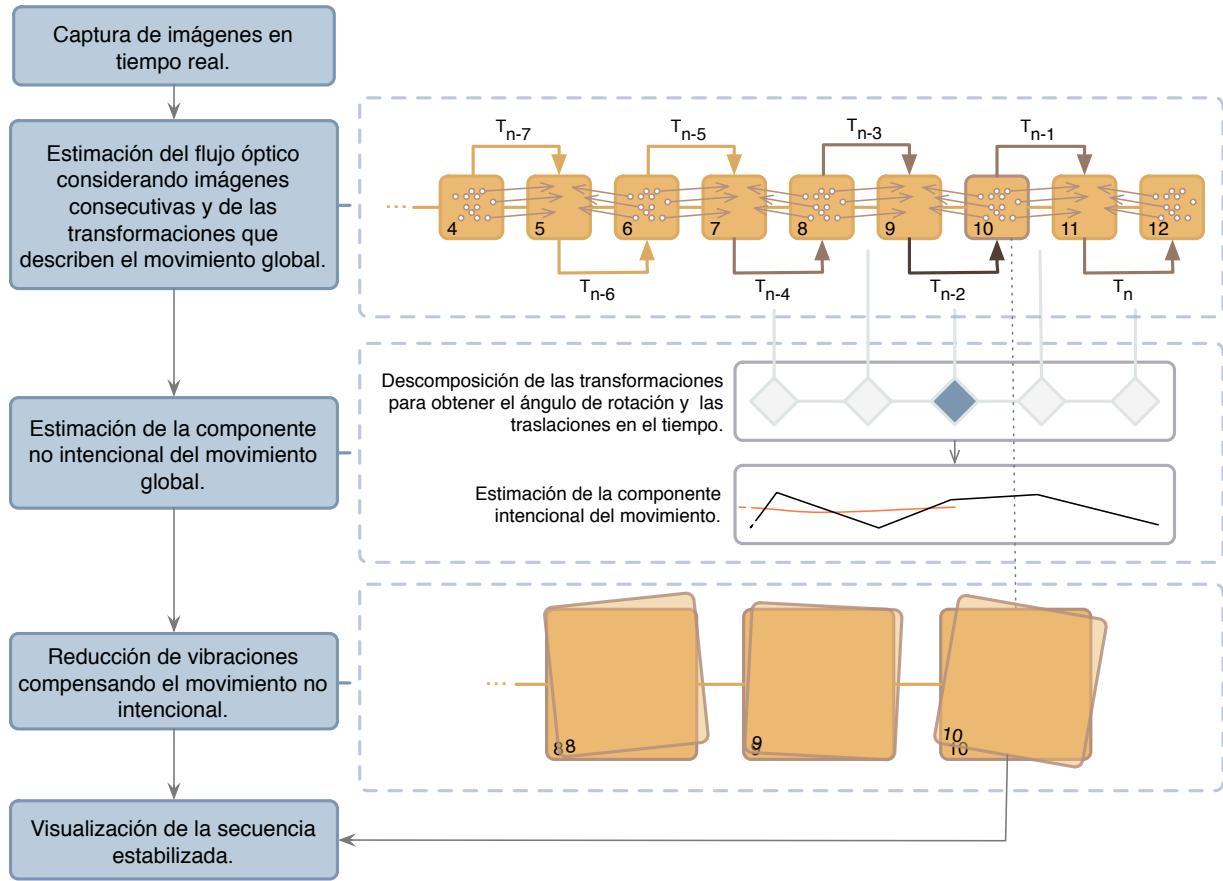


Figura 3: Esquema general del funcionamiento de la aplicación implementada para la estabilización de video. Se muestran 9 imágenes de una secuencia (cuadros de color amarillo), entre las cuales se estima una transformación T que modela el movimiento global. En algunos casos se lleva a cabo la estimación de flujo óptico considerando como referencia la imagen previa de un par, mientras que en otros se utiliza la que fue capturada más recientemente. Cada vez que se estima una transformación, se extraen de ella el ángulo de rotación y las traslaciones horizontales y verticales. Estos valores se acumulan en una ventana corrediza, de tamaño 5 en el ejemplo, de la cual puede estimarse la componente intencional del movimiento global. A partir de estos cálculos puede derivarse la componente no intencional del movimiento global (vibraciones) que debe ser compensada. Finalmente, se visualiza la secuencia en tiempo con mayor calidad al reducir los movimientos impulsivos.

El movimiento global que se percibe en una secuencia de imágenes posee dos componentes: movimiento intencional y no intencional. El objetivo de la aplicación implementada es, precisamente, reducir la componente no intencional con el fin de mejorar la calidad del video. Su diseño general se presenta en la Figura 3, donde cinco fases son fácilmente identificables. En la primera se captura el video. En la segunda se lleva a cabo el proceso de estimación del flujo óptico entre pares de imágenes consecutivas. Así mismo, se estima la transformación que modela el movimiento global percibido entre estas imágenes. En la tercera se extrae el ángulo de rotación y las traslaciones horizontales y verticales de la última transformación encontrada y, entonces, estos valores se acumulan en una ventana corrediza en el tiempo que representa el movimiento acumulado a lo largo de la secuencia. A partir del conjunto de valores de estos parámetros se encuentra la componente intencional del movimiento estimado. Posteriormente, en la cuarta fase se compensa el movimiento no intencional (vibraciones) que puede ser derivado a partir de los resultados anteriores. Finalmente, en la quinta se presentan las imágenes estabilizadas.

Se utilizó como único lenguaje de programación C++ para crear la aplicación bajo Mac OS X. La implementación de las funcionalidades principales dependen de la librería *Open Source Computer Vision Library* (OpenCV)[24], versión 1.0.0, que incluye numerosas implementaciones de algoritmos populares en el campo de visión por computador y soporta la captura de imágenes desde diferentes fuentes de video. Para llevar a cabo los procesos de estimación se utilizó la librería *GNU Scientific Library* (GLS)[18], versión 1.11. La interfaz fue creada utilizando *Qt*[46], versión 4.4, que facilita el manejo de hilos dentro de un mismo proceso.⁴

A continuación se detallan los aspectos más importantes de la implementación que se llevó a cabo, según las etapas antes mencionadas.

3.1. Captura de imágenes

La captura de la secuencia depende de la librería OpenCV. En Mac OS X, funciona gracias a la interfaz que provee QuickTime[3] para procesar imágenes recibidas por el puerto *FireWire*.

La señal digital de la *HandyCam* es procesada lo más rápido posible. Cada vez que se puede recibir una imagen, su información se incorpora al proceso de estabilización.

Las dimensiones de las imágenes a procesar no necesariamente son las originales. En particular,

⁴En el Apéndice C se describen las opciones que ofrece la aplicación, así como el diseño de su interfaz.

se decidió trabajar con imágenes de 320 píxeles de ancho, puesto que varios autores reportan resultados satisfactorios utilizando esta misma configuración[7, 11, 28, 9].

3.2. Estimación del flujo óptico y movimiento global

Para poder calcular una transformación que represente el movimiento global que se percibe entre imágenes consecutivas, producto del movimiento de la cámara, primero se estima el flujo óptico. Una vez obtenida una correlación entre los rasgos sobresalientes de un par de imágenes, la transformación encontrada a partir de éstos pudiese utilizarse para facilitar procesamientos posteriores del video, además de la estabilización.

3.2.1. Selección de “buenos” rasgos

La selección se lleva a cabo de manera alternada, tal como proponen Morimoto y Chellappa [35]. Dado un conjunto de “buenos” rasgos en un instante de tiempo t , éste se considera para encontrar una estimación del campo de movimiento entre las imágenes I^t e I^{t-1} y entre I^t e I^{t+1} . En la Figura 3, que describe de manera general la aplicación, se hace referencia a las últimas 9 imágenes que se pudieron capturar de una secuencia, enumeradas desde la 4 hasta la 12. La selección de rasgos ocurre en las pares y, según el caso, se encuentra la estimación de los vectores de flujo óptico en reversa o en la misma dirección en que se presenta la secuencia.

Esta estrategia busca acelerar el proceso de estabilización al reducir la cantidad de veces que deben seleccionarse “buenos” rasgos. Dos opciones alternativas a esta metodología parecerían ser útiles. Por un lado podrían seleccionarse buenos rasgos en un instante y luego buscar su correlación en varias imágenes sucesoras. Sin embargo, el algoritmo base de Lucas y Kanade [31] que estima la traslación de rasgos no es ideal al transcurrir el tiempo, tal como lo demuestran Shi y Tomasi [38]. Entonces, debería considerarse la versión alternativa del procedimiento que se fundamenta en un campo de movimiento afín, tal como se describe en la sección 2.2.3, a un costo adicional en el proceso de estimación de flujo óptico.

Otra opción pudiese ser considerar como conjunto de rasgos sobresalientes la correspondencia de aquellos que fueron seleccionados anteriormente. Sin embargo, existe una alta probabilidad que el ruido conlleve a procesar regiones en las imágenes en las cuales se dificulta estimar el flujo óptico.

Se utilizó la implementación en OpenCV de la estrategia de Bouget [8] para obtener el grupo de rasgos sobre los cuales calcular un aproximado de su movimiento. Tal como se mencionó en

la sección 2.2.4, los píxeles con calidad superior al umbral establecido son filtrados, reteniendo únicamente aquellos que poseen un mínimo autovalor que es máximo local en su vecindad de 3×3 píxeles. Considerando los mínimos autovalores asociados a cada píxel y su posición en la imagen sobre la que se quieren seleccionar “buenos” rasgos, el procedimiento para detectar estos máximos locales se fundamenta en la utilización del **operador dilatación**.

Considerando una matriz bidimensional M con números reales en sus casillas y una estructura S que determina el tamaño y la forma de la vecindad considerada; el operador dilatación retorna una matriz $D = dilatar(M, S)$, de igual dimensión que M , tal que

$$(\forall i, j : (i, j) \in M : D(i, j) = (\max x, y : (x, y) \in S(i, j) : M(x, y))) \quad (26)$$

Sea I la imagen sobre la cual se quieren seleccionar “buenos” rasgos y sea E una matriz con los mínimos autovalores de Z , según (17), asociados a cada píxel en I . Siendo λ_{\max} el máximo de los mínimos autovalores y k el porcentaje que define el nivel de calidad aceptado, consideramos E_t como la matriz de igual dimensión de E que cumple

$$\begin{aligned} (\forall x, y : (x, y) \in E : (E(x, y) > k\lambda_{\max}) \Rightarrow (E_t(x, y) = E(x, y)) \wedge \\ (E(x, y) \leq k\lambda_{\max}) \Rightarrow (E_t(x, y) = 0.0)) \end{aligned}$$

Entonces, el algoritmo de la librería OpenCV para la retención de los “buenos” rasgos que representan máximos locales, puede resumirse en dos pasos. Primero se calcula la matriz $D = dilatar(E_t, S)$ con S un rectángulo de tamaño 3×3 ; y, finalmente, son considerados máximos locales aquellos píxeles, con posición (x, y) , para los cuales se cumple que $D(x, y) = E_t(x, y)$ y $E_t(x, y) \neq 0.0$.

Si bien Bouget [8] pareciera que se define de manera estricta los máximos locales como aquellos píxeles para los cuales su mínimo autovalor asociado es mayor que el de sus vecinos contiguos, la implementación en la librería del detector de “buenos” rasgos relaja la definición y, de esta manera, considera máximos cuyo valor es mayor o igual al de sus vecinos.

El conjunto de rasgos seleccionados hasta el momento es ordenado de manera decreciente, según los mínimos autovalores asociados. Considerando primero aquellos rasgos de calidad superior, se van seleccionando píxeles que se distancian como mínimo un valor preestablecido. Así, finalmente,

concluye el proceso de selección de “buenos” rasgos para una imagen I .

3.2.2. Estimación del campo de movimiento

La estimación de movimiento se lleva a cabo entre dos imágenes. Dado un par, podemos ordenarlas según el momento en que son capturadas, tal que a la primera imagen la denominamos I^{t-1} y a la segunda I^t . Si bien los detalles que se presentan a continuación describen el procedimiento considerando como imagen de referencia I^t , los planteamientos son equivalentes para los casos en los cuales la estimación del campo de movimiento se lleva a cabo en reversa. La finalidad de esta fase es encontrar la correspondencia de algunos puntos que son considerados “buenos” rasgos, al menos en una de las imágenes sucesivas que se procesan. Siendo esta correspondencia la representación del campo de movimiento estimado, lo importante es relacionar de manera correcta los puntos con la imagen a la cual pertenecen. Cuando la correlación se lleva a cabo en reversa, los rasgos seleccionados pertenecen a la última imagen I^t ; mientras que cuando se lleva a cabo en dirección contraria, éstos provienen de la primera, I^{t-1} .

Se utilizó la implementación piramidal del algoritmo de Lucas y Kanade [31] en OpenCV [8]. El procedimiento para la estimación del flujo óptico de un punto en una imagen I^{t-1} con respecto a I^t , puede resumirse según el algoritmo 1. Como precondición estas las imágenes deben poseer las mismas dimensiones. El algoritmo supone que son en blanco y negro, tal que el valor de cada píxel representa la intensidad en ese punto. Adicionalmente, con el fin de proveer las condiciones mínimas para obtener una buena aproximación del vector de flujo óptico, el punto \vec{p} que se desea correlacionar debe ser un “buen” rasgo.

Al calcular la estimación de movimiento deben considerarse diferentes variables, entre las que se incluyen:

Tamaño de la vecindad W . La región sobre la cual resolver la ecuación estándar para el cálculo de flujo óptico de Lucas y Kanade [31], tal como se presenta en (5), se define por medio de un rectángulo de $(2w_x + 1) \times (2w_y + 1)$ píxeles. Inicialmente, esta región se centra sobre el punto \vec{p} que se desea correlacionar y, posteriormente, se desplaza a lo largo de la imagen que sirve de referencia. En [8] se reportan como valores usuales para w_x y w_y , con $w_x = w_y, 2, 3, 4, 5, 6$ y hasta 7 píxeles.

Algorithm 1: Estimación iterativa del vector de flujo óptico para un punto considerando dos imágenes consecutivas de una secuencia y utilizando pirámides de $m + 1$ niveles, con $0 \leq m$.

Input: Imágenes consecutivas I^{t-1} , I^t (referencia) de una secuencia y un punto \vec{p} en I^{t-1} a correlacionar. Tamaño de la región W , definida como una ventana de $(2w_x + 1) \times (2w_y + 1)$ píxeles, y de las pirámides a utilizar en función de su altura. Cota inferior β sobre la magnitud del movimiento estimado entre ciclos y número K de iteraciones máximas.

Output: Correspondencia de \vec{p} en I^t .

```

1 begin
2   Construir representaciones piramidales:  $\{I_L^{t-1}\}_{L=L_0, \dots, L_m}$  y  $\{I_L^t\}_{L=L_0, \dots, L_m}$ 
3    $\vec{g}_{L_m} = [g_x \ g_y]_{L_m}^T \leftarrow [0 \ 0]^T$ 
4   for  $L \leftarrow L_m$  to  $L_0$  do
    // Se ubica el punto  $\vec{p}$  en  $I_L^{t-1}$ 
5    $\vec{p}_L = [p_x \ p_y]_L^T \leftarrow \vec{p}/2^L$ 
6   Se estima la derivada de  $I_L^{t-1}$  respecto a  $x$  en la vecindad de  $\vec{p}$ :  $I_x(x, y)$ 
7   Se estima la derivada de  $I_L^{t-1}$  respecto a  $y$  en la vecindad de  $\vec{p}$ :  $I_y(x, y)$ 
8   Se calcula la matriz  $Z$ , definida en (17), con información del gradiente:
    
$$Z \leftarrow \sum_{x=p_{x_L}-w_x}^{p_{x_L}+w_x} \sum_{y=p_{y_L}-w_y}^{p_{y_L}+w_y} \begin{bmatrix} I_x^2(x, y) & I_x(x, y)I_y(x, y) \\ I_x(x, y)I_y(x, y) & I_y^2(x, y) \end{bmatrix}$$

    // Proceso de estimación iterativo
9    $\vec{v}_0 = [v_x \ v_y]_0^T \leftarrow [0 \ 0]^T$ 
10  for  $k \leftarrow 1$  to  $K$  do
    Se calcula  $\vec{e}$ , según (5), en la región de interés:
    
$$\vec{e} \leftarrow \sum_{x=p_{L_x}-w_x}^{p_{L_x}+w_x} \sum_{y=p_{L_y}-w_y}^{p_{L_y}+w_y} \begin{bmatrix} d(x, y)I_x(x, y) \\ d(x, y)I_y(x, y) \end{bmatrix}$$

    siendo  $d(x, y) = I_L^{t-1}(x, y) - I_L^t(x + g_{x_L} + v_{x_{k-1}}, y + g_{y_L} + v_{y_{k-1}})$ 
12  Se estima el flujo óptico:  $\vec{n}_k \leftarrow Z^{-1}\vec{e}$ 
13  Se actualiza la predicción para el siguiente ciclo:  $\vec{v}_k = \vec{v}_{k-1} + \vec{n}_k$ 
14  if  $|\vec{n}_k| < \beta$  then
    break
16  Estimación de movimiento final en nivel  $L$ :  $\vec{h}_L \leftarrow \vec{v}_k$ 
17  Predicción de movimiento para el siguiente nivel:  $\vec{g}_{L-1} = 2(\vec{g}_L + \vec{h}_L)$ 
18  Vector de flujo óptico final:  $\vec{h} \leftarrow \vec{g}_0 + \vec{h}_0$ 
19  return  $\vec{p} + \vec{h}$ 
20 end

```

Tamaño de las pirámides a utilizar. El tamaño de las pirámides⁵ debe ser escogido en función de las dimensiones originales de las imágenes y el campo de movimiento esperado entre éstas. Bouget [8] hace referencia, por ejemplo, a pirámides de hasta 5 niveles de profundidad para imágenes de 640×480 píxeles.

Número de iteraciones máximas. Para el refinamiento iterativo, tal como se describió en la sección 2.2.2, se establece un máximo número de ciclos. Bouget [8] indica que un valor para esta cota puede ser 20, sin embargo, reporta que usualmente 5 iteraciones son suficientes para converger.

Cota mínima sobre la variación del movimiento estimado entre iteraciones. En el proceso de refinamiento también se establece como condición para seguir iterando un mínimo valor β sobre la variación del estimado obtenido entre ciclos. A esto se debe la condición en la línea 14 del algoritmo.

Siguiendo la notación del algoritmo 1, designaremos los niveles de una pirámide con la letra L . De esta manera, si consideramos una imagen I de dimensiones $n_x \times n_y$, entonces I_{L_0} es la imagen en el nivel cero de la pirámide, donde se posee la mayor resolución y la escala es la original ($n_x^{L_0} = n_x$ y $n_y^{L_0} = n_y$). La estructura piramidal se construye de manera recursiva, tal como se muestra en la Figura 19 del Apéndice F, reduciendo a la mitad el tamaño de las imágenes de un nivel al siguiente.⁶

La implementación del algoritmo no impone como restricción que para calcular el vector de flujo óptico la región W de $(2w + 1) \times (2w + 1)$ píxeles debe estar contenida por completo en las imágenes. Si éste fuera el caso, entonces se crearía una banda “prohibida” cerca del borde. Siendo L_m el máximo nivel de las pirámides utilizadas, esta banda tendría un ancho de $2^m w_x$ píxeles⁷. Por ejemplo, para $w_x = w_y = 5$ utilizando una pirámide de hasta 4 niveles de profundidad (L_3), se tiene una banda de 40 píxeles de ancho.

Entonces, el algoritmo puede procesar puntos cerca del borde de las imágenes permitiendo que parte de la ventana de integración sobresalga las dimensiones de éstas. Debe considerarse que las

⁵El tamaño de la pirámide se refiere a su altura o la cantidad de niveles que conforman la estructura. No parece lógico utilizar una pirámide de máximo un nivel de profundidad, pues este caso resulta equivalente a procesar las imágenes en sus dimensiones originales.

⁶Detalles adicionales sobre la construcción de las pirámides se presentan en el Apéndice F.

⁷Puesto que la región W se considera como un rectángulo cuadrado, $w_x = w_y$, entonces el ancho de la banda también puede ser expresado como $2^m w_y$ píxeles.

sumatorias expresadas en el pseudocódigo deben llevarse a cabo únicamente sobre la porción válida de la vecindad. Es decir, sobre los puntos (x, y) para los cuales se puede calcular $I_x(x, y)$, $I_y(x, y)$ y $d(x, y)$.

Si el punto \vec{p} sobresale las dimensiones de I^{t-1} o si durante los cálculos resulta que su correspondencia $(p_x + g_{xL} + v_{x_{k-1}}, p_y + g_{yL} + v_{y_{k-1}})$ sobresale las de I^t , no se puede estimar el vector de flujo óptico. En la implementación del procedimiento de Lucas y Kanade [31] en OpenCV, dada una lista de puntos para los cuales estimar su movimiento entre dos imágenes, se retorna una lista que indica, para cada uno, si pudo encontrarse una correspondencia o no. Los puntos para los cuales no se pudo completar el proceso expuesto por el algoritmo 1, son calificados como **rasgos “perdidos”**.

Más información sobre la implementación de citetbouguet99:techreport y, en particular, sobre el manejo de subpixeles se detalla en el Apéndice G. Como otro detalle, para obtener un valor aproximado del gradiente se utiliza el operador ***Scharr***.⁸

3.2.3. Estimación de la transformación que modela el movimiento global

A partir de los vectores de flujo óptico encontrados entre dos pares de imágenes consecutivas, se puede derivar una transformación que refleja el movimiento de la cámara. En la Figura 3, estas transformaciones se representan por la letra T a lo largo de la secuencia.

Con el objetivo de estimar el movimiento global fueron considerados varios modelos. Como métodos de estimación se optó por seleccionar mínimos cuadrados (M.C.), por su gran popularidad tal como se evidencia en la sección 2.3.2, y mínimos cuadrados totales (M.C.T.)⁹, como una alternativa adicional. A continuación se describe la combinación de estos factores que se implementó para la aplicación de estabilización.

Modelo afín utilizando mínimos cuadrados. La descripción de este tipo de transformación se presenta en la sección 2.3.1. Considerando la ecuación (20) y un conjunto de n correspondencias $(x_i, y_i) \rightarrow (\dot{x}_i, \dot{y}_i)$, $1 \leq i \leq n$, se estima el modelo por medio de mínimos cuadrados a

⁸En el Apéndice D se explica de manera general cómo puede estimarse el gradiente de una imagen y se ofrecen detalles sobre el operador ***Scharr***.

⁹La descripción de una regresión lineal múltiple por mínimos cuadrados totales se presenta en el Apéndice H.

través del sistema sobredeterminado $A\vec{x} = \vec{b}$, tal que

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a_1 & a_2 & a_5 & a_3 & a_4 & a_6 \end{bmatrix}^T = \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \vdots \\ \dot{x}_n \\ \dot{y}'_n \end{bmatrix}$$

Modelo afín utilizando mínimos cuadrados totales. Al igual que en el caso anterior, se considera la representación matricial de esta transformación según la ecuación (20). Para cada una de las correspondencias $(x_i, y_i) \rightarrow (\dot{x}_i, \dot{y}_i)$, $1 \leq i \leq n$, encontradas entre un par de imágenes, este modelo busca ajustar 6 parámetros tal que $a_1x + a_2y + a_5 \approx \dot{x}$ y $a_3x + a_4y + a_6 \approx \dot{y}$.

Por medio de una regresión lineal múltiple de mínimos cuadrados totales aplicada a cada planteamiento, se desea hallar el plano $AX + BY + CZ + D = 0$ que se ajuste lo mejor posible a los datos en 3D¹⁰, con $C = 1$ y $A = a_1$ o $A = a_3$, $B = a_2$ o $B = a_4$ y $D = a_5$ o $D = a_6$, respectivamente. Más detalles sobre este procedimiento se presentan en el Apéndice H.

Modelo similar utilizando mínimos cuadrados. La ecuación (21) describe la transformación geométrica de este modelo por medio de 4 parámetros. Dado un conjunto de n correspondencias $(x, y) \rightarrow (\dot{x}, \dot{y})$, $1 \leq i \leq n$, el método de mínimos cuadrados puede ser aplicado para resolver el sistema $A\vec{x} = \vec{b}$, expresado como

$$\begin{bmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & -y_n & 1 & 0 \\ y_n & x_n & 0 & 1 \end{bmatrix} \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix}^T = \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \vdots \\ \dot{x}_n \\ \dot{y}'_n \end{bmatrix}$$

Modelo bilineal utilizando mínimos cuadrados. Según la ecuación (22), este modelo se compone de 8 parámetros. Tal como en los casos anteriores donde se utiliza mínimos cuadrados,

¹⁰La solución de una regresión por mínimos cuadrados totales considera como medida de error la distancia ortogonal de los puntos dados al hiperplano que mejor se ajusta a los datos.

considerando n correspondencias, se busca resolver

$$\begin{bmatrix} x_1 & y_1 & 1 & x_1y_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & 1 & x_1y_1 \\ \vdots & \vdots \\ x_n & y_n & 1 & x_ny_n & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_n & y_n & 1 & x_ny_n \end{bmatrix}^T = \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \vdots \\ \dot{x}_n \\ \dot{y}'_n \end{bmatrix}$$

Todos los métodos de estimación utilizados son iterativos. En cada ciclo se determinan los vectores de movimiento que resultan atípicos ante la transformación encontrada. En el proceso de ajuste del modelo a los datos, se establece un máximo número de iteraciones. En cada nuevo paso de refinamiento se consideran únicamente los datos que no fueron clasificados como atípicos anteriormente. Si su número es menor a la mínima cantidad de datos que requiere el modelo de transformación utilizado, también se detiene el proceso.

La clasificación de los vectores depende de un umbral ε establecido sobre la distancia entre la correspondencia de su rasgo inicial (x_i, y_i) y su posición al aplicar la transformación que modela el movimiento global. Si esta distancia es mayor a ε , se considera que el vector es producto de un cuerpo u objeto que se mueve independiente en la escena o de una mala estimación del flujo óptico y que, en cualquier caso, no sigue el movimiento global percibido entre el par de imágenes procesadas.

3.3. Estimación del movimiento no intencional

Dada una transformación T_i entre dos imágenes consecutivas de una secuencia, se define su vecindad de la siguiente manera:

$$vecindad(T_i) = \{T_j \mid i - t \leq j \leq i + t\} \quad (27)$$

tal que $t \geq 0$ es un entero que determina la cantidad de transformaciones vecinas consideradas a la derecha y a la izquierda de T_i . Tal como se muestra en la Figura 3, por ejemplo, la $vecindad(T_{n-2})$ se compone por las transformaciones que van desde T_{n-4} hasta T_n , inclusive.

Dado un cierto t y siendo T_n la última transformación encontrada que modela el movimiento global entre dos imágenes, nos interesa enfocarnos en la vecindad de la transformación T_{n-t} para

estimar, de ella y sus vecinos, el movimiento no intencional en tiempo real. Esta vecindad se compone de las transformaciones que van desde T_{n-2t} hasta T_n .

A partir de este conjunto, se define una ventana corrediza de tamaño $2t + 1$, sobre la cual se registran el ángulo de rotación y las traslaciones horizontales y verticales acumuladas que pueden derivarse de las transformaciones que lo componen. De los valores que se tienen de estos parámetros en la ventana, se estima la componente intencional del movimiento, bajo el supuesto que ésta es suave. La diferencia entre esta componente en el instante $n - t$ y el movimiento global acumulado hasta la transformación T_{n-t} , finalmente, representa el movimiento no intencional que se desea suprimir en la imagen I^{n-t} , siendo I^n la última capturada. Naturalmente, este proceso conlleva a un retraso en la visualización de las imágenes que se estabilizan; sin embargo, este efecto puede reducirse en función del tamaño de la vecindad considerada. Mientras más grande ésta es, más suave es el estimado de la componente intencional del movimiento, a pesar que la imagen que se visualiza no es tan reciente.

3.3.1. Simplificación de la transformación que modela el movimiento global

Para estabilizar la secuencia se considera una transformación simple que incluye únicamente rotación y traslación, a partir de los parámetros que pueden extraerse del modelo que representa el movimiento global estimado, tal como se muestra en la Figura 4.

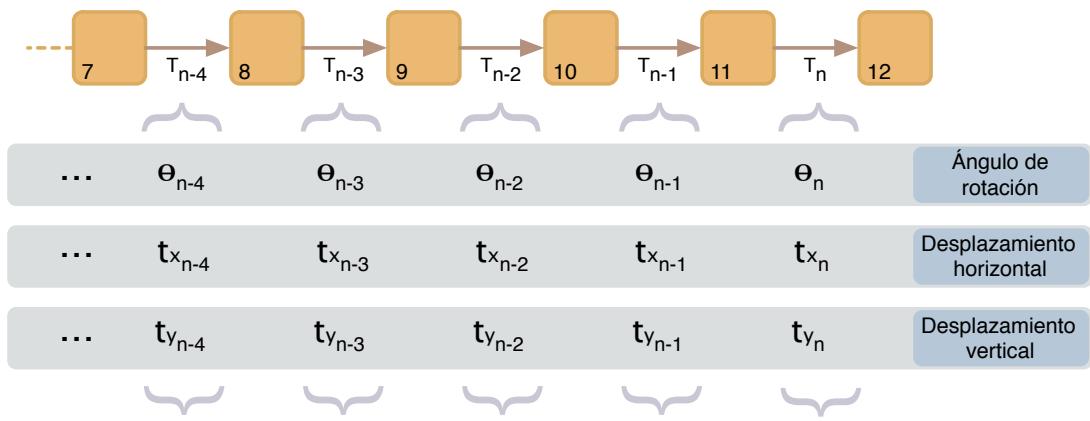


Figura 4: Proceso de simplificación de las transformaciones estimadas, siguiendo el ejemplo del esquema general presentado en la Figura 3. En la secuencia de imágenes (cuadros amarillos), T_n representa la última transformación encontrada.

Cada vez que se tiene un nuevo par de imágenes sobre las cuales estimar el flujo óptico, también

se calcula la transformación que modela el movimiento global percibido. Cualquiera sea el modelo utilizado, de la estimación de la transformación fácilmente se pueden extraer el desplazamiento horizontal y el vertical. Si se trata de una transformación afín, según la ecuación (20), estos desplazamientos están dados por los parámetros a_5 y a_6 . Si se utiliza un modelo similar, tal como en (21), entonces éstos se representan por t_x y t_y . Si es un modelo bilineal, siguiendo la definición de (22), se pueden extraer de b_5 y b_6 .

Para todos los modelos considerados al estimar el movimiento global, parte de su formulación se describe por medio de una matriz cuadrada de 2×2 que representa una transformación lineal. Si la transformación que modela el movimiento global no incluye reflexión, entonces de una factorización polar de esta matriz se puede obtener el ángulo de rotación que mejor la explica. Utilizando una descomposición en valores singulares para encontrar esta factorización [39] y siendo M la transformación lineal, se tiene

$$M = USV^T = (UV^T)(VSV^T) \quad (28)$$

donde U y V son matrices ortogonales y S es diagonal de números no negativos. El factor UV^T puede representar rotación o reflexión, según el signo de su determinante. Shoemake y Duff [40] definen VSV^T como una operación de *stretch*, que en algún sistema de coordenadas representa un escalamiento. Más detalles sobre esta descomposición se ofrecen en el Apéndice I.

Cuando el determinante de UV^T es positivo, se tiene una rotación

$$UV^T = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \quad (29)$$

y se puede calcular de manera sencilla el ángulo θ .

Cuando se trata de una reflexión, la transformación encontrada se considera una “mala” aproximación del movimiento real de la cámara.

En el caso de poder calcular exitosamente las componentes de traslación y rotación de la transformación que modela el movimiento global, estos parámetros sirven para estimar el movimiento acumulado de manera recursiva en el tiempo. Supongamos que de la primera transformación se obtiene un ángulo de rotación θ_0 . Entonces, al encontrar la siguiente descomposición, el ángulo de

rotación acumulado puede calcularse como $\theta_0 + \theta_1$. Un planteamiento equivalente puede aplicarse para los desplazamientos, tal que finalmente se pueden construir tres cadenas que representan el movimiento con respecto a la primera imagen de la secuencia, considerando respectivamente traslación horizontal, vertical y rotación. De manera general, siendo ρ alguno de estos parámetros y habiendo extraído su valor de las transformaciones que van desde T_1 hasta T_n , su cadena de valor acumulado puede representarse de la siguiente manera:

$$C_\rho = < \sum_{i=1}^1 \rho_i, \sum_{i=1}^2 \rho_i, \dots, \sum_{i=1}^n \rho_i > \quad (30)$$

Como una decisión particular de la implementación, cuando se determina que la transformación encontrada poco explica el movimiento de la cámara, entonces se supone continuidad en el movimiento global. Se utiliza la descomposición de la transformación anterior a ésta para continuar la construcción de las cadenas que representan el movimiento acumulado. En otras palabras, si se determina que la transformación T_n es una “mala” estimación, en vez de utilizar su descomposición para seguir construyendo las cadenas, se utiliza de nuevo la de T_{n-1} . Cuando pareciera que la primera transformación estimada, T_1 , es errónea, entonces se utiliza la descomposición de la transformación identidad.

3.3.2. Estimación de la componente intencional del movimiento global

A partir de la vecindad de tamaño $2t + 1$ de la transformación T_{n-t} , se definió anteriormente una ventana corrediza en el tiempo. Esta ventana sirve para identificar los valores en las cadenas de movimiento acumulado, construidas previamente, que se utilizan para estimar la componente intencional del movimiento que se derivada de T_{n-t} y sus vecinos.

Tomando en cuenta los $2t + 1$ últimos valores que componen la cadena del ángulo de rotación acumulado, así como aquellos de las cadenas de desplazamiento, y una función gausiana, se llevan a cabo tres operaciones de convolución. A partir de sus resultados, se genera una transformación que busca aproximar la componente intencional del movimiento global percibido.

Para ilustrar este planteamiento, sea C_θ la cadena de n valores que representan del ángulo de rotación acumulado en el tiempo, tal como se presenta en la Figura 5, y sea G una función gausiana con media 0. Entonces, el ángulo de rotación $\check{\theta}$ de la componente intencional del movimiento

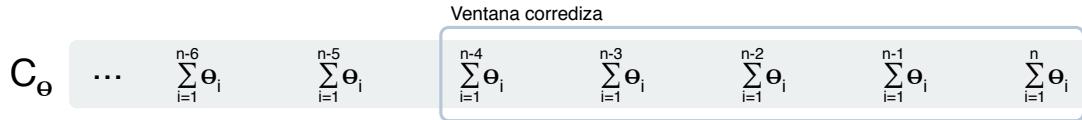


Figura 5: Ejemplo de la ventana corrediza, de tamaño $2t+1$, que identifica los valores a considerar en la cadena C_θ , compuesta por ángulo de rotación acumulado en cada instante. Siguiendo el ejemplo que se presenta en la Figura 3, se considera $t = 2$, siendo T_{12} la última transformación estimada que representa el movimiento global.

percibido en el instante $n - t$ puede estimarse como

$$\check{\theta}_{n-t} = \frac{\sum_{i=n-2t}^n C_\theta(i) G((n-t)-i)}{\sum_{j=-t}^t G(j)} \quad (31)$$

donde el denominador simplemente normaliza los valores de la gausiana que son considerados.

Obteniendo de manera similar los desplazamientos $\check{t}_{x_{n-t}}$ y $\check{t}_{y_{n-t}}$, la transformación 2D que modela la componente intencional del movimiento puede expresarse de la siguiente manera

$$\check{T}_{n-t}(\vec{x}) = \begin{bmatrix} \cos(\check{\theta}_{n-t}) & -\sin(\check{\theta}_{n-t}) \\ \sin(\check{\theta}_{n-t}) & \cos(\check{\theta}_{n-t}) \end{bmatrix} \vec{x} + \begin{bmatrix} \check{t}_{x_{n-t}} \\ \check{t}_{y_{n-t}} \end{bmatrix} \quad (32)$$

Según la desviación estándar de las gausianas utilizadas, el movimiento intencional estimado tendrá cierto grado de suavidad. Mientras más grande es la desviación, más peso se le atribuye a los parámetros de las transformaciones vecinas de T_{n-t} y, por lo tanto, más suave será el estimado.

3.3.3. Detección de movimientos impulsivos

Siendo T_n la última transformación encontrada, \check{T}_{n-t} es la última estimación de la componente intencional del movimiento que puede calcularse por medio del procedimiento antes explicado. Considerando los ángulos de rotación θ_{n-t} y $\check{\theta}_{n-t}$, los desplazamientos horizontales $t_{x_{n-t}}$ y $\check{t}_{x_{n-t}}$ y los desplazamientos verticales $t_{y_{n-t}}$ y $\check{t}_{y_{n-t}}$, se puede derivar de su diferencia el movimiento impulsivo que se desea compensar.

Por ejemplo, si $\theta_{n-t} = \frac{pi}{100}$ y $\check{\theta}_{n-t} = \frac{pi}{200}$, entonces debe compensarse una rotación extra de $\frac{pi}{100} - \frac{pi}{200} = \frac{pi}{200}$ radianes o 1.8 grados. De esta manera, el ángulo de rotación acumulado $\tilde{\theta}_{n-t}$, que representa la componente no intencional del movimiento en el instante $n - t$, puede encontrarse como $\tilde{\theta}_{n-t} = \theta_{n-t} - \check{\theta}_{n-t}$. Utilizando este mismo método para los desplazamientos acumulados a lo

largo de los ejes de coordenadas cartesianas, puede obtenerse el valor de $\tilde{t}_{x_{n-t}}$ y $\tilde{t}_{y_{n-t}}$.

3.4. Reducción de vibraciones

Para finalizar el proceso de estabilización, sólo falta compensar la componente de movimiento no intencional antes encontrada. Para ello se considera la inversa de la transformación que la representa:

$$\tilde{T}_{n-t}^{-1}(\vec{x}) = \begin{bmatrix} \cos(-\tilde{\theta}_{n-t}) & -\sin(-\tilde{\theta}_{n-t}) \\ \sin(-\tilde{\theta}_{n-t}) & \cos(-\tilde{\theta}_{n-t}) \end{bmatrix} \vec{x} + \begin{bmatrix} -\tilde{t}_{x_{n-t}} \\ -\tilde{t}_{y_{n-t}} \end{bmatrix} \quad (33)$$

Finalmente, el valor de un píxel \vec{x} en la imagen I^{n-t+1} compensada puede calcularse como $I^{n-t+1}(\tilde{T}_{n-t}(\vec{x}))$. Si bien se quiere compensar el movimiento no intencional, se aplica la transformación \tilde{T}_{n-t} directamente al generar la imagen estabilizada. Esto se debe a que para aplicar una transformación cualquiera T a una imagen, se utiliza su inversa¹¹. Entonces, para aplicar la transformación $\tilde{T}_{n-t}^{-1}(\vec{x})$, se utiliza $(\tilde{T}_{n-t}^{-1})^{-1} = \tilde{T}_{n-t}$.

3.5. Visualización de la secuencia estabilizada

Al principio de la secuencia, cuando se tienen menos de $2t + 1$ transformaciones, no se puede llevar a cabo el proceso de estabilización. Apenas se cuenta con suficiente información, se estabiliza la imagen I^{n-t+1} y éste es el resultado que se muestra en la aplicación. Al final de la secuencia, cuando ya no se tienen más transformaciones que procesar y, por lo tanto, la ventana corrediza no puede ser llenada completamente, se muestra en pantalla la imagen original recibida desde la *HandyCam*.

Para mejorar la calidad del video y como una opción adicional para el usuario, se puede enmarcar la imagen estabilizada. De esta manera se pueden ocultar las regiones sin información que resultan al compensar la señal original.

3.6. Consideraciones especiales ante la presencia de ruido

Las imágenes transmitidas analógicamente desde ChocoLate usualmente poseen un nivel elevado de ruido. Este factor afecta gravemente el proceso de estimación de flujo óptico y, en consecuencia, la calidad de la estabilización de las secuencias en tiempo real. Afortunadamente este ruido no es constante y, por ello, a pesar de su presencia es posible estabilizar la mayoría del tiempo.

¹¹Más detalles sobre el procedimiento para compensar una imagen se explican en el Apéndice J.

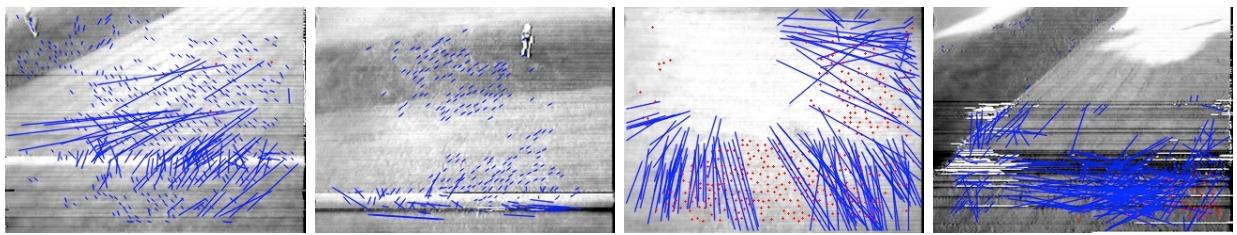


Figura 6: Ejemplos de ruido presente en imágenes transmitidas analógicamente desde ChocoLate. Las líneas azules representan vectores de flujo óptico y los puntos rojos son “buenos” rasgos que durante el proceso de estimación de movimiento fueron considerados “perdidos”.

En la Figura 6 se muestran distorsiones comunes en los videos. En estas imágenes, la escena se compone por el mismo campo abierto donde, a veces, se aprecian personas caminando en diferentes direcciones. Tal como se mencionó anteriormente, cuando se estima que una transformación poco explica el movimiento global, se utiliza la descomposición de la transformación anterior a ella para predecir el movimiento. Esto es común ante la presencia de ruido cuando se estiman transformaciones que incluyen reflexión.

Menos drástica es la situación ante operaciones de inclinación. Sin embargo, si el ángulo de *shearing* es de magnitud significativa, entonces el proceso de estabilización también se ve afectado de manera negativa. Lamentablemente, tal como se explica en el Apéndice I, la descomposición polar de una transformación no cuenta con una forma explícita para representar este tipo de operaciones. Sin embargo, como otra decisión en el proceso de implementación, si la descomposición de esta transformación refleja una variación importante entre el escalamiento horizontal y el vertical que de ella se derivan, entonces también se considera que poco explica el movimiento real de la cámara. Así mismo, si el porcentaje de escalamiento en una dirección supera o se aleja más de un límite preestablecido con respecto a 1, que equivale a que no se lleva a cabo una operación de zoom, entonces también se utiliza la información de la transformación anterior.

La estrategia implementada resulta sumamente conveniente por su rapidez y la posibilidad de ajustarse a las necesidades del usuario. En vista de ello, también se decidió extenderla para limitar los desplazamientos y la rotación. No sólo es útil ante la presencia de ruido, sino también ante una mala estimación del flujo óptico producto, por ejemplo, de un movimiento acelerado de la cámara.

Capítulo 4

Experimentos y Resultados

En este capítulo se presentan los experimentos realizados para evaluar la estabilización de secuencias de imágenes bajo diferentes condiciones.¹

Los experimentos pueden ser agrupados en dos grandes fases. Los primeros buscan determinar empíricamente propiedades sobre las variables que influyen en la estimación del flujo óptico. En función de los resultados obtenidos, el siguiente grupo busca evaluar el desempeño de la transformación utilizada en el proceso de estimación del movimiento global entre imágenes consecutivas en una secuencia. Así como también evalúa el proceso de estabilización.

Si bien uno de los requerimientos de la aplicación de estabilización implementada es poder operar en tiempo real, la mayoría de los experimentos fueron llevados a cabo sobre un repositorio de videos extraídos de *YouTube.com* y otros capturados desde ChocoLate², de 320×240 píxeles. La utilización de estas secuencias de imágenes facilita llevar a cabo las pruebas y permite registrar numerosos resultados. Entre sus características vale destacar que presentan contenidos diversos, entre los que se encuentran campos amplios, calles, estacionamientos y parques, y fueron capturadas bajo diferentes condiciones climáticas, tales como ambientes cálidos en costas o fríos ante la presencia de nieve. Una característica importante sobre los videos capturados desde ChocoLate es la transmisión analógica de los mismos que, usualmente, conlleva a un nivel elevado de ruido en sus imágenes. Algunos de los videos de *YouTube.com* también presentan ruido; sin embargo, en su mayoría poseen un nivel de calidad superior al de las imágenes capturadas desde el helicóptero del GIA.

4.1. Evaluación del proceso de estimación del flujo óptico

Con el fin de evaluar el desempeño del algoritmo de estimación del campo de movimiento bajo diferentes condiciones, se consideraron las etapas que conforman el proceso y se llevaron a cabo 3 experimentos.

Los programas utilizados en esta fase llevan a cabo únicamente el proceso de estimación de

¹Todos los experimentos se llevaron a cabo bajo Mac OS X, en una computadora de 2.16GHz Intel Core 2 Duo con 2GB de memoria.

²La metodología de trabajo empleada al momento de registrar los videos capturados desde ChocoLate se presenta en el Apéndice B.

flujo óptico, según la implementación iterativa en OpenCV [8] del algoritmo de Lucas y Kanade [31]. Los procedimientos que utilizan son un subconjunto de aquellos que permiten llevar a cabo la estabilización de imágenes en la aplicación implementada. De esta manera, no incluyen una interfaz que permita visualizar los resultados; sin embargo, registran cuanta información sea posible en cada caso.

Considerando el procedimiento utilizado para la detección de “buenos” rasgos, entre las variables influyentes en este proceso para la estimación del flujo óptico se encuentran: el tamaño de los videos para su procesamiento³, el nivel de profundidad máximo de la pirámide a utilizar, el número máximo de rasgos a procesar, la distancia mínima entre éstos, el nivel de calidad mínimo para considerarlos “buenos” y las dimensiones de la ventana W , definida según el sistema 4. Así mismo, en el proceso iterativo vale considerar también el criterio de parada establecido, tal como se expuso en la sección 3.2.2.

En la literatura no se encuentran muchos detalles sobre las implementaciones del método de detección de rasgos para el algoritmo de Lucas y Kanade [31]. Una de las pocas excepciones es el trabajo de Barron et ál. [5], quienes utilizan 1.0 y 5.0 como valores mínimos de calidad aceptada para los rasgos. Naturalmente, mientras más alto es el nivel de calidad mínimo, menos densa resulta la estimación del campo de movimiento. En todas las secuencias de imágenes sintéticas estudiadas por ellos que buscan representar movimientos reales, el algoritmo con nivel de calidad de 5.0 ofrece un menor error promedio, con una menor desviación estándar, que el de menor umbral. Vale destacar que la selección de este umbral depende de las propiedades de las imágenes procesadas, tal como se expone en [37]. En el caso de utilizar imágenes reales, ello implica que este valor depende de la calibración de la cámara usada.

Buscando llevar a cabo el proceso de estabilización rápidamente, se decidió utilizar imágenes de 320 píxeles de ancho en todos los experimentos de este grupo, determinando su altura en función de la relación de aspecto del video original⁴. En este sentido, varios autores de trabajos mencionados previamente reportan resultados utilizando este mismo tamaño para el procesamiento de imágenes [7, 11, 28, 9].

Puesto que es de interés obtener un algoritmo capaz de procesar video en tiempo real, se decidió considerar 500 como el número máximo de rasgos a procesar por imagen. Dado que los

³El tamaño de los videos se refiere al tamaño de las imágenes que conforman la secuencia (*ancho* \times *alto* en píxeles).

⁴En particular, los videos del repositorio utilizado en los experimentos poseen una relación de aspecto de 4:3.

procesos de estimación ganan robustez ante la presencia de más muestras que resten importancia a datos atípicos, intuitivamente pareciera que mientras más rasgos “buenos” se procesan, mejor será el estimado de la transformación global ante la presencia de *outliers*. Por lo tanto, para este experimento se decidió utilizar como distancia mínima entre rasgos 5 píxeles, tal que se busca ofrecer un bajo grado de dispersión a lo largo de las imágenes, mientras que no se disminuye significativamente el número de rasgos. Bajo las mismas consideraciones, se decidió utilizar como umbral de calidad un mínimo del 5 % del máximo de los mínimos autovalores para cada imagen procesada. En otras palabras, dados todos los mínimos autovalores de una imagen, se busca el mayor y el umbral de calidad es este valor multiplicado por 0.05, tal como se explica en la sección 3.2.1 siguiendo el planteamiento de [8].

En general, en estos experimentos resultan interesantes los siguientes aspectos:

- (a) Error de la estimación de movimiento, según el planteamiento de la ecuación (1), para cada uno de los rasgos procesados que no se “ pierden” durante el proceso iterativo⁵. En los resultados se presenta el promedio de este error al agrupar los pares de imágenes según el caso, así como también se reporta el error estándar de las medias obtenidas⁶.
- (b) Proporción de rasgos “ perdidos”. Estos rasgos, tal como se definen en la sección 3.2.2, son aquellos para los cuales no se pudo obtener un estimado de su movimiento. La proporción de rasgos “ perdidos”, considerando todos aquellos que fueron seleccionados para su posterior correlación, se define como:

$$\frac{\text{Número de rasgos “ perdidos”}}{\text{Número de rasgos seleccionados}} \quad (34)$$

En estos experimentos se procesaron videos que no poseen, cualitativamente, un nivel exagerado de ruido. No se utilizó ninguna medida cuantitativa de la cantidad de ruido presente en las imágenes para su selección. Los videos utilizados son segmentos de los del repositorio antes mencionado, en los cuales se perciben pocas distorsiones que modifican severamente el contenido de las imágenes. Puesto que todos los videos cuentan con cerca de 30 imágenes por segundo y su proceso de selección se basó en la percepción visual de la cantidad de ruido presente en los mismos, es posible que existan

⁵El error de la estimación de movimiento puede considerarse una proyección de la calidad de la estimación del flujo óptico.

⁶La formulación matemática del error estándar de la media se presenta en el Apéndice A.

imágenes en las cuales el nivel de ruido es sumamente elevado. Sin embargo, son reducidos los casos en los que este ruido se percibe durante su reproducción. En el proceso de estimación del flujo óptico, para cada imagen en los segmentos de video utilizados, es necesario contar con otra que sirve de base. En nuestro caso, dada una imagen en particular, se considera su sucesora en la secuencia como la de referencia. De esta manera, fueron procesadas 16.477 pares de imágenes de segmentos de videos de ChocoLate y 14.909 pares de los de *YouTube.com*⁷. En todas las secuencias procesadas es ignorada la última imagen, puesto que ésta no cuenta una sucesora que permita estimar el movimiento.

4.1.1. Experimento I: Influencia del tamaño de la vecindad

El objetivo de este experimento es considerar diferentes tamaños para la ventana W , que define la vecindad sobre la cual resolver el sistema presentado en la ecuación (4); y, a partir de éstos, evaluar el comportamiento del proceso de estimación del campo de movimiento a lo largo de diferentes secuencias de imágenes. En particular, se consideraron ventanas de 3×3 , 5×5 , 7×7 , 9×9 , 11×11 y 13×13 píxeles.

Se seleccionó L_3 como la máxima profundidad de la pirámide utilizada para la correlación de rasgos, puesto que es el valor predeterminado de la implementación del algoritmo de Lucas y Kanade [31] en OpenCV. En otras palabras, la pirámide se compone de 4 niveles. En [8] se reportan como valores usuales para este parámetro L_2 , L_3 y L_4 .⁸

Considerando los ejemplos que provee la librería OpenCV, se estableció inicialmente $\beta = 0.3$ como uno de los criterios de parada en el proceso de refinamiento del vector de flujo óptico de un rasgo, tal como se define para el algoritmo 1⁹. Tanto en los ejemplos de la librería como en [8], es común observar como referencia un límite máximo de 20 iteraciones. Siguiendo las recomendaciones, esa fue la cota superior establecida sobre esta variable.

El proceso de estimación del campo de movimiento puede descomponerse en dos pasos de manera general. Primero se seleccionan “buenos” rasgos y, luego, para éstos se encuentra su correspondencia en la imagen de referencia. Según Bouguet[8], no es necesario utilizar una vecindad W grande para la selección de estos rasgos. A pesar de no ofrecer pruebas que soporten este argumento, indica que

⁷En el Apéndice K se describe de manera detallada la cantidad de secuencias consideradas de los videos descargados de *YouTube.com*, así como el contenido general de las mismas.

⁸En el Experimento II, sección 4.1.2, se ofrecen detalles la influencia de la máxima profundidad de la pirámide en diferentes aspectos del proceso de estimación del flujo óptico.

⁹En el Experimento III, sección 4.1.3, se evalúan diferentes opciones para esta variable.

una ventana 3×3 píxeles es suficiente y debe ser usada. Adicionalmente, expresa que utilizar una vecindad pequeña suele no resultar beneficioso en el proceso de correlación.

Sin embargo, está claro que la definición de un “buen” rasgo [37] para el algoritmo de Tomasi y Kanade [44] se basa en las propiedades de la matriz Z , definida en (17). Puesto que ésta, a su vez, resulta del sistema en (4), el tamaño de W según la fundamentación teórica del algoritmo es el mismo para el proceso de selección de rasgos como para su posterior correlación.

Con el fin de simplificar la notación de ahora en adelante, definiremos la función $T(W)$, que devuelve el ancho de una vecindad W ¹⁰. W_s indicará la vecindad utilizada para la selección de “buenos” rasgos, mientras que W_e se referirá a aquella para la cual se resuelve el sistema (4), obteniendo así una aproximación del vector de flujo óptico dado un rasgo. De esta manera, en los resultados que se muestran a continuación se consideraron dos opciones:

Mantener $T(W_s)$ pequeño y variar las dimensiones de W_e . Siendo $D = \{3, 5, 7, 9, 11, 13\}$ el conjunto de valores considerados en este experimento para el ancho de las regiones cuadradas W_e , se establece en este caso $T(W_s) = (\min d : d \in R : d) = 3$.

Variar por igual las dimensiones de W_s y W_e . Las regiones W_s y W_e que se utilizan para seleccionar “buenos” rasgos y obtener una correlación de éstos, respectivamente, poseen las mismas dimensiones.

Dado un tamaño para la ventana W_s , con el fin de evaluar su influencia en el proceso de estimación del campo de movimiento de una imagen, se consideró el número de buenos “rasgos” seleccionados que se obtiene en cada caso. Los resultados de los experimentos que utilizan W_s constante de 3×3 píxeles no ofrecen ninguna información sobresaliente en este aspecto, puesto que la cantidad de rasgos seleccionados es la misma para todas sus variantes.

Resultados

Para los pares de imágenes de segmentos de videos de *YouTube.com*, en la Figura 7 pareciera, ligeramente, que en promedio mientras aumenta el tamaño de W_s , disminuye la cantidad de “buenos” rasgos seleccionados. Sin embargo, no existe un patrón realmente significativo que ofrezca bases suficientes para establecer algún tipo de dependencia entre estas dos variables. En el apéndice

¹⁰Las vecindades consideradas en la implementación del algoritmo de Lucas y Kanade[31] en OpenCV, según [8], son cuadradas. En otras palabras, su ancho es igual a su largo. Entonces, la función T nos permite conocer el tamaño de una vecindad a partir de una de sus dimensiones.

E se muestran dos ejemplos detallados sobre este proceso de selección. En uno de ellos, utilizando un menor tamaño de ventana se seleccionan más rasgos; mientras que en el otro sucede lo contrario.

Un resultado diferente se observa para las imágenes capturadas desde ChocoCam. En la misma Figura 7, se percibe que a medida que aumenta el tamaño de W_s , en general, pareciera aumentar el número de rasgos seleccionados. Adicionalmente, para este grupo de imágenes se puede apreciar que, muchas veces, se alcanza la máxima cantidad de rasgos preestablecida en este experimento. De hecho, el valor de la mediana de los datos para estos casos se visualiza en el límite de 500 rasgos.

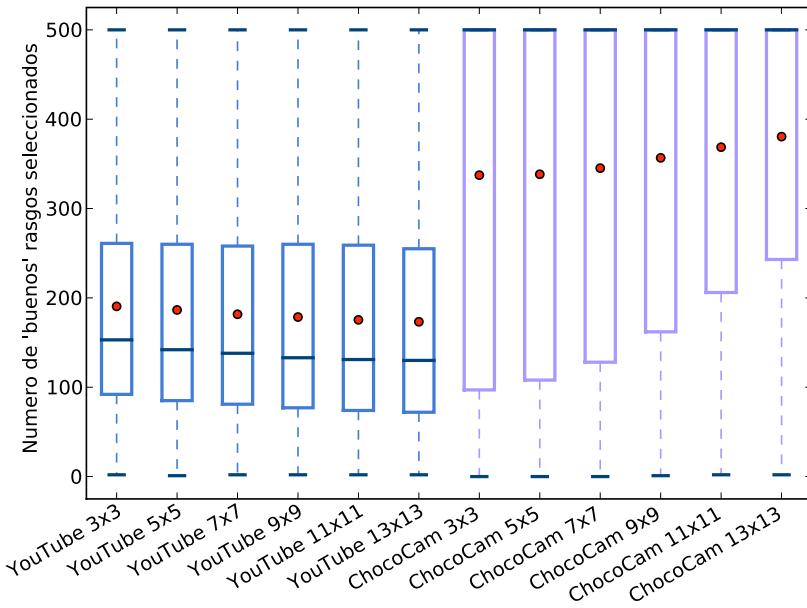


Figura 7: Número de “buenos” rasgos seleccionados para la estimación del flujo óptico entre pares de imágenes de los segmentos de videos utilizados en el Experimento I, según el origen de las imágenes procesadas y el tamaño de W_s expresado como *ancho* \times *largo*. Para cada caso considerado, el valor promedio se representa por un punto de color rojo sobre el diagrama de caja.

Podría suponerse que la variación de la cantidad de rasgos seleccionados se fundamenta en el valor del máximo de los mínimos autovalores encontrado en cada caso. De esta manera, podría establecerse una relación entre el tamaño de W_s y el número de “buenos” rasgos seleccionados, puesto que las dimensiones de la vecindad modifican la cantidad de píxeles que son considerados en la ecuación (17). Entonces, dado que la matriz Z cambia según W_s , sus los autovalores posiblemente también cambiarán y esto podría explicar la cantidad de rasgos que se procesan en cada caso. Así, pudiese darse el caso que en las imágenes con mayor nivel de calidad, la cantidad de rasgos

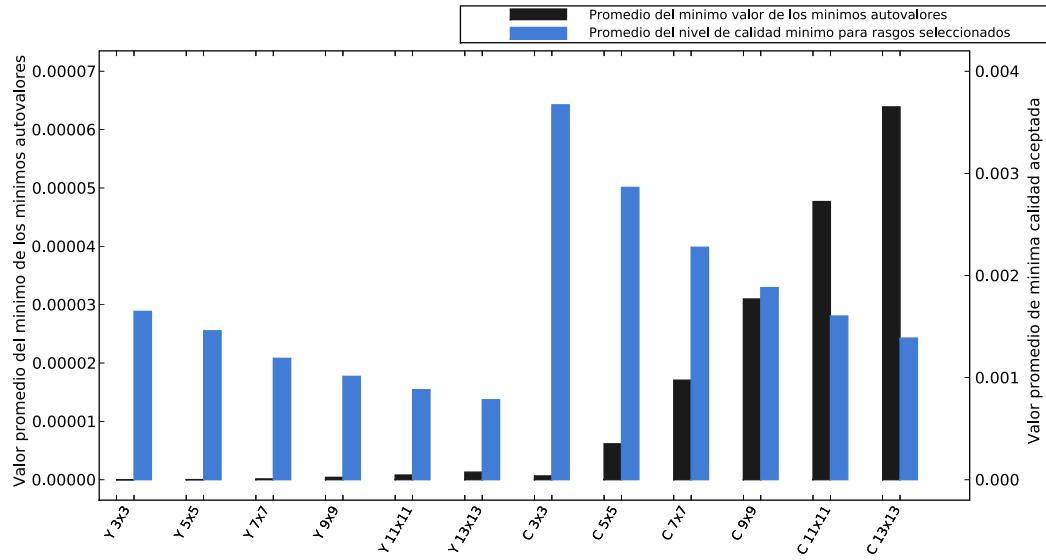


Figura 8: Promedio del mínimo valor de los mínimos autovalores (barras de color negro) y del nivel de calidad establecido (barras azules) entre pares de imágenes de los segmentos de videos utilizados en el Experimento I, según el repositorio al cual pertenecen las imágenes procesadas (“Y” para *YouTube.com* y “C” para aquellas capturadas desde ChocoLate) y el tamaño de W_s expresado como *ancho* \times *largo*.

seleccionados es menor.

En la Figura 8 se muestra el promedio del mínimo valor de los mínimos autovalores calculados para la selección de rasgos por par de imágenes, según su origen y el tamaño de la vecindad W_s utilizada. Igualmente se presenta el nivel de calidad en promedio establecido para cada caso, considerando el 0.05 % del máximo de los mínimos autovalores. En el Apéndice L.1 se ofrecen más detalles de estos resultados, así como el pequeño error estándar de los promedios obtenidos en cada caso.

De manera general se puede observar que los valores promedio del mínimo de los mínimos autovalores encontrados son sumamente pequeños. Se puede apreciar que para las imágenes provenientes de videos capturados desde ChocoLate, a medida que aumentan las dimensiones de la vecindad, también parece aumentar, en promedio, este mínimo valor. Sin embargo, para aquellas de videos de *YouTube.com* no se percibe una variación significativa.

Claramente, mientras más pequeña es W_s , mayor nivel de calidad promedio se observa. Si consideramos que para las imágenes capturadas desde el helicóptero del GIA, a medida que crecen las dimensiones de W_s , aumenta el mínimo valor de los mínimos autovalores y disminuye el nivel de calidad; entonces, parece razonable que aumente el número de rasgos seleccionados, tal como

se percibe en la Figura 7. Utilizando el coeficiente de correlación de Pearson¹¹ como una medida de cuánto se relacionan dos variables, podemos observar en el Cuadro 1 que, en promedio para los pares de imágenes de este grupo, mientras disminuye el máximo valor de los mínimos autovalores calculados, aumenta el número de “buenos” rasgos seleccionados por cada par de imágenes¹². Así, mientras disminuye el nivel de calidad, aumenta la cantidad de rasgos “buenos” que se procesan para estimar el campo de movimiento.

Se puede observar que los resultados en este aspecto para las imágenes de *YouTube.com* indican una baja dependencia lineal entre el número de rasgos seleccionados y el nivel de calidad establecido en estos casos, sobretodo para las vecindades más pequeñas. En la Figura 21, del Apéndice L.1, se muestran los resultados obtenidos para W_s de 3×3 píxeles, relacionando el máximo valor de los mínimos autovalores con el número de rasgos seleccionados.

Origen	Tam. W_s	$r(N, MAX)$	$r(N, MIN)$	$r(MIN, MAX)$
<i>YouTube.com</i>	3×3	-0.099730	0.108286	0.069119
	5×5	-0.060082	0.135622	0.109404
	7×7	-0.054425	0.200109	0.152160
	9×9	-0.076974	0.252208	0.187682
	11×11	-0.094439	0.269093	0.189129
	13×13	-0.120304	0.260756	0.188660
<i>ChocoCam</i>	3×3	-0.717427	0.263936	-0.052368
	5×5	-0.669433	0.295785	-0.030798
	7×7	-0.661654	0.305506	0.002656
	9×9	-0.637806	0.309741	0.042506
	11×11	-0.623127	0.320416	0.065290
	13×13	-0.602507	0.340267	0.071053

Cuadro 1: Las variables N , MAX y MIN representan el número de rasgos, el valor del máximo y el del mínimo de los mínimos autovalores para cada par de imágenes procesadas, respectivamente. $r(X, Y)$ representa el coeficiente de correlación de Pearson entre las variables X e Y , tal como se define en el Apéndice A. Los pares de imágenes procesadas se agrupan según su origen y el tamaño de la vecindad W_s utilizada en cada caso.

En el Cuadro 1 se puede apreciar que el mayor nivel de correlación entre el mínimo valor de los mínimos autovalores encontrados y el número de rasgos seleccionados, en promedio, se tiene para los pares de imágenes capturadas desde ChocoLate, utilizando W_s de 13×13 píxeles. Todos los valores de correlación para estas dos variables son positivos, por lo cual pareciera que existe la

¹¹El coeficiente de correlación de Pearson se define matemáticamente en el Apéndice A.

¹²El coeficiente de correlación de Pearson para el número de rasgos seleccionados y el valor del máximo de los mínimos autovalores encontrado, en promedio por par de imágenes, es equivalente al que se obtiene entre el mínimo nivel de calidad establecido y el número de rasgos. Esto se debe a que el nivel de calidad es el 0.05 % del máximo de los mínimos autovalores.

tendencia que mientras mayor son los mínimos autovalores, entonces más rasgos se seleccionan. A medida que aumentan las dimensiones de W_s , esto se observa con más fuerza considerando el valor del coeficiente en cada caso.

Por otro lado, vale destacar que no se observa una correlación lineal sobresaliente entre el mínimo y el máximo de los mínimos autovalores calculados por pares de imágenes. Se podría esperar un coeficiente de correlación negativo entre estas variables; sin embargo, ni siquiera se aprecia una tendencia única en los valores de la quinta columna del Cuadro 1. Algunos son positivos y otros negativos.

En el Cuadro 4 del Apéndice L.1, se presenta el promedio del error de estimación de movimiento obtenido para W_e de diferentes tamaños, siendo ésta la vecindad utilizada para la estimación del vector de flujo óptico dado un rasgo. En estos resultados se consideran los dos casos propuestos sobre W_s ; es decir, cuando $T(W_s)$ es mínimo y cuando $T(W_s) = T(W_e)$. Adicionalmente, en el Cuadro 5 se muestra el pequeño error estándar de estos promedios¹³. Considerando que mientras más grande es W_e , más términos son considerados en la ecuación (1), en la Figura 9 se presenta el error promedio escalado, que hace referencia al error promedio por píxel en la vecindad utilizada. Estos valores se obtienen dividiendo el error obtenido entre la cantidad de píxeles que conforman W_e .

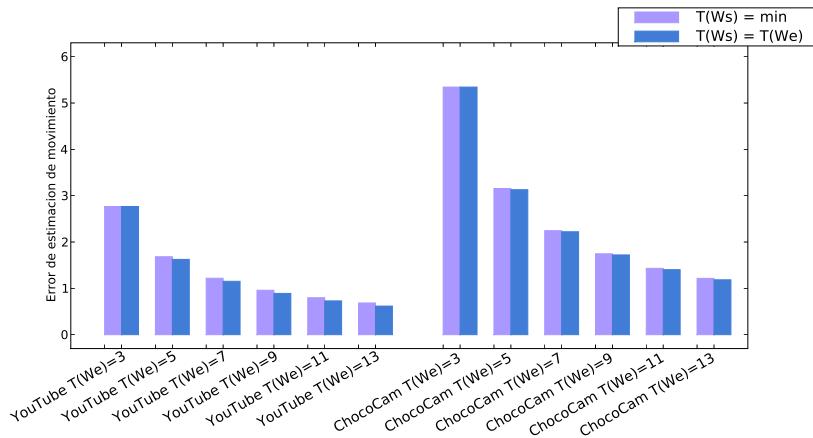


Figura 9: Error promedio escalado de la estimación de movimiento en pares de imágenes de los segmentos de videos seleccionados para el Experimento I, agrupados según el origen de estos pares y las dimensiones de W_s y W_e . Este error se obtiene dividiendo el promedio de error encontrado entre la cantidad de píxeles que conforman W_e .

¹³En el Apéndice A se ofrece la definición matemática del error estándar de la media.

Se puede observar que el error promedio para los pares de imágenes capturadas desde el helicóptero del GIA, dados W_s y W_e , es mayor que el promedio que se registra para aquellas pertenecientes a segmentos de videos de *YouTube.com*. Este hecho puede atribuirse a la mayor cantidad de ruido en las que fueron transmitidas analógicamente desde ChocoLate.

Si bien, originalmente, con un W_s más grande se tienen mayores errores promedio; al escalar los resultados según el tamaño de la vecindad, se observa como tendencia que mientras mayor es $T(W_s)$, menor error promedio se obtiene al estimar el movimiento de los rasgos. Ello nos indica que en las ventanas muy pequeñas el error promedio de estimación por píxel es mayor al que se obtiene utilizando ventanas más grandes. Resulta natural pensar que en estos casos es más factible caer en mínimos locales que realmente no representan el mejor estimado de movimiento que se hubiese podido encontrar.

Dado un grupo de pares de imágenes procesadas según su origen y W_e de cierto tamaño, el error promedio de estimación de movimiento utilizando W_s de 3×3 píxeles es mayor o igual que el error que se obtiene al utilizar $W_s = W_e$. Al considerar el error promedio por píxel, esta diferencia no es tan notoria para las imágenes capturadas desde ChocoLate, aún cuando todavía se percibe para las de *YouTube.com*. Considerando la fundamentación teórica del algoritmo de Lucas y Kanade[31] y la definición de un “buen” rasgo, según [37], parece razonable este resultado. Al utilizar un mismo tamaño para la vecindad en el proceso de selección de rasgos como para su posterior correlacion, se garantiza que los rasgos cuyo movimiento se estima son los mejores a procesar.

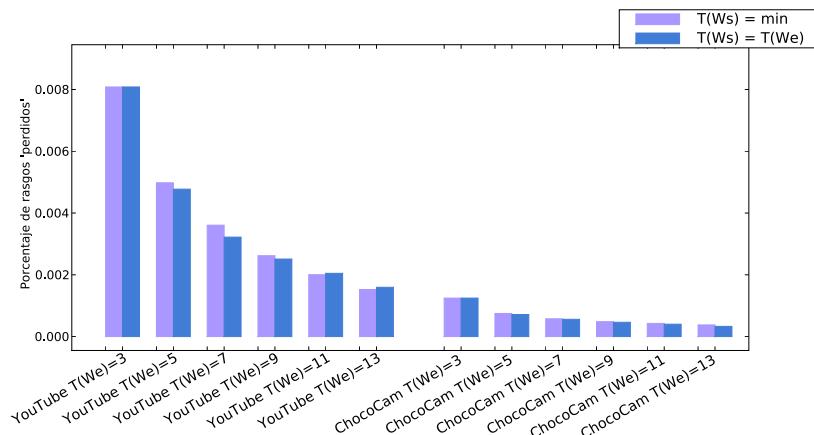


Figura 10: Proporción de rasgos “perdidos” en función de la cantidad de rasgos seleccionados entre todos los pares de imágenes de un mismo origen en los segmentos de video utilizados en el Experimento I, agrupados según las dimensiones de W_s y W_e .

La Figura 10 refleja el porcentaje de rasgos seleccionados para los cuales la estimación de su posición en la imagen que sirve de referencia sobresale sus dimensiones¹⁴. Tal como fue mencionado anteriormente, estos rasgos son clasificados como “perdidos” [8] y no son útiles para el proceso de estabilización. Puesto que la magnitud de estos porcentajes es realmente pequeña, las variaciones en estos promedios no parecen significativas. En la gran mayoría de pares de imágenes procesadas es usual no obtener rasgos “perdidos”.

El factor más interesante en este aspecto es que la magnitud de los promedios de rasgos “perdidos” para las imágenes de *YouTube.com* son superiores a los que se obtuvieron para aquellas de videos capturados desde ChocoLate. Estos rasgos, que resultan poco útiles, usualmente se deben a un movimiento significativo de la cámara durante el proceso de captura. Bajo esta circunstancia es más probable que los rasgos cerca de los bordes no puedan ser correlacionados, tal como se muestra en la Figura 22 del Apéndice L.1, utilizando la implementación piramidal del algoritmo de Lucas y Kanade [31]. Cuando se registraron los videos desde el helicóptero del GIA, usualmente se buscaba mantener al vehículo aéreo fijo en un punto en el aire, por lo cual son reducidos los casos donde la magnitud de la velocidad del movimiento de la cámara es elevada.

Conclusiones

En este experimento se busca estudiar la influencia del tamaño de las ventanas W_s y W_e empleadas durante el proceso de estimación de flujo óptico utilizado en la aplicación de estabilización implementada.

El número de rasgos seleccionados, considerando todos los pares de imágenes procesados, no es constante en promedio. Al menos los resultados no muestran contradicciones al agrupar las imágenes según su origen. Para aquellas capturadas desde ChocoLate, la disminución del nivel de calidad mientras aumenta el mínimo valor de los mínimos autovalores, pareciera ir en concordancia con el mayor número de rasgos seleccionados al aumentar las dimensiones de W_s .

El coeficiente de correlación no es prueba suficiente para establecer una relación causal entre las variables consideradas. Los resultados obtenidos podrían atribuirse al nivel de ruido presente en los segmentos de video de donde son extraídas las imágenes. La magnitud del movimiento de la cámara, que se percibe en las secuencias utilizadas en el experimento, también podría influir en

¹⁴Los resultados obtenidos respecto al porcentaje de rasgos “perdidos” se detallan en el Cuadro 6 del Apéndice L.1.

éstos.

Considerando que el error de estimación depende de las dimensiones de W_e y que la utilización de una ventana pequeña en este proceso suele generar un mayor porcentaje de error promedio por píxel que la compone, pareciera que utilizar W_e de poco tamaño no ofrece beneficios significativos al aproximar el campo de movimiento.

En general, los resultados al utilizar $W_s = W_e$ son superiores a los de W_s de 3×3 píxeles. A pesar que éstos difieren poco cuando se considera el error promedio de estimación escalado, de este experimento no pueden derivarse conclusiones que favorezcan la utilización de $T(W_s)$ mínimo.

4.1.2. Experimento II: Influencia del máx. nivel de profundidad de la pirámide

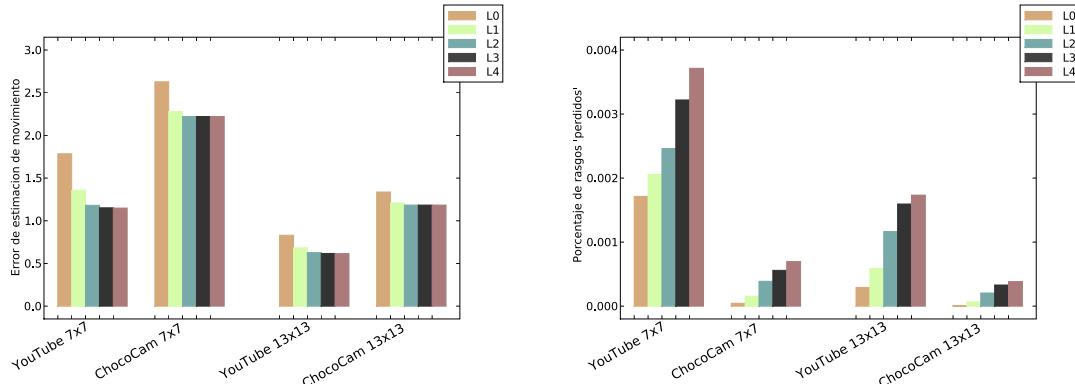
Este experimento tiene como objetivo evaluar de manera sencilla si el máximo nivel de profundidad de la pirámide utilizada afecta significativamente el proceso de estimación del flujo óptico, considerando como valores posibles L_0 , L_1 , L_2 , L_3 y L_4 , siguiendo la terminología utilizada en la sección 3.2.2. Al igual que en el experimento anterior se optó por establecer como condiciones de parada del algoritmo iterativo, $\beta = 0.3$ y 20 iteraciones máximo. Con el fin de mostrar resultados bajo diversas situaciones, se utilizó una ventana W de 7×7 y 13×13 píxeles, de manera consistente para la selección de rasgos como para la correlación de los mismos. Se descartaron ventanas muy pequeñas dado que con ellas se encontraron los mayores errores promedio de estimación por píxel.

Los resultados en este caso consideran, únicamente, el error de estimación movimiento y la proporción de rasgos “perdidos”, definidos en la sección 4.1.1.

Resultados

En la Figura 11(a) se muestra el error promedio de la correlación de “buenos” rasgos escalado (por píxel en W), utilizando los niveles de profundidad máxima establecidos para este experimento. El caso base que limita a un único nivel de profundidad L_0 , que equivale a utilizar una implementación no piramidal del algoritmo de Lucas y Kanade [31], sobresale como aquel con mayor error en promedio. Son reducidas las variaciones que se observan entre los resultados que se presentan al utilizar como máximo nivel L_2 , L_3 y L_4 . Los valores obtenidos en cada caso para el promedio del error, así como su correspondiente error estándar, se presentan con más detalle en el Apéndice L.2.

En la Figura 11(b) se presenta el porcentaje de rasgos “perdidos” en promedio, tal como se detallan en el Cuadro 9 del Apéndice antes mencionado. Claramente se observa que mientras mayor



(a) Error promedio de la estimación de movimiento por píxel según el nivel máximo de profundidad de la pirámide utilizada

(b) Proporción de rasgos “perdidos” considerando diferentes niveles máximos de profundidad para la pirámide utilizada

Figura 11: Resultados del Experimento II

profundidad posee la pirámide utilizada, mayor número de “buenos” rasgos poseen correspondencia fuera de la imagen de referencia y, por lo tanto, deben ser descartados en pasos siguientes del proceso de estabilización.

Conclusiones

Es evidente que la utilización de una pirámide mejora el proceso de correlación de rasgos en función del error de estimación de movimiento, en promedio. En particular, los máximos L_2 , L_3 y L_4 , que se refieren al tercer, cuarto y quinto nivel de profundidad respectivamente, ofrecen los mejores resultados entre las opciones consideradas. Adicionalmente, mientras menos niveles se utilizan, menor porcentaje de rasgos “perdidos” se registran en el experimento.

Dado que se desea que la aplicación de estabilización pueda correr en tiempo real de la manera más robusta posible y se busca obtener la menor cantidad de rasgos “perdidos”, la mejor opción resulta ser la utilización de L_2 como máximo. Vale destacar que esto aplica únicamente para el procesamiento de imágenes de 320×240 píxeles. En este sentido, utilizando el doble del tamaño, por ejemplo, se espera que un máximo de 3 niveles de profundidad no sea la mejor opción.

Los resultados del Experimento I, sección 4.1.1, se obtuvieron utilizando una pirámide de máximo 4 niveles de profundidad. Si bien ahora se recomienda utilizar 3 niveles con el fin de optimizar el proceso de estimación del campo de movimiento, las conclusiones derivadas previamente siguen siendo relevantes. En este experimento, al variar las dimensiones de W se obtienen resultados

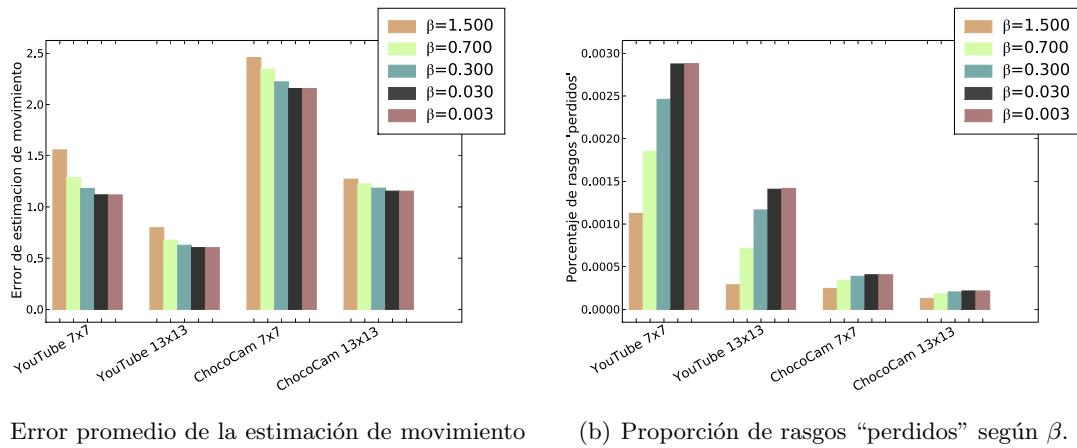
coherentes con los anteriores.

4.1.3. Experimento III: Influencia de la cota mínima β como criterio de parada

Siguiendo con la evaluación de las variables que influencian el proceso de estimación de flujo óptico, en este experimento se consideraron los siguientes valores para la cota mínima β : 1.5, 0.7, 0.3, 0.03 y 0.003. Esta cota establece un criterio de parada en el proceso de refinamiento del vector de flujo óptico de un “buen” rasgo, tal como se explica en la sección 3.2.2.

En este caso se optó por utilizar pirámides de máximo 3 niveles de profundidad. Como condición adicional de parada, 20 es el número máximo de iteraciones que pueden llevarse a cabo. Para las dimensiones de la ventana W , sobre la cual resolver el sistema 4, se consideraron como opciones 7×7 y 13×13 píxeles, de manera consistente para la selección y correlación de rasgos.

Resultados



(a) Error promedio de la estimación de movimiento por píxel en W según β .
(b) Proporción de rasgos “perdidos” según β .

Figura 12: Resultados del Experimento III.

En la Figura 12(a) se observa que mientras más pequeña es la cota β , menor error promedio de estimación de movimiento por píxel en W se obtiene, considerando cada uno de los “buenos” rasgos seleccionados en los pares de imágenes procesadas para los que se encontró una correspondencia exitosamente. En los Cuadros 10 y 11, del Apéndice L.3, se presentan detalladamente las medias de los errores obtenidos para cada caso y su respectivo error estándar.

La proporción de rasgos “perdidos” se presenta en la Figura 12(b) y, con más detenimiento, en el Cuadro 12 del Apéndice L.3. Si bien el valor de estas proporciones resulta pequeño, dada

la gran cantidad de pares de imágenes procesadas, se aprecia que mientras más pequeña es la cota β , mayor proporción se obtiene en general. Vale destacar que, de manera concordante con los resultados obtenidos en el Experimento I, sección 4.1.1, mientras más grande es W , menor proporción se observa en los resultados.

Conclusiones

Claramente mientras menor es β , más precisión se tiene en la estimación del vector de flujo óptico de un rasgo. Esto podría influir en la proporción de rasgos “perdidos”.

Los resultados obtenidos con el menor valor de β considerado no representan mayores ventajas. Las diferencias en éstos entre 0.003 y 0.03 son mínimas. Entonces, en comparación con los otros valores estudiados, siempre que sea posible es recomendable utilizar 0.03 como valor para este criterio de parada en el proceso iterativo de estimación.

Pareciera que considerar diferentes valores a 20 para el número de iteraciones máximas no ofrecería resultados interesantes. Esto se debe a que variando únicamente el valor de β se obtuvieron diferentes promedios de error de estimación, lo cual evidencia que realmente pone fin al proceso de aproximación del vector de flujo óptico. En consecuencia, el máximo número de iteraciones no es alcanzado en estos casos y una variación del valor de esta cota o bien restaría calidad al estimado o no afectaría significativamente los resultados.

4.2. Evaluación del proceso de compensación

El grupo de experimentos que se presenta a continuación tiene como objetivo evaluar los diferentes aspectos que, una vez estimado el campo de movimiento, permiten estabilizar una secuencia de imágenes.

4.2.1. Experimento IV: Evaluación del modelo y del método de estimación

En este experimento se desea comparar qué tan bien explican diferentes transformaciones el movimiento percibido desde una cámara posicionada rígidamente en un helicóptero a control remoto, considerando los modelos y métodos de estimación implementados según la sección 3.2.3.

Para evaluar la calidad del modelo utilizado se emplea una relación conocida, en inglés, como

peak signal-to-noise ratio (PSNR)[32], tal que

$$\text{PSNR}(g, f) = 10 \log \frac{255}{\text{MSE}(g, f)} \quad (35)$$

donde

$$\text{MSE}(g, f) = \frac{1}{(w)(h)} \sum_{i=0}^{w-1} \sum_{j=0}^{h-1} (g(i, j) - f(i, j))^2 \quad (36)$$

y w y h representan las dimensiones de las imágenes f y g comparadas. Mientras menos diferencia se tiene entre los valores de intensidad de f y g , mayor será el valor de esta relación. Puesto que al compensar una imagen quedan en ella regiones sin información, la fórmula se modificó para considerar únicamente el área válida que resulta al aplicar la transformación. De esta manera, es razonable comparar el PSNR obtenido originalmente con el que resulta al compensar.

Con el fin de establecer una medida de qué tan ventajosa resulta la compensación de una imagen f a partir de la estimación del movimiento global que se percibe entre ésta y su sucesora g , se define la “ganancia” como

$$\text{PSNR}(g_c, f) - \text{PSNR}(g, f) = 10 \log \frac{\sum_{i=0}^{w-1} \sum_{j=0}^{h-1} (g(i, j) - f(i, j))^2}{\sum_{i=0}^{w-1} \sum_{j=0}^{h-1} (g_c(i, j) - f(i, j))^2} \quad (37)$$

siendo g_c la imagen compensada. Esta relación nos permite estimar en cuánto se reduce la diferencia de intensidad entre imágenes consecutivas en la secuencia al aplicar la transformación encontrada.

En este experimento se procesaron segmentos de videos tanto del repositorio de *YouTube.com*, como del grupo de los que fueron capturados desde el helicóptero del GIA. En particular, se buscó tener un repertorio variado de movimientos de cámara, así como de los ambientes que conforman las escenas. En total, se llevaron a cabo las pruebas con 1785 pares de imágenes de los videos de Internet y 1881 de las secuencias capturadas desde ChocoLate.

Si bien en este caso se procesan menos imágenes en comparación a aquellas utilizadas en la fase de experimentación anterior, reducir su cantidad fue necesario en función de los factores que influencian el proceso de estimación del movimiento global. En este sentido, las pruebas se llevaron a cabo considerando 4 combinaciones del modelo y el método de estimación utilizados. Para cada una de éstas se evalúa la influencia de diferentes umbrales ε , que determinan los datos atípicos dentro del conjunto de vectores de flujo óptico que son estimados según el algoritmo de Lucas y

Kanade [31]. El grupo de valores estudiados de ε se compone por 15.0, 8.0, 5.0, 3.0, 1.0 y 0.25.

También se establecen como máximo 1, 2, 3 y hasta 4 iteraciones en el proceso de refinamiento de la transformación. En vista de la necesidad de poder correr la aplicación de estabilización en tiempo real, se consideró como máxima cardinalidad del conjunto de “buenos” rasgos seleccionados 500, 300 y 100 puntos. Estos casos adicionales en los que se limita aún más la cantidad de rasgos se deben a pequeñas pruebas iniciales donde se evidencia un tiempo de procesamiento elevado al agregar al proceso de estabilización la fase de estimación del movimiento global.

Se decidió continuar el procesamiento con imágenes de 320 píxeles de ancho, con una distancia mínima entre rasgos de 5 píxeles. En función de los resultados obtenidos en el grupo de experimentos anterior, se utilizan vecindades W de 7×7 y 13×13 píxeles. Así mismo, se emplean una pirámide de 2 niveles de profundidad máximo y $\beta = 0.03$.

Resultados

La ganancia promedio obtenida en cada caso y el error estándar de este valor se presenta en la sección M.1.1. Cualquiera de los modelos de transformación considerados logra disminuir, en promedio, la diferencia entre los valores de intensidad de los píxeles que conforman las imágenes de los pares procesados, en comparación a no aplicar ninguna transformación. En el Cuadro 2, donde se presentan los mejores resultados según el modelo y el método de estimación utilizado para un máximo de 100 rasgos, se observa que todas las ganancias son significativamente positivas.¹⁵

En las tablas que resumen los mejores resultados sobre la ganancia, se percibe que mientras mayor es el número de iteraciones máximas que se pueden llevar a cabo, mayor ganancia promedio se obtiene. Entre no refinar el modelo, o llevar a cabo una única iteración, y mejorarlo detectando vectores atípicos, es claro que con esta última opción se logra aumentar la calidad de la compensación. En los cuadros que detallan todos los resultados en la sección M.1.1 esta propiedad se cumple siempre que $\varepsilon > 1$, considerando las opciones propuestas para ε .

El aumento de la ganancia promedio parece despreciable, de manera general, al aumentar el máximo número de ciclos de 2 a 3 o a 4. Este hecho se refleja en los resultados que se presentan en la sección M.1.2, donde se muestra el promedio de iteraciones llevadas a cabo para mejorar el

¹⁵Los cuadros que resumen los resultados para máximo 500 y 300 rasgos se muestran en el Apéndice M.1.1. En esta sección únicamente se muestran los resultados para 100 rasgos máximo, puesto que los otros cumplen las mismas propiedades que éstos y, tal como se concluye en el Experimento V, sección 4.2.2, un máximo de 100 puede ser usado para correr la aplicación en tiempo real satisfactoriamente.

Origen	$T(W)$	Modelo/Método	Máx. 1 It.	Máx. 2 Its.	Máx. 3 Its.	Máx. 4 Its.
<i>You Tube</i>	7	Afín/M.C.	10.350	10.705 ($\varepsilon = 8.0$)	10.724 ($\varepsilon = 8.0$)	10.730 ($\varepsilon = 8.0$)
		Afín/M.C.T	10.336	10.713 ($\varepsilon = 5.0$)	10.721 ($\varepsilon = 8.0$)	10.727 ($\varepsilon = 5.0$)
		Similar/M.C.	8.519	8.743 ($\varepsilon = 15.0$)	8.766 ($\varepsilon = 15.0$)	8.769 ($\varepsilon = 15.0$)
		Bilineal/M.C	10.477	10.826 ($\varepsilon = 5.0$)	10.863 ($\varepsilon = 8.0$)	10.866 ($\varepsilon = 8.0$)
	13	Afín/M.C	10.595	10.721 ($\varepsilon = 3.0$)	10.730 ($\varepsilon = 5.0$)	10.736 ($\varepsilon = 5.0$)
		Afín/M.C.T	10.592	10.725 ($\varepsilon = 3.0$)	10.730 ($\varepsilon = 3.0$)	10.739 ($\varepsilon = 5.0$)
		Similar/M.C.	8.744	8.788 ($\varepsilon = 15.0$)	8.791 ($\varepsilon = 15.0$)	8.791 ($\varepsilon = 15.0$)
		Bilineal/M.C	10.718	10.829 ($\varepsilon = 3.0$)	10.839 ($\varepsilon = 3.0$)	10.844 ($\varepsilon = 3.0$)
<i>ChocoCam</i>	7	Afín/M.C	4.832	4.921 ($\varepsilon = 3.0$)	4.925 ($\varepsilon = 3.0$)	4.926 ($\varepsilon = 3.0$)
		Afín/M.C.T	4.806	4.904 ($\varepsilon = 3.0$)	4.907 ($\varepsilon = 3.0$)	4.910 ($\varepsilon = 3.0$)
		Similar/M.C.	4.735	4.793 ($\varepsilon = 8.0$)	4.796 ($\varepsilon = 8.0$)	4.796 ($\varepsilon = 8.0$)
		Bilineal/M.C	4.602	4.705 ($\varepsilon = 5.0$)	4.715 ($\varepsilon = 3.0$)	4.715 ($\varepsilon = 5.0$)
	13	Afín/M.C	4.929	4.950 ($\varepsilon = 5.0$)	4.957 ($\varepsilon = 5.0$)	4.958 ($\varepsilon = 5.0$)
		Afín/M.C.T	4.926	4.946 ($\varepsilon = 5.0$)	4.953 ($\varepsilon = 5.0$)	4.956 ($\varepsilon = 5.0$)
		Similar/M.C.	4.792	4.810 ($\varepsilon = 8.0$)	4.817 ($\varepsilon = 8.0$)	4.817 ($\varepsilon = 8.0$)
		Bilineal/M.C	4.771	4.822 ($\varepsilon = 3.0$)	4.830 ($\varepsilon = 3.0$)	4.833 ($\varepsilon = 3.0$)

Cuadro 2: Mejores ganancias obtenidas en promedio para un máximo de 100 rasgos. Entre paréntesis se indica el ε utilizado en cada caso.

estimado de la transformación, utilizando los mejores valores de ε según el caso. El error estándar encontrado es sumamente pequeño y nos hace pensar que pocas veces se alcanza el máximo número de ciclos establecido, sobre todo cuando se tratan de 4.

El rango de valores de ε considerados en este experimento pudiesen no incluir una buena opción para llevar a cabo la estimación con un modelo similar, puesto que muchas veces se obtienen los mejores resultados con $\varepsilon = 15.0$. Bajo este tipo de transformación, la calidad de la compensación de las imágenes de videos del repositorio de *YouTube.com* es significativamente menor a la que se obtiene utilizando un modelo afín. Para las secuencias capturadas desde el helicóptero del GIA, la diferencia en los resultados no es tan notable. El ruido presente en estas imágenes, debido a su transmisión analógica, podría ser alguna de las causas que influye en la obtención de resultados diferentes.

Entre utilizar un modelo afín o uno bilineal no se observan grandes diferencias en cuanto a la ganancia promedio. El uso de mínimos cuadrados totales con respecto al método clásico no parece ofrecer ventajas. A pesar de ser muy pequeña la diferencia, el primero sólo supera al segundo cuando se consideran los resultados de las imágenes provenientes de *YouTube.com* con un máximo de 2 iteraciones para refinar el modelo.

El porcentaje de área válida de las imágenes que se resulta luego de haberlas compensado

se presenta, en promedio, en la sección M.1.3. En general, el espacio que ocupa su contenido inicialmente se reduce a cerca de un 98 % de su valor original.

Conclusiones

Todos los modelos considerados para compensar el movimiento que se percibe entre los pares de imágenes procesadas ofrecen resultados significativos. Ante su utilización se observa, en promedio, una disminución importante en la diferencia de intensidad entre las imágenes de un par. La “ganancia” establecida como medida de comparación nos indica que la fase de compensación es efectiva, sin embargo, no permite definir claramente cuál modelo es mejor. Es importante destacar que estos resultados aplican cuando el tiempo entre la captura de las imágenes procesadas es reducido, tal como sucede en las secuencias que se utilizaron en este experimento. Si este no fuera el caso, entonces un modelo afín podría no ser suficiente modelar el movimiento percibido, tal como parece ocurrir en el sistema implementado por Jung y Sukhatme [28].

Bajo un modelo afín o uno bilineal, el uso de ε entre 3.0 y 8.0, considerando el conjunto de valores planteados para esta variable, parece ofrecer los mejores resultados en promedio. En el caso de una transformación similar, las compensaciones de mayor calidad se obtuvieron con un ε mayor o igual a 8.0. Para este modelo, cuando la mejor opción del umbral es 15.0, parece que el rango de valores considerados no es lo suficientemente amplio para abarcar la que podría ser la mejor opción.

Evidentemente existe cierta dependencia del proceso de compensación total de las imágenes sobre ε . No se recomienda la utilización de $\varepsilon = 0.25$ o $\varepsilon = 1$, puesto que al refinar el modelo no siempre se obtienen mejores resultados.

Un aspecto positivo es que el origen de las imágenes procesadas no parece afectar relevantemente el valor que debe considerarse para esta cota en función de obtener mayor calidad en la compensación. De esta manera, si consideramos ε dentro de los rangos antes mencionados, según el modelo utilizado, no parece importar si las imágenes provienen de *YouTube.com* o de ChocoLate.

El área que resulta en promedio luego de la compensación ronda el 98 % del tamaño original de las imágenes. Las variaciones de los resultados en este aspecto no demuestran diferencias relevantes al cambiar el modelo utilizado.

Por otro lado, en el experimento se percibe que el máximo número de iteraciones establecido para refinar la transformación que modela el movimiento global poco influye en la cantidad promedio de

ciclos llevados a cabo finalmente. En este sentido, podemos afirmar que el proceso de estimación tiende a converger rápidamente.

4.2.2. Experimento V: Evaluación de la velocidad de procesamiento

Una vez obtenida la transformación que modela el movimiento global, sólo falta estabilizar en cierto grado la secuencia de imágenes para disminuir las vibraciones que en ella se perciben. Es importante la rapidez de este proceso puesto que se desea que la aplicación de estabilización corra en tiempo real.

Considerando los mismos modelos de transformación de la sección anterior y una vecindad de 15 transformaciones, según la ecuación (27), en este experimento se busca evaluar la factibilidad de procesar en tiempo real la secuencia de imágenes. Se optó por utilizar como valor para ε la mejor opción encontrada anteriormente, tal como se detalla en el Cuadro 51 del Apéndice M.2. Se estableció como criterio de parada para la estimación de la transformación que modela el movimiento global, un máximo de 4 iteraciones. Así como se propuso para el primer experimento, sección 4.1.1, en éste se consideran dos casos: $T(W_s) = 3$ o $T(W_s) = T(W_e)$; siendo W_s la ventana utilizada para la selección de “buenos” rasgos y W_e la que define el área sobre la cual estimar los vectores de flujo óptico, según el algoritmo de Lucas y Kanade [31]. En particular, para $T(W_e)$ se utilizó como conjunto de valores {7, 13}.

Las pruebas se llevaron a cabo sobre cuatro secuencias de imágenes que denominaremos de la 1 a la 4. La primera, de 2 minutos y medios, presenta a personas que se desplazan a lo largo de un campo amplio; la segunda, de igual duración, muestra a un robot terrestre que se mueve también en un área abierta; las dos últimas, de 90 y 60 segundos respectivamente, se distinguen por el movimiento de un carro que, en algunos casos, atraviesa completamente el plano de las imágenes. De esta manera, en total se consideraron 7 minutos y medio de video grabados desde el helicóptero del GIA, lo cual equivale, aproximadamente, a 13500 imágenes. Con estos videos, se reprodujo el marco de trabajo que se reseña en el capítulo 3 utilizando la *HandyCam*.

Las imágenes transmitidas desde la cámara se reciben con un ancho de 640 píxeles; sin embargo, para su procesamiento y visualización se transformaron para tener un ancho de 320 píxeles.

Resultados

Al probar estabilizar imágenes utilizando 500 rasgos “buenos” como máximo, muchas veces el video se visualiza entrecortado. Tanto al reducir esta cota superior como el tamaño de W_e , de 13 a 7, se logra mejorar la velocidad. Sin embargo, al considerar los resultados promedios que se obtuvieron de todas las secuencias procesadas, no se observa ventaja al utilizar $T(W_s)$ de 3 píxeles, en comparación a $T(W_s) = T(W_e)$, según el caso. Por medio de la Figura 13, generada a partir del Cuadro 56 del Apéndice M.2, se pueden visualizar estos resultados.

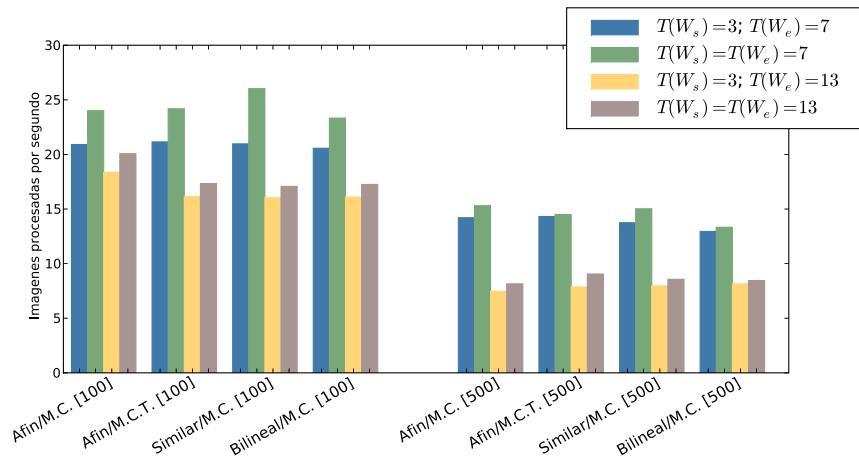


Figura 13: Promedio del aproximado número de imágenes procesadas por segundo en total (considerando todas las secuencias utilizadas en el Experimento V). Los resultados se agrupan según el modelo de transformación y el método de estimación utilizados, así como según la máxima cantidad de rasgos “buenos” que pueden seleccionarse (indicada entre corchetes).

Si bien en algunos momentos durante la estabilización de los videos se procesan cerca de 30 imágenes por segundo, esta velocidad no es constante. En los peores casos bajo la mejor configuración en promedio, utilizando un máximo de 100 rasgos que pueden seleccionarse y $T(W_s) = T(W_e) = 7$, la media del aproximado de imágenes procesadas por segundo se acerca a 20. En los mejores, fácilmente llega 28. Estos resultados pueden apreciarse en los Cuadros 52, 53, 54 y 55 del Apéndice M.2.

Conclusiones

En este experimento se encontró que establecer como máximo 500 rasgos a procesar no es factible si se desean estabilizar imágenes en tiempo real, de manera satisfactoria, a un tamaño de 320×240 píxeles y bajo la configuración establecida. Utilizando 100 máximo, exitosamente se pudo

obtener un resultado donde se aprecia una mejora en la calidad de las secuencias consideradas, con una cota superior de 4 sobre el número de iteraciones para refinar el modelo que representa el movimiento global.

Finalmente se puede establecer un punto de comparación interesante al variar el tamaño de W_s entre 7×7 y 13×13 píxeles. Utilizando una ventana más pequeña se logra agilizar el proceso. Por otro lado, al usar W_e mínimo de 3 píxeles, tal como indica Bouget [8] que debe hacerse, no se encontraron resultados que favorezcan este planteamiento. Entonces, pareciera que la formulación teórica del algoritmo de Lucas y Kanade [31] juega un papel relevante al definir la misma región para la selección como para la correlación de rasgos “buenos”.

4.2.3. Experimento VI: Desempeño final de la estabilización

Este experimento busca ofrecer pruebas que demuestren la efectividad del proceso de estabilización implementado. Para ello se consideró una de las secuencias donde se evidencian vibraciones del video “T-Rex 600 Heli Cam”, obtenido de *YouTube.com*¹⁶. También se consideró una secuencia capturada desde ChocoLate en la cual, aunque no se evidencian tantos movimientos impulsivos, se aprecia fácilmente la presencia de ruido.

En este caso se utiliza sólo W de 7×7 píxeles, dado los resultados del experimento anterior. Se establecieron como máximo 100 rasgos “buenos” para la estimación del flujo óptico y 4 iteraciones para el refinamiento del modelo. Al igual que antes, se opta por utilizar ε según los mejores valores encontrados para las 4 combinaciones de modelo de transformación y método de estimación implementadas, tal como se detalla en el Cuadro 57 del Apéndice M.3.

Con el objetivo de poder apreciar la influencia del tamaño de la vecindad de transformaciones utilizada, se llevaron a cabo las pruebas con grupos de 7 y 15 transformaciones. Siguiendo la recomendación de Matsushita et ál. [33], si v representa el tamaño de la vecindad considerada y lo podemos expresar como $v = 2t + 1$, entonces la desviación estándar de la gausiana utilizada en cada caso está dada por \sqrt{t} .

Para evaluar si suprimir transformaciones que probablemente poco explican el movimiento de la cámara ofrece mayores ventajas, se procesaron las secuencias sin este paso adicional, así como llevando a cabo el procedimiento explicando en la sección 3.6. Se decidió considerar que una transformación debe suprimirse cuando los escalamientos horizontales y verticales que de ella se derivan

¹⁶En el Apéndice K se ofrecen más detalles sobre el video considerado para este experimento.

se diferencian por más de 0.05, o cuando alguno de los escalamientos es mayor a 1.05 o menor a 0.95. En el caso límite en que ambos escalamientos son de 0.95, produce una reducción en el tamaño de la imagen de cerca del 10 %.

Resultados

Durante la ejecución del programa que estabiliza las imágenes se puede apreciar que para las dos vecindades de transformaciones consideradas, en la secuencia se reducen de manera satisfactoria muchos de los movimientos que perturban visualmente. Esto se percibe para todas las combinaciones de modelo de transformación y método de estimación implementadas, tal como puede observarse en las figuras que indican el movimiento acumulado en el tiempo en el Apéndice M.3. En estas gráficas, adicionalmente, también se observa que los estimados no son los mismos para diferentes modelos, lo cual evidencia que un error en cualquier momento complica fácilmente la comparación de las últimas imágenes capturadas con la primera, que algunos autores usan de referencia[49, 34].

En éste mismo Apéndice también se puede apreciar que al aumentar la cardinalidad de la vecindad antes mencionada, se obtiene una estimación más suave del movimiento intencional.

La aplicación de estabilización siempre anula las transformaciones que representan reflexión. Como una decisión durante su implementación, también se considera la posibilidad de descartar otras transformaciones que probablemente tampoco explican el movimiento global, tal como se detalla en la sección 3.6. En el Apéndice M.3 se puede observar la compensación aplicada al no filtrar y al llevar a cabo este paso adicional para grupos de imágenes de los videos procesados. Claramente, las compensaciones no son iguales en ambos casos.

Al detallar las imágenes compensadas del video extraído de *YouTube.com*, muchas veces el filtrar perturba el proceso de estabilización dada la presencia de vibraciones de gran magnitud. En estos casos, tal como se evidencia en las imágenes 23 y 24 del Apéndice M.3, al utilizar la transformación anterior no se predice correctamente la siguiente. Entonces, los parámetros que determinan si una transformación debe ser filtrada o no, parecieran no ser los indicados. Éstos clasifican erróneamente las transformaciones que indican un movimiento significativo como aquellas que poco representan el movimiento global percibido producto del movimiento propio de la cámara.

Utilizando estos mismos valores, en la secuencia capturada desde el helicóptero del GIA, sí se mejora efectivamente la calidad del video. La presencia de ruido degradado de manera importante las

imágenes y muchas veces al estabilizarlas, sin filtrar las transformaciones, se observan saltos en su compensación. Más aún, en estos casos se suelen percibir grandes regiones sin información.

En el Apéndice M.3.2 se muestra un mismo grupo de imágenes estabilizadas según diferentes modelos y métodos de estimación. La utilización de una transformación bilineal genera los peores resultados visuales al momento de reducir vibraciones sin suprimir transformaciones que, posiblemente, poco reflejan el movimiento de la cámara. Podría suponerse que este problema se deriva de la simplificación de la transformación en un modelo “rígido”, para llevar a cabo la reducción de la componente no intencional del movimiento global que se percibe. Sin embargo, observando las Figuras 41 y 42 del Apéndice M.3, podemos notar que el problema en las imágenes consideradas se fundamenta en la estimación de la transformación que modela el movimiento global. Adicionalmente, considerando que en el Experimento IV, sección 4.2.1, para el grupo de secuencias capturadas desde ChocoLate esta transformación evidenció una disminución en la ganancia promedio que reporta, su uso parece poco ventajoso ante la presencia significativa de ruido.

Conclusiones

En este experimento se evidencia la estimación de la componente intencional del movimiento y la posibilidad de graduar la calidad de la secuencia estabilizada según el tamaño de la vecindad de transformaciones que se considera. Mientras más grande es ésta, más suave será el estimado del movimiento intencional percibido.

Suprimiendo del proceso de estabilización las transformaciones que posiblemente poco explican el movimiento global, ante la presencia de ruido significativo en las imágenes, se evidencia una mejora importante en la fase de compensación de una secuencia. Utilizando estimados previos del movimiento, se busca predecir los futuros ante la estimación de transformaciones que probablemente degradan la calidad del video final. Si bien esta solución no es perfecta, puesto que la predicción puede ser errónea, visualmente suele mejorarse el resultado al procesar imágenes capturadas desde ChocoLate. El fracaso de este procedimiento en el video de *YouTube.com* puede atribuirse a una selección poco conveniente de los parámetros que permiten clasificar la transformaciones, en función de las propiedades de la secuencia con la cual se realizaron las pruebas.

Una característica relevante del modelo bilineal pudo observarse en los resultados obtenidos. A partir de las imágenes estudiadas, pareciera que éste tipo de transformación es más susceptible que

los otras al ruido.

Es importante aclarar que los valores considerados para clasificar una transformación como útil o no para la estabilización, sólo sirven para evidenciar su efecto en los videos procesados en este experimento. Éstos deben ajustarse según las necesidades del usuario.

Capítulo 5

Conclusiones y Recomendaciones

En este capítulo se describen los aportes del presente trabajo, así como también se resumen sus conclusiones más importantes. En este sentido, vale mencionar el ensamblaje de un helicóptero a control remoto para llevar a cabo las pruebas, la implementación de la aplicación para la estabilización de imágenes, propuesta como objetivo principal del proyecto, y los logros alcanzados durante su desarrollo. A continuación, también se plantean algunas ideas futuras que resultan interesantes según el caso:

- Con el fin de poder llevar a cabo pruebas en vivo, se ensambló un helicóptero a control remoto que ahora forma parte de los robots con los que cuenta el Grupo de Inteligencia Artificial. Adicionalmente, se le adaptó una pequeña cámara que transmite imágenes inalámbricamente, convirtiéndolo en uno de los pocos equipos de este estilo destinado a la investigación en el país. El vehículo aéreo ahora puede ser usado para realizar otros trabajos relacionados, por ejemplo, con el área de visión. Tal como sucedió para cumplir las metas de este proyecto, sería necesario aprender a volarlo. Sin embargo, con él podrían abarcarse problemas que exigen o se favorecen de la utilización de una plataforma aérea.
- Se implementó exitosamente una aplicación para la estabilización en tiempo real de imágenes capturadas desde un vehículo aéreo. Si bien su funcionalidad depende de diversas librerías, con ella se logró integrarlas para crear, finalmente, un programa extensible para el procesamiento de imágenes. Utilizando, por ejemplo, la estimación del movimiento global que se lleva a cabo, se puede agregar un detector de movimientos independientes de cuerpos u objetos visibles en el video que se estabiliza. También con este resultado intermedio se puede llevar a cabo un proceso de composición de mosaicos, tal que se completan los vacíos presentes en la imagen estabilizada o se crea un mapa del espacio percibido en las escenas. La información derivada de las imágenes compensadas también puede aprovecharse para orientar al robot que las captura.

Considerando la revisión bibliográfica llevada a cabo como fase inicial del proyecto, pocas veces se encuentran trabajos que implementan sistemas de estabilización digital de video

online y reportan la velocidad de procesamiento obtenida. La implementación más rápida de la cual se tiene referencia es la descrita por Broggi et ál. [9]. Con ella llegan a procesar 4100 imágenes en 4 segundos, aunque su aplicabilidad se restringe a la estabilización vertical de secuencias capturadas desde un automóvil. El método utilizado supone la existencia de estructuras horizontales de las cuales se puede estimar la traslación a lo largo del eje “y”.

En cuanto a los sistemas de estabilización que consideran tanto rotación como desplazamiento, no se conoce de ninguno que supere en tiempo al de Chen y Lovell [11]. Con éste se logran procesar hasta 17 imágenes, de 320×240 píxeles, por segundo. Sin embargo, utilizando estas mismas dimensiones para el video, con la aplicación implementada se llegó a registrar, en promedio, la estabilización efectiva en un rango que va, aproximadamente, de 20 a 28 imágenes por segundo, utilizando la mejor configuración encontrada durante la fase experimental para los videos procesados. Estos resultados se obtuvieron sin llevar a cabo la estabilización en función de *hardware* especializado y, con ellos, se pudo apreciar una mejora en la calidad de los videos. Esto nos evidencia que la aplicación efectivamente puede correr en tiempo real, según los valores que poseen sus parámetros más importantes.

Naturalmente, la calidad de las secuencias estabilizadas es una medida cualitativa. De la fase experimental puede concluirse que la efectividad de la aplicación depende en gran medida de su configuración. Entonces, con la intención de poder ajustarse a las necesidades del usuario, ésta cuenta con opciones para modificar el valor de sus parámetros más relevantes.

Sería interesante ponerla a prueba bajo otras condiciones diferentes a las consideradas para la captura de imágenes aéreas. Si un cuerpo que está muy cerca de la cámara se mueve, podemos predecir que el estimado del movimiento global, generalmente, poco reflejará el del dispositivo de captura. Esto se debe a que el movimiento dominante que se percibiría entre imágenes sería el del cuerpo antes mencionado.

Este problema afecta por igual a cualquier sistema de estabilización por *software* que se conoce. Si no se cuenta con otros sensores que permitan corroborar el estimado de movimiento global, posiblemente se obtendrán resultados desfavorables. En los casos en los cuales esto no sucede, sería interesante evaluar si al ajustar la aplicación se logra estabilizar video en otras circunstancias de manera satisfactoria. La facilidad que ésta ofrece para cambiar el modelo que

busca reflejar el movimiento global, en teoría, le permitirían adaptarse a diferentes situaciones.

- A partir de las pruebas realizadas sobre la implementación del algoritmo de Lucas y Kanade [31] en OpenCV[8], se describe la influencia de varios de sus parámetros más importantes. Con ello se busca orientar sobre el uso de estas variables, puesto que casi no se tiene referencia en otros trabajos. Es importante mencionar que algunos autores, como Barron et ál. [5], reportan la efectividad de alguna versión de este algoritmo, en comparación a otros métodos, para llevar a cabo una estimación del campo de movimiento que se percibe entre dos imágenes. Considerando que la versión de OpenCV es de fácil acceso y suele utilizarse con diferentes fines, tal como lo demuestran [27, 30, 33], las pruebas que se llevaron a cabo sobre su desempeño conforman un aporte relevante. De ellas puede observarse su funcionamiento en un poco más de 31000 pares de imágenes con diferentes características, tanto en su contenido como en la cantidad de ruido que presentan y el movimiento que entre ellas se percibe.

No es común encontrar reportes de pruebas ampliamente realizadas sobre las fases que componen el proceso de estabilización. Generalmente, los autores no indican cómo se llevó a cabo la selección de los parámetros que determinan el desempeño de su implementación y muestran resultados sobre una reducida cantidad de imágenes. Sólo se conocen dos excepciones en este aspecto. Por un lado, al estabilizar videos verticalmente en [9] se consideran 7 secuencias en la fase experimental, que se traducen en un poco mas de 15000 imágenes.

Más acorde con los objetivos de este proyecto, es el trabajo de Matsushita et ál. [33], donde se presenta un sistema de estabilización *offline* que incluye una fase de completación de las imágenes compensadas. Con el uso de una transformación afín para modelar el movimiento global que se percibe en una secuencia, en éste se llevan a cabo pruebas en 80 minutos de video, que equivalen a 144000 imágenes. Sin embargo, si bien utiliza la misma implementación del algoritmo de estimación de flujo óptico en la que se basa la aplicación creada para estabilizar imágenes desde un vehículo aéreo, no se ofrecen detalles sobre los valores considerados para ajustar el algoritmo de Lucas y Kanade [31].

Como continuación a las pruebas realizadas, sería interesante llevar a cabo experimentos sobre secuencias en las que se conozca el movimiento que describe la cámara al capturarlas. Como un avance en este aspecto, Baker et ál. [4] han logrado crear secuencias específicas para evaluar

inicialmente varios procedimientos para el cálculo de flujo óptico, incluyendo el utilizado en este trabajo. Sin embargo, de nuevo no se ofrecen detalles sobre la configuración utilizada y el por qué de ésta. Si bien los videos no presentan imágenes aéreas ni permiten llevar a cabo un estudio exhaustivo, tal como ellos mismos lo indican, realizar más pruebas con datos certeros sobre los vectores de movimiento sería provechoso.

- Como una particularidad de la aplicación implementada, ésta permite utilizar diferentes modelos de transformación para estimar el movimiento global que se percibe entre imágenes. Lo más común es que se decida previamente a la implementación el tipo de modelo a utilizar, sin ofrecer pruebas de su rendimiento en función a otras opciones. El único caso que se conoce en el que se lleva a cabo la compensación de imágenes y se describen varios modelos que fueron evaluados, es el trabajo de Jung y Sukhatme [28], quienes buscan detectar movimiento en tiempo real desde cámaras en diferentes robots. No se ofrece información sobre cómo se llevó a cabo la selección entre las transformaciones, pero se indica que un modelo bilinal, en comparación a uno afín y otro pseudo-perspectivo, resulta ser el más apropiado para llevar a cabo la estimación del movimiento procesando hasta 10 imágenes por segundo.

Con la necesidad de obtener pruebas que evidenciaran la efectividad de los modelos considerados en este proyecto, se llevaron a cabo experimentos que buscan determinar cuál resulta más ventajoso para la estabilización de imágenes aéreas. Tanto uno afín, uno similar o uno bilineal, resultaron útiles para llevar a cabo la compensación en este proceso. Comparándolos mutuamente, no se puede determinar a ciencia cierta cuál estima mejor el movimiento global percibido en los más de 3500 pares de imágenes estudiados, pero sí se observa que ante el procesamiento de imágenes con relativamente poco ruido, un modelo similar parece ofrecer la menor “ganancia” en promedio. Esta medida hace referencia a la reducción de la diferencia de intensidad entre las imágenes de un par procesado después de la compensación total del movimiento que en él se percibe, en función de su diferencia original.

Siguiendo la mayoría de las implementaciones que se conocen sobre sistemas para la estabilización de imágenes, se utilizó como método de estimación, principalmente, mínimos cuadrados iterativos. Se demostró empíricamente que el refinamiento de la transformación puede ser ventajoso para restarle influencia a los vectores de flujo óptico que poco reflejan el movimien-

to global. Esto puede traducirse en una mejor estimación de la transformación que modela el movimiento.

Adicionalmente, por medio de una descomposición del modelo afín, se estudiaron los resultados obtenidos al realizar la estimación por medio de una regresión de mínimos cuadrados totales. No se tiene referencia de la utilización de este método para llevar a cabo la estimación de la transformación que modela el movimiento global. Si bien no se encontraron ventajas significativas que nos hagan preferirlo ante el método clásico, las pruebas llevadas a cabo nos permiten concluir que puede ser usado de manera intuitiva en este proceso.

- El método de estabilización implementado es simple y con él se evidenció una mejora satisfactoria en la calidad de los videos utilizados durante la fase experimental de este proyecto. Su mayor desventaja es el retraso en el video estabilizado que se muestra finalmente en la aplicación. Esta característica la poseen también otros sistemas que consideran un conjunto de las transformaciones encontradas como información del “futuro”, tal que resultan útiles para estimar la componente intencional del movimiento que se percibe [20, 36, 11, 33, 26]. Afortunadamente, este retardo puede adaptarse a las necesidades del usuario y demorando la reproducción del video en apenas 3 imágenes se suele apreciar una estabilización efectiva.

En la aplicación se utiliza una función gausiana que, al variar su desviación estándar, permite ajustar el peso que se le asigna a cada una las transformaciones consideradas para encontrar esta componente de movimiento. Matsushita et ál. [33] también utilizan este tipo de función pero no describen claramente cómo se lleva a cabo el proceso. En la fase de estabilización de su sistema se hace referencia al desplazamiento, sin embargo, no se menciona la forma de trabajo empleada para suavizar el movimiento que representa la matriz de transformación geométrica de 2×2 que incluye el modelo afín utilizado. Por lo tanto, pareciera que la estabilización se lleva a cabo considerando únicamente las traslaciones horizontales y verticales relativas, mientras que las transformaciones completas se utilizan para llenar la imagen estabilizada con información de las vecinas. Este proceso se conoce como *motion inpainting* y busca mejorar aún más la calidad del video final.

El método implementado, por el contrario, puede ajustarse a los diferentes modelos propuestos y de ellos deriva las componentes que describen la rotación y las traslaciones estimadas. No

se conoce de otro sistema para la estabilización que posea esta cualidad, a pesar que varios finalmente compensan bajo un modelo “rígido”[20, 11, 49, 26], tal como es el caso de la aplicación producto de este proyecto.

Una estimación errónea del campo de movimiento conlleva a la aparición de artefactos visuales en la secuencia estabilizada, tal como lo mencionan Litvin et ál. [29]. Buscando solventar este problema, en [49] se propone la utilización de un filtro de partículas para estimar el movimiento global percibido entre las imágenes. En este procedimiento resulta necesario comparar la primera imagen de la secuencia, que sirve como referencia, con la actual que se estabiliza. Entonces, cabe preguntarse qué sucede cuando estas ya no guardan suficiente relación para que su comparación sea efectiva.

La descomposición de la transformación antes mencionada, le permite a la aplicación detectar transformaciones que posiblemente poco reflejan el movimiento de la cámara en 3D. Por medio de este paso adicional, se logró mejorar la calidad de los videos en situaciones donde los vectores de flujo óptico difieren de este movimiento ampliamente por la presencia de ruido. Esto se observa comúnmente en las imágenes transmitidas inalámbricamente desde el helicóptero ensamblado. Cuando este paso no se lleva a cabo, principalmente al modelar el movimiento bajo una transformación bilineal, se suelen percibir saltos en la secuencia estabilizada, así como regiones amplias sin contenido. Entonces, aún cuando Yang et ál. no reportan tiempos de procesamiento, es de interés comparar su resultado con la metodología implementada en este proyecto. Así mismo, sería interesante llevar a cabo pruebas con otros sistemas para poder conocer su nivel de robustez bajo condiciones equivalentes.

Finalmente, es importante mencionar que aún cuando se logró estabilizar imágenes capturadas desde un vehículo aéreo en tiempo real, tal como sucede con otros métodos, la efectividad del procedimiento utilizado depende del video que se procesa y de los diferentes parámetros que influencian las fases que lo componen. A pesar de ésto, con los experimentos llevados a cabo se busca orientar sobre la selección de valores al probar su efectividad sobre diferentes secuencias de imágenes. Quedando todavía muchos aspectos a mejorar en el proceso de estabilización, en diversos problemas los resultados obtenidos pueden ser aprovechados para facilitar su resolución.

Bibliografía

- [1] ALIGN Corporation Limited (2004). ALIGNRC. <http://www.align.com.tw/>.
- [2] Anton, H. (1987). *Elementary Linear Algebra*. John Wiley & Sons Inc, 5th edition.
- [3] Apple Inc. (2008). QuickTime. <http://www.apple.com/quicktime/>.
- [4] Baker, S., Roth, S., Scharstein, D., Black, M., Lewis, J., y Szeliski, R. (2007). A database and evaluation methodology for optical flow. pages 1–8.
- [5] Barron, J. L., Fleet, D. J., y Beauchemin, S. S. (1994). Performance of optical flow techniques. *Int. J. Comput. Vision*, 12(1):43–77.
- [6] Beauchemin, S. S. y Barron, J. L. (1995). The computation of optical flow. *ACM Comput. Surv.*, 27(3):433–466.
- [7] Bell, W., Felzenszwalb, P., y Huttenlocher, D. (1999). Detection and long term tracking of moving objects in aerial video. Disponible en: <http://www.cs.cornell.edu/vision/wbell/identtracker/>.
- [8] Bouget, J.-Y. (1999). Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the algorithm. Technical report, Microprocessor Research Labs, Intel Corporation.
- [9] Broggi, A., Grisleri, P., Graf, T., y Meinecke, M. (2005). A software video stabilization system for automotive oriented applications. *61st IEEE Vehicular Technology Conference (VTC)*, 5:2760–2764.
- [10] Chang, H.-C., Lai, S.-H., y Lu, K.-R. (2004). A robust and efficient video stabilization algorithm. En *IEEE International Conference on Multimedia and Expo 2004 (ICME'04)*, volume 1, pages 29–32.
- [11] Chen, S. y Lovell, B. C. (2001). Real-Time MMX-Accelerated Image Stabilization System. En *Proceedings of Image and Vision Computing '01 New Zealand*, pages 163–168.
- [12] de Groen, P. (1998). An Introduction to Total Least Squares. *ArXiv Mathematics e-prints*.
- [13] Ding, W., Gong, Z., Xie, S., y Zou, H. (2006). Real-time vision-based object tracking from a moving platform in the air. En *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 681–685. Peking: IEEE. ISBN: 1-4244-0259-X. Disponible en: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4058437.
- [14] Duric, Z. y Rosenfeld, A. (1996). Image sequence stabilization in real time. *Real-Time Imaging*, 2(5):271–284.
- [15] Estalayo, E., Salgado, L., Jaureguizar, F., y García, N. (2006). Efficient image stabilization and automatic target detection in aerial FLIR sequences. En *Automatic Target Recognition XVI. Edited by Sadjadi, Firooz A.. Proceedings of the SPIE, Volume 6234, pp. 62340N (2006)*, volume 6234 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*.

- [16] Fischler, M. A. y Bolles, R. C. (1987). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. pages 726–740.
- [17] Forsyth, D. A. y Ponce, J. (2003). *Computer Vision: A Modern Approach*. Prentice Hall Series in Artificial Intelligence. Pearson Education.
- [18] Free Software Foundation (2008). GNU Scientific Library. <http://www.gnu.org/software/gsl/>.
- [19] Golub, G. H. y Loan, C. V. (1980). An analysis of the total least squares problem. Technical report, Cornell University, Ithaca, NY, USA.
- [20] Guestrin, C., Cozman, F., y Krotkov, E. (1997). Image stabilization for feature tracking and generation of stable video overlays. Technical Report CMU-RI-TR-97-42, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.
- [21] Guestrin, C., Cozman, F., y Krotkov, E. (1998). Fast software image stabilization with color registration. En *Proceedings of IROS 98*, volume 1, pages 19 – 24.
- [22] Hodge, V. y Austin, J. (2004). A survey of outlier detection methodologies. *Artif. Intell. Rev.*, 22(2):85–126.
- [23] Horn, B. y Schunck, B. (1981). Determining optical flow. *AI*, 17(1-3):185–203.
- [24] Intel Corporation (2008). Open Source Computer Vision Library (OpenCV). <http://opencvlibrary.sourceforge.net/>.
- [25] Irani, M., Rousso, B., y Peleg, S. (1994). Computing occluding and transparent motions. *Int. J. Comput. Vision*, 12(1):5–16.
- [26] Johansen, D. L. (2006). Video stabilization and target localization using feature tracking with small UAV video. Master's thesis, Brigham Young University.
- [27] Jung, B. y Sukhatme, G. S. (2004). Detecting moving objects using a single camera on a mobile robot in an outdoor environment. En *International Conference on Intelligent Autonomous Systems, Amsterdam, The Netherlands*, pages 980–987.
- [28] Jung, B. y Sukhatme, G. S. (2005). Real-time motion tracking from a mobile robot. Cres-05-008, Center for Robotics and Embedded Systems, University of Southern California. Disponible en: http://cres.usc.edu/cgi-bin/print_pub_details.pl?pubid=464.
- [29] Litvin, A., Konrad, J., y Karl, W. C. (2003). Probabilistic video stabilization using Kalman filtering and mosaicing. En Vasudev, B., Hsing, T. R., Tescher, A. G., y Ebrahimi, T., editores, *Image and Video Communications and Processing 2003. Edited by Vasudev, Bhaskaran; Hsing, T. Russell; Tescher, Andrew G.; Ebrahimi, Touradj. Proceedings of the SPIE, Volume 5022, pp. 663-674 (2003)*, volume 5022 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 663–674.

- [30] Lookingbill, A., Lieb, D., Stavens, D., y Thrun, S. (2005). Learning activity-based ground models from a moving helicopter platform. En *ICRA'05*, pages 3948–3953.
- [31] Lucas, B. y Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. En *IJCAI81*, pages 674–679. Disponible en: citeseer.ist.psu.edu/lucas81iterative.html.
- [32] Marcenaro, L., Vernazza, G., y Regazzoni, C. (2001). Image stabilization algorithms for video-surveillance applications. En *ICIP01*, pages I: 349–352.
- [33] Matsushita, Y., Ofek, E., Tang, X., y Shum, H.-Y. (2005). Full-frame video stabilization. En *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 50–57, Washington, DC, USA. IEEE Computer Society.
- [34] Morimoto, C. y Chellappa, R. (1996a). Automatic digital image stabilization. En *IEEE International Conference on Pattern Recognition*.
- [35] Morimoto, C. y Chellappa, R. (1996b). Fast electronic digital image stabilization. *13th International Conference on Pattern Recognition (ICPR'96)*, 3:284.
- [36] Ratakonda, K. (1998). Real-time digital video stabilization for multi-media applications. En *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (ISCAS'98)*, volume 4, pages 69–72.
- [37] Shi, J. y Tomasi, C. (1993). Good features to track. Technical report, Cornell University, Ithaca, NY, USA.
- [38] Shi, J. y Tomasi, C. (1994). Good features to track. En *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle. Disponible en: <http://citeseer.ist.psu.edu/shi94good.html>.
- [39] Shoemake, K. (1994). Polar matrix decomposition. *Graphics Gems IV*, pages 207–221.
- [40] Shoemake, K. y Duff, T. (1992). Matrix animation and polar decomposition. En *Proc. of the Graphics Interface '92*, pages 258–264, Vancouver, Canada.
- [41] Stockman, G. y Shapiro, L. G. (2001). *Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA.
- [42] Tian, T. Y., Tomasi, C., y Heeger, D. J. (1996). Comparison of approaches to egomotion computation. En *CVPR '96: Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*, page 315, Washington, DC, USA. IEEE Computer Society.
- [43] Tico, M. y Vehvilainen, M. (2005). Constraint motion filtering for video stabilization. *IEEE International Conference on Image Processing (ICIP 2005)*, 3:569–572.

- [44] Tomasi, C. y Kanade, T. (1991). Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University. Disponible en: <http://citeseer.ist.psu.edu/tomasi91detection.html>.
- [45] Torr, P. H. y Murray, D. W. (1993). Outlier detection and motion segmentation. En Schenker, P. S., editor, *Proc. SPIE Vol. 2059, p. 432-443, Sensor Fusion VI, Paul S. Schenker; Ed.*, volume 2059 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 432–443.
- [46] Trolltech (2008). Qt. <http://trolltech.com/products/qt/>.
- [47] Urdan, T. (2005). *Statistics in Plain English*, volume 2. Lawrence Erlbaum Associates, Inc.
- [48] Verri, R. y Poggio, T. (1989). Motion field and optical flow: qualitative properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:490–498.
- [49] Yang, J., Schonfeld, D., Chen, C., y Mohamed, M. (2006). Online video stabilization based on particle filters. pages 1545–1548.
- [50] Zitova, B. y Flusser, J. (2003). Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000. Disponible en: [http://dx.doi.org/10.1016/S0262-8856\(03\)00137-9](http://dx.doi.org/10.1016/S0262-8856(03)00137-9).

Apéndice A

Definiciones de interés

Espacio vectorial. Conjunto cerrado bajo la sumatoria de vectores y la multiplicación de éstos por un escalar. El espacio Euclídeo \mathbb{R}^n , por ejemplo, es un espacio vectorial de n dimensiones.

Norma l_2 para un vector. Sea \vec{v} un vector cualquiera en \mathbb{R}^n . Se define $|\vec{v}|$ como su módulo o norma l_2 tal que

$$|\vec{v}| = \sqrt{\sum_{k=1}^n v_k^2} \quad (38)$$

Vector unitario. Sea \vec{v} un vector cualquiera en \mathbb{R}^n . Se define \hat{v} como el vector unitario en su dirección tal que

$$\hat{v} = \frac{\vec{v}}{|\vec{v}|} \quad (39)$$

Gradiente. Para una función cualquiera $z : \mathbb{R}^n \rightarrow \mathbb{R}$, se define ∇z como el gradiente de z tal que

$$\nabla z = \left(\frac{\partial z}{\partial z_1}, \dots, \frac{\partial z}{\partial z_n} \right) \quad (40)$$

Producto escalar. En el espacio \mathbb{R}^n , se define el operador producto escalar de dos vectores cualesquiera \vec{a} y \vec{b} de la siguiente manera

$$\vec{a} \cdot \vec{b} = (a_1, \dots, a_n) \cdot (b_1, \dots, b_n) = \sum_{i=1}^n a_i b_i \quad (41)$$

Expansión de Taylor de una función de dos variables. Sea f una función real de dos variables, entonces su expansión de Taylor está dada por

$$\begin{aligned} f(x + \Delta_x, y + \Delta_y) &= f(x, y) + \left[\frac{\partial f}{\partial x}(x, y) \Delta_x + \frac{\partial f}{\partial y}(x, y) \Delta_y \right] + \\ &\quad \frac{1}{2!} \left[\Delta_x^2 \frac{\partial^2 f}{\partial x^2}(x, y) + 2\Delta_x \Delta_y \frac{\partial^2 f}{\partial x \partial y}(x, y) + \Delta_y^2 \frac{\partial^2 f}{\partial y^2}(x, y) \right] + \dots \end{aligned}$$

Transformación lineal. Una transformación lineal entre dos espacios vectoriales U y V es una función $T : U \rightarrow V$, tal que

$$1. \quad T(\vec{v}_1 + \vec{v}_2) = T(\vec{v}_1) + T(\vec{v}_2)$$

$$2. \quad T(\alpha\vec{v}) = \alpha T(\vec{v})$$

Transformación afín en \mathbb{R}^2 . Dados un vector \vec{p} , una traslación \vec{q} y una transformación lineal A en \mathbb{R}^2 ; se define una transformación afín en \mathbb{R}^2 como una función $F : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ tal que

$$F(\vec{p}) = A\vec{p} + \vec{q} \quad (42)$$

F se caracteriza por preservar la colinealidad de puntos y la proporción de distancia a lo largo de una línea. En otras palabras, todos los puntos pertenecientes a una recta inicialmente lo seguirán haciendo posterior a la transformación; y el punto medio de un segmento de recta lo seguirá siendo luego de la aplicación de F .

Convolución bidimensional. La convolución de una señal discreta $f(x, y)$ en la vecindad N de $[x_i, y_j]$ con una máscara M se define por,

$$M \circ N[x_i, y_j] = \sum_{a=a_{inicial}}^{a_{final}} \sum_{b=b_{inicial}}^{b_{final}} M(a, b) f(x_i - a, y_j - b) \quad (43)$$

Llevar a cabo esta operación se conoce como filtrar la señal con la máscara, o *kernel*, M . [17]

Matriz diagonal. Matriz cuadrada cuyas entradas fuera de la diagonal principal (\searrow) son cero.

Matriz ortogonal. Matriz cuadrada cuya traspuesta es su inversa,

$$Q^T Q = Q Q^T = I \quad (44)$$

siendo I la identidad de igual tamaño que Q .

Matriz singular. Matriz cuadrada, con determinante cero, que no posee inversa.

Autovalores y autovectores de una transformación lineal. Sea A la matriz asociada a una transformación lineal. Si existe un vector $\vec{X} \in \mathbb{R}^2$, tal que $\vec{X} \neq \vec{0}$, que cumpla

$$AX = \lambda X \quad (45)$$

para algún escalar λ , entonces λ se conoce como el autovalor de A asociado al autovector derecho X . Esta ecuación suele utilizarse como la definición más general de un autovector, aunque su recíproco $XA = \lambda X$ también es válido y define, en este caso, un autovector izquierdo.

Siendo A es una matriz cuadrada, la ecuación (45) puede expresarse como $(A - \lambda I)X = 0$, con I la identidad. De esta manera, los autovalores pueden encontrarse resolviendo $\det(A - \lambda I) = 0$.

Matriz real definida positiva. Matriz cuadrada simétrica de números reales se considera definida positiva si $z^T M z > 0$ para un vector z no nulo en \mathbb{R} . Si M cumple esta propiedad, entonces todos los autovalores λ_i de ella son positivos.

Número de condición. Dada una matriz cuadrada A y una norma $\|\cdot\|$, su número de condición se define por,

$$\text{num_cond}(Z) = \begin{cases} \|A\| \|A^{-1}\|, & \text{si } A \text{ es no singular} \\ +\infty, & \text{si } A \text{ es singular} \end{cases} \quad (46)$$

El número de condición de la matriz A , asociado a un sistema $Ax = b$, es una medida de la sensibilidad de la solución del sistema a los errores presentes en los datos.

Media de una muestra. Considerando un conjunto X de n datos, su media aritmética o promedio se define matemáticamente como

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n} \quad (47)$$

Varianza de una muestra. Considerando un conjunto X de n datos, su varianza s^2 es una medida de la dispersión de la muestra. s denota la **desviación estándar**, tal que

$$s_X^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{(n - 1)} \quad (48)$$

Covarianza entre dos variables. Considerando dos conjuntos X e Y de n datos cada uno, su covarianza se calcula por

$$\text{Cov}(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{(n - 1)} \quad (49)$$

La covarianza entre X e Y nos ofrece una medida de cuánto varian en conjunto las dos variables. Si es positiva, entonces X e Y crecen en conjunto. Si es negativa, mientras una crece la otra decrece. Si su valor es 0, no se percibe una relación lineal entre las mismas.

Coeficiente de correlación de Pearson. Un coeficiente de correlación nos ofrece una medida, en promedio, de cuánto se asocian los valores entre dos variables. Dadas dos muestras representadas por medio de las variables X e Y , el coeficiente de correlación de Pearson, según [47], se define como

$$r(X, Y) = \frac{Cov(X, Y)}{s_X s_Y} \quad (50)$$

Error estándar de la media (S.E.M.). Un estimador del error que se espera en el valor de la media de una población a partir del estimado de su media en una muestra X , puede calcularse como

$$SE_{\bar{X}} = \frac{s_X}{\sqrt{n}} \quad (51)$$

siendo n el tamaño de X y s_X su desviación estándar (la raíz de su varianza).

Descomposición espectral de una matriz simétrica. Sea A una matriz cuadrada, de $n \times n$, y simétrica, con n autovectores linealmente independientes. A se puede factorizar como

$$A = Q\Lambda Q^T \quad (52)$$

donde Q es una matriz ortogonal y Λ es diagonal. Adicionalmente, Λ incluye los autovalores de A , tal que $\Lambda_{ii} = \lambda_i$.

Descomposición en valores singulares de una matriz. Sea M una matriz de números reales y de dimensiones $m \times n$. Ésta se puede descomponer como

$$M = U\Sigma V^T \quad (53)$$

siendo U y V matrices ortogonales, de $m \times m$ y $n \times n$, respectivamente, y Σ una matriz de $m \times n$ con números no negativos en su diagonal y ceros fuera de ella. Σ contiene los valores singulares de M . En el caso en que éstos se ordenen de manera no creciente, M determina de manera única Σ , sin embargo no necesariamente a U o V .

Apéndice B

ChocoLate y la metodología de trabajo

ChocoLate es un helicóptero *TRex 600*, de 1.2 metros de largo, de *ALIGN Corp., Ltd.*[1], tal como se muestra en la Figura 14. Fue ensamblado como antesala a este trabajo para poder capturar desde él imágenes aéreas. Cuenta con un sistema eléctrico que incluye un giroscopio, cuyo objetivo es facilitar el manejo del helicóptero durante el vuelo. Adicionalmente, le fue adaptada una microcámara a color, de manera rígida, que transmite video a una frecuencia de 2.4 Ghz. Dado su pequeño tamaño, $4.09 \times 2.39 \times 2.39$ cm, ésta no entorpece el vuelo del vehículo. Sin embargo, las vibraciones que producen el motor del helicóptero y su movimiento suelen disminuir la calidad de los videos que desde él se registran.



Figura 14: ChocoLate

La transmisión de imágenes en tiempo real fue puesta a prueba en la Universidad, en el campo abierto vecino a la Biblioteca Central. Durante las pruebas se grabó actividad en tierra de personas, carros y otro robot, usualmente utilizado por el Grupo de Inteligencia Artificial. Con la participación de estos entes se logró que las escenas no fueran totalmente estáticas, como se puede observar en la Figura 15.



Figura 15: Imágenes capturadas desde ChocoLate.

El movimiento descrito por ChocoLate durante las pruebas realizadas usualmente es continuo o lento, desplazándose a una altura superior a 10 metros. En muchos casos se busca mantenerlo fijo en un punto en el aire. Sin embargo, de vez en cuando ésto no es posible y se evidencian cambios bruscos tanto en la dirección como en la magnitud de su velocidad.

Apéndice C

Detalles sobre la aplicación implementada

La aplicación corre bajo Mac OS X y presenta una interfaz sencilla creada utilizando *Qt*[46]. Dos ventanas principales la componen, tal como se muestran en la Figura 16. En una se muestra el video estabilizado y en otra se presenta la opción para comenzar o detener el procesamiento, así como aquella que habilita la visualización de un marco sobre el video. Adicionalmente, se indica la cantidad de imágenes que se estabilizan por segundo y el tamaño del video que se reproduce.

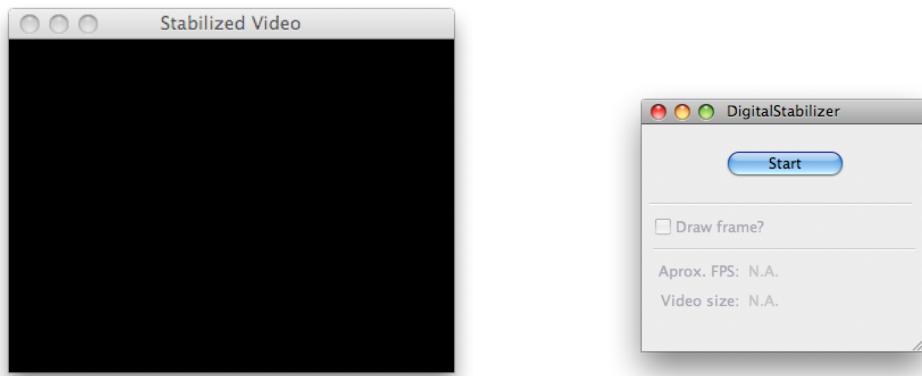


Figura 16: Ventanas que conforman constantemente la interfaz de la aplicación. A la izquierda se muestra la ventana para la visualización del video y a la derecha la que presenta las opciones principales para comenzar y detener el procesamiento.

En las preferencias generales se permite modificar los siguientes parámetros:

- Tamaño del video que se visualiza.
- Tamaño del video para su procesamiento.
- Tamaño de la vecindad de transformaciones considerada para llevar a cabo la estabilización, tal como se plantea según la ecuación (27).
- Ancho del marco que se visualiza, si se habilita la opción, sobre la imagen estabilizada.

De manera similar, se permite modificar un conjunto de variables referentes al proceso de estimación de flujo óptico:

- Máximo número de “buenos” rasgos a seleccionar.

- Porcentaje de calidad mínima sobre los rasgos seleccionados en función del máximo de los mínimos autovalores encontrado.
- Distancia mínima entre los rasgos.
- Tamaño de la vecindad W considerada para la selección y correlación de “buenos” rasgos.
- Máximo nivel de profundidad de las pirámides utilizadas para acelerar la estimación de los vectores de movimiento y aumentar la robustez del proceso.

En cuanto a los factores que influencian la estimación de la transformación que modela el movimiento global, se permite configurar las siguientes opciones:

- Tipo de modelo a utilizar y método de estimación (afín, similar o bilineal por mínimos cuadrados o afín por mínimos cuadrados totales).
- Máximo número de iteraciones a llevar a cabo para refinar el modelo.
- Cota ε que permite clasificar vectores de flujo óptico como atípicos, tal como se describe en la sección 3.2.3.
- Umbrales que permiten predecir si una transformación posiblemente representa el movimiento global percibido producto del movimiento propio de la cámara o no. Estas opciones aplican sólo si se desean filtrar las transformaciones estimadas.

Apéndice D

Estimación del gradiente

El cambio máximo de contraste en el plano de la imagen $f(x, y)$ está dado a lo largo de la dirección de la función de gradiente $\nabla f = (\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y})$, aplicando la definición que se presenta en el Apéndice A.

En la dirección horizontal, el contraste se puede aproximar por

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

que representa el cambio de intensidad a lo largo de los vecinos izquierdo y derecho del píxel en la posición $[x, y]$ dividido entre $\Delta x = 2$ unidades. Dado que puede existir ruido en la imagen y en cualquier dirección pueden aparecer bordes de objetos y regiones, resulta beneficioso promediar 3 estimados diferentes del contraste en la vecindad de $[x, y]$:

$$\frac{\partial f}{\partial x} \approx \frac{1}{3} \sum_{j=y-1}^{y+1} \frac{f(x+1, j) - f(x-1, j)}{2} \quad (54)$$

La aproximación anterior estima el contraste en la dirección horizontal promediando equitativamente el cambio de intensidad a lo largo de las filas $y-1$, y y $y+1$.

El contraste en la dirección vertical puede aproximarse de manera similar:

$$\frac{\partial f}{\partial y} \approx \frac{1}{3} \sum_{i=x-1}^{x+1} \frac{f(i, y+1) - f(i, y-1)}{2} \quad (55)$$

Las ecuaciones anteriores definen el operador *Prewitt* utilizado para detectar bordes en imágenes médicas [41]. Puede ser reescrito como una convolución (expresada por el operador \circ , según la definición (43)) a partir de las máscaras M_x y M_y ,

$$M_x = \begin{array}{|c|c|c|} \hline -1 & 0 & +1 \\ \hline -1 & 0 & +1 \\ \hline -1 & 0 & +1 \\ \hline \end{array}; \quad M_y = \begin{array}{|c|c|c|} \hline +1 & +1 & +1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array} \quad (56)$$

De esta manera, las ecuaciones (54) y (55) pueden expresarse a partir de las máscaras definidas

en (56) y la vecindad N alrededor de $[x, y]$,

$$\frac{\partial f}{\partial x} \approx \frac{1}{6}(M_x \circ N[x, y])$$

$$\frac{\partial f}{\partial y} \approx \frac{1}{6}(M_y \circ N[x, y])$$

Usualmente, la división entre 6 suele ignorarse para acelerar el cómputo, obteniendo así una aproximación escalada del gradiente. La convolución antes mencionada puede resumirse en sobreponer la máscara M en la vecindad N de $[x, y]$, tal que se lleva a cabo la sumatoria del producto de cada valor $f(i, j)$, perteneciente a $N[x, y]$, por el peso M_{ij} .

El operador *Scharr*, utilizado en la implementación piramidal del algoritmo de Lucas y Kanade [31] en OpenCV [8], se expresa de manera análoga por medio de las siguientes máscaras:

$$M_{sharr_x} = \frac{1}{32} \begin{array}{|c|c|c|} \hline & -3 & 0 & +3 \\ \hline -10 & & 0 & +10 \\ \hline & -3 & 0 & +3 \\ \hline \end{array}; \quad M_{sharr_y} = \frac{1}{32} \begin{array}{|c|c|c|} \hline & +3 & +10 & +3 \\ \hline 0 & & 0 & 0 \\ \hline & -3 & -10 & -3 \\ \hline \end{array} \quad (57)$$

Apéndice E

Ejemplos sobre la selección de rasgos

En la sección 2.2.4 se presenta de manera resumida el método propuesto por Bouguet[8] para la selección de “buenos” rasgos en el proceso de estimación del campo de movimiento entre dos imágenes. Este método se utiliza en la aplicación para la estabilización de imágenes capturadas desde un vehículo aéreo que se implementó, tal como se explica en la sección 3.2.1. A continuación se presentan ejemplos de este procedimiento que forman parte de los resultados obtenidos en el Experimento I, sección 4.1.1, para un segmento de un video de *YouTube.com*.

Una difícil situación para el proceso de estabilización se presenta cuando la cantidad de rasgos seleccionados es baja, puesto que la estimación del campo de movimiento es poco confiable. Tal es el caso, por ejemplo, del par de imágenes número 650¹ en la secuencia que se muestra en la Figura 17, correspondiente al segmento *FlyingWinter3* compuesto por 708 pares. El procedimiento para detectar rasgos primero encuentra los mínimos autovalores de la matriz Z de la ecuación (17), para cada píxel de la imagen que se desea relacionar con la de referencia, según su vecindad W . Considerando el mayor de estos valores, se establece el umbral de calidad mínima aceptada. Entonces, se elimina del proceso de selección aquellos puntos en la imagen que poseen un mínimo autovalor menor a este rango. La intensidad en las imágenes 17(c) y 17(e) representa el valor del mínimo autovalor para cada punto que sobrepasa el umbral ya mencionado. Los píxeles con valor 0, son aquellos que no llegan al mínimo nivel de calidad aceptado. Los que muestran alta intensidad representan los píxeles en la imagen original para los cuales su mínimo autovalor se acerca al máximo encontrado previamente.

Buscando eliminar de procesamientos posteriores rasgos que no ofrezcan información sobresaliente para el estimado del flujo óptico, se aplica a la imagen de mínimos autovalores antes generada, tal como en la Figura 17(c), un operador de dilatación. Así son generadas las imágenes 17(g) y 17(i).

Considerando estas imágenes de los mínimos autovalores antes y después del operador de dilatación se pueden encontrar máximos locales. En las imágenes 17(k) y 17(m) éstos se muestran

¹Dado un par de imágenes entre las cuales se desea estimar el flujo óptico, para la selección de los “buenos” rasgos no se consideran las dos que lo conforman. De hecho, en este proceso únicamente es relevante la intensidad en los píxeles de aquella que se desea aproximar a la de referencia. Esta selección de rasgos se lleva a cabo siguiendo la implementación en OpenCV de la propuesta de Bouguet[8], tal como se explica en la sección 3.2.1.

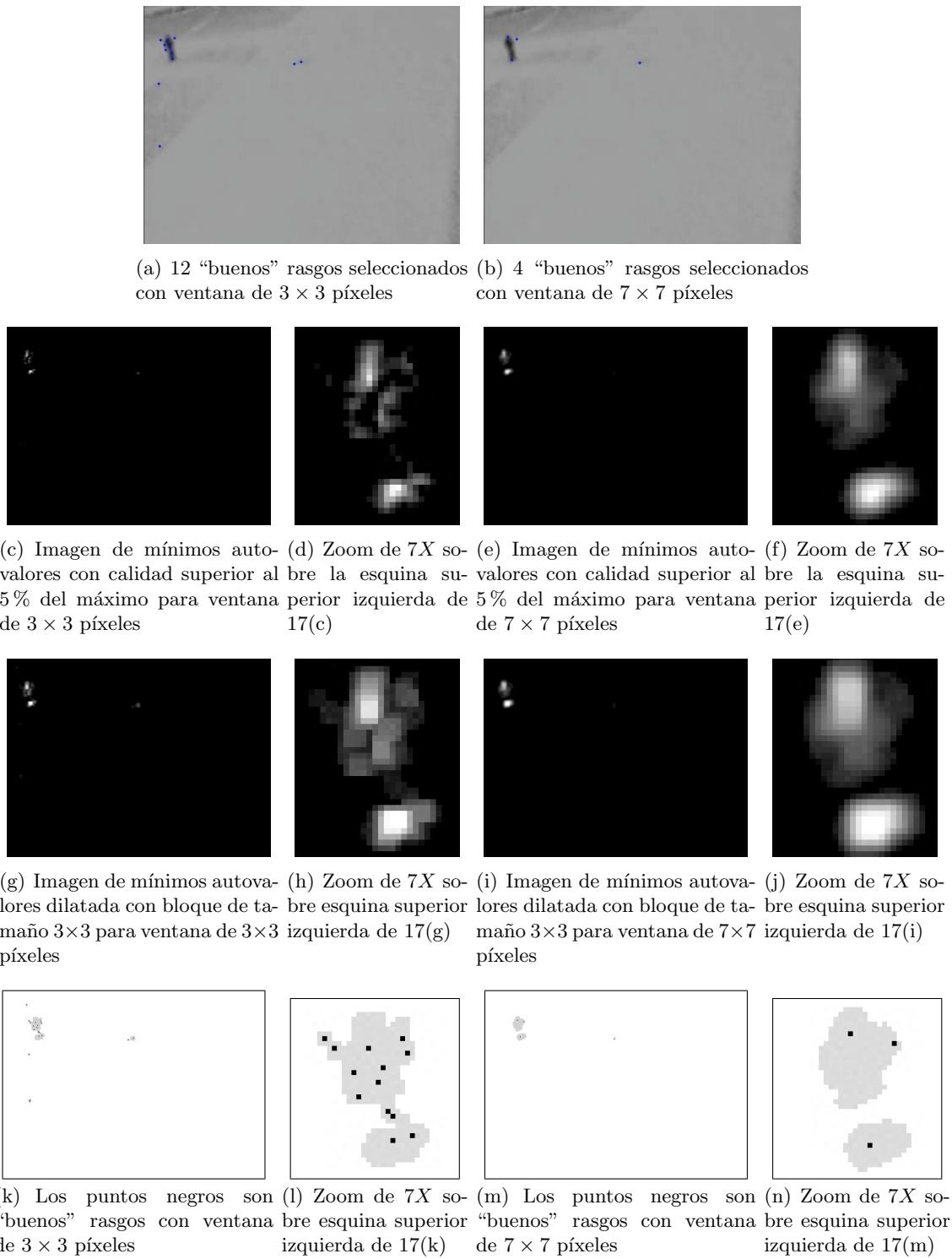


Figura 17: Proceso de selección de rasgos en par de imágenes número 650 del segmento de video *FlyingWinter3*, considerando tamaños de ventana de 3×3 y 7×7 píxeles y utilizando una pirámide de 4 niveles de profundidad máxima.

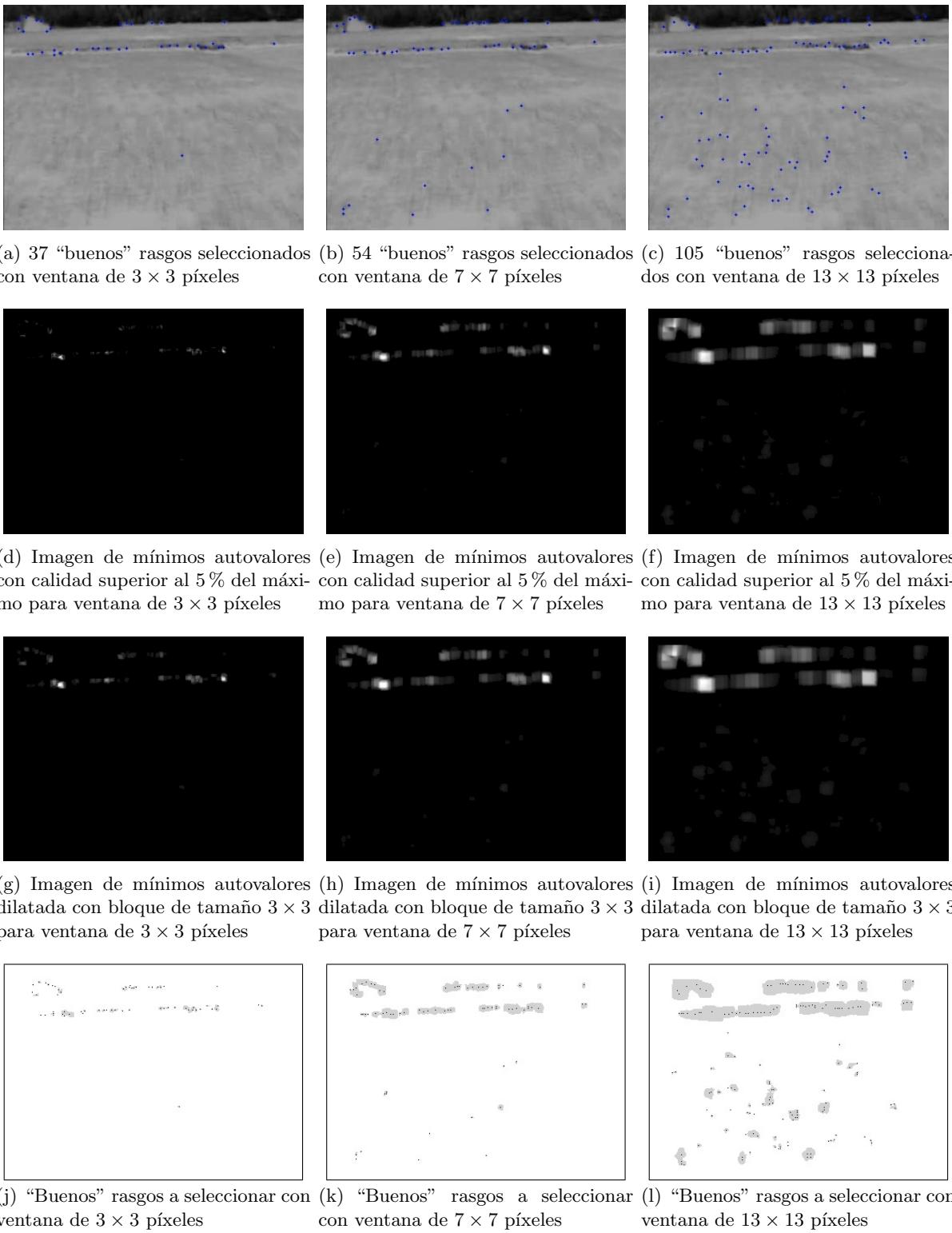


Figura 18: Proceso de selección de rasgos en par de imágenes número 223 del segmento de video *FlyingWinter3*, considerando tamaños de ventana de 3×3 , 7×7 y 13×13 píxeles y utilizando una pirámide de 4 niveles de profundidad máxima.

de color negro, siendo fácilmente detectados por ser los píxeles que en ambas imágenes poseen el mismo valor. Las regiones de color gris son aquellas para las cuales los píxeles que las conforman no cumplen esta propiedad, puesto que al aplicar el operador de dilatación no son el máximo en su vecindad de 3×3 píxeles y, por lo tanto, su valor se ve modificado en la imagen dilatada.

En las imágenes de la Figura 17, utilizando un menor tamaño de ventana, el operador de dilatación encuentra más máximos locales y selecciona más rasgos “buenos” a procesar. Sin embargo, en la Figura 18 que muestra el mismo procedimiento para el par de imágenes número 223, mientras mayor tamaño tiene W , más rasgos se seleccionan para procesamientos posteriores.

Apéndice F

Construcción las pirámides en el proceso de estimación del flujo óptico

En la aplicación de estabilización que se implementó, se utilizó el procedimiento que ofrece la librería OpenCV para calcular el flujo óptico de un conjunto de rasgos, según la propuesta piramidal de Bouguet[8] sobre el algoritmo de Lucas y Kanade[31]. En la sección 3.2.2 se describe de manera general el método y se presenta el algoritmo. A continuación se describen los aspectos más relevantes sobre la construcción de las pirámides que se emplean para mejorar la estimación del campo de movimiento.

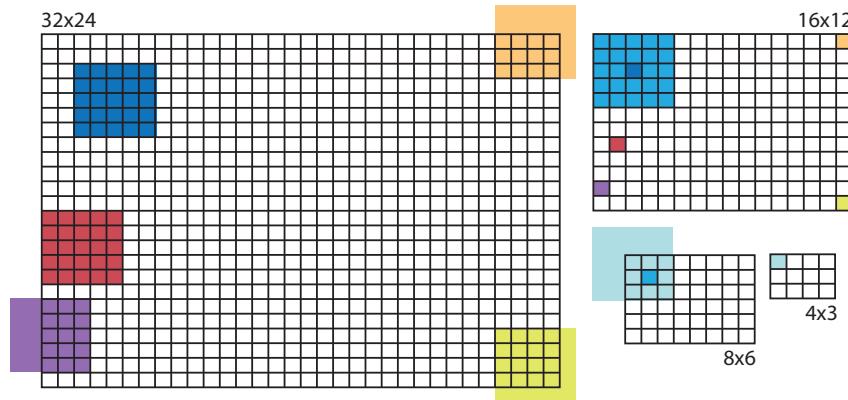


Figura 19: Pirámide de 4 niveles para una imagen inicial de 32×24 píxeles. Dado un nivel y un píxel coloreado en éste, los píxeles del mismo color en el nivel inferior identifican la vecindad de 5×5 considerada para computar el valor de su intensidad, según la implementación de Bouguet[8].

Utilizando la misma notación de la sección 3.2.2, se considera L_0 la imagen en el nivel cero de la pirámide cuyas dimensiones son las originales. Si I_{L-1} es la imagen en el nivel $L - 1$, entonces la

siguiente de menor escala puede ser definida a partir de ésta como

$$I_L(x, y) = \begin{array}{|c|c|c|c|c|} \hline & (\frac{1}{16})^2 & (\frac{1}{16})(\frac{1}{4}) & (\frac{1}{16})(\frac{3}{8}) & (\frac{1}{16})(\frac{1}{4}) & (\frac{1}{16})^2 \\ \hline (\frac{1}{16})(\frac{1}{4}) & & (\frac{1}{4})^2 & (\frac{1}{4})(\frac{3}{8}) & (\frac{1}{4})^2 & (\frac{1}{4})(\frac{1}{16}) \\ \hline (\frac{1}{16})(\frac{3}{8}) & (\frac{3}{8})(\frac{1}{4}) & & (\frac{3}{8})^2 & (\frac{3}{8})(\frac{1}{4}) & (\frac{3}{8})(\frac{1}{16}) \\ \hline (\frac{1}{16})(\frac{1}{4}) & (\frac{1}{4})^2 & (\frac{1}{4})(\frac{3}{8}) & & (\frac{1}{4})^2 & (\frac{1}{4})(\frac{1}{16}) \\ \hline (\frac{1}{16})^2 & (\frac{1}{16})(\frac{1}{4}) & (\frac{1}{16})(\frac{3}{8}) & (\frac{1}{16})(\frac{1}{4}) & & (\frac{1}{16})^2 \\ \hline \end{array} \circ N_{L-1}[2x, 2y] \quad (58)$$

donde $N_{L-1}[2x, 2y]$ representa la matriz con la vecindad de 5×5 píxeles de $(2x, 2y)$ en la imagen I_{L-1} y el operador \circ denota una convolución, tal como se define en el Apéndice A. La ecuación (58) es válida para píxeles (x, y) que cumplen $0 \leq 2x \leq (n_x^{L-1} - 1)$ y $0 \leq 2y \leq (n_y^{L-1} - 1)$. Sin embargo, para su cálculo resulta necesario el valor de otros cuya ubicación sobresale las dimensiones de I_{L-1} . Por ejemplo, para el píxel $(0, 0)$ en la imagen I_{L_3} de 4×3 píxeles que se muestra en la Figura 19, su vecindad de 5×5 incluye el $(-2, -2)$ en el nivel L_2 . En estos casos se replica la información del borde, considerando

$$I_{L-1}(-2, y) = I_{L-1}(-1, y) = I_{L-1}(0, y)$$

$$I_{L-1}(x, -2) = I_{L-1}(x, -1) = I_{L-1}(x, 0)$$

$$I_{L-1}(n_x^{L-1} + 1, y) = I_{L-1}(n_x^{L-1}, y) = I_{L-1}(n_x^{L-1} - 1, y)$$

$$I_{L-1}(x, n_y^{L-1} + 1) = I_{L-1}(x, n_y^{L-1}) = I_{L-1}(x, n_y^{L-1} - 1)$$

Las dimensiones de la imagen I_L pueden ser calculadas igualmente de manera recursiva, tal que son los enteros más grandes que cumplen

$$n_x^L \leq \frac{n_x^{L-1} + 1}{2} \quad (59)$$

$$n_y^L \leq \frac{n_y^{L-1} + 1}{2} \quad (60)$$

Apéndice G

Cálculos a nivel de subpíxeles en el proceso de estimación del flujo óptico

La aplicación para la estabilización en tiempo real de imágenes capturadas desde un helicóptero se fundamenta en el cálculo del flujo óptico, según la versión piramidal del algoritmo de Lucas y Kanade [31] presente en OpenCV[8].

Para obtener precisión en el proceso de estimación de movimiento, resulta esencial en la implementación manejar los cómputos a nivel de subpíxeles. En otras palabras, es necesario poder calcular la intensidad de las imágenes no sólo en coordenadas (x, y) enteras, sino también en aquellas ubicadas entre píxeles. Para calcular la intensidad en estos lugares, Bouguet[8] propone utilizar una interpolación bilineal. Considerando (x, y) una coordenada en una imagen, que no necesariamente corresponde con la posición de un píxel, y $(x_o, y_o) = (\lfloor x \rfloor, \lfloor y \rfloor)$ su parte entera; entonces, podemos definir su diferencia como $\alpha_x = x - x_o$ y $\alpha_y = y - y_o$. La intensidad computada en la imagen para (x, y) , utilizando una interpolación bilineal con los valores originales de los píxeles, está dada por

$$I(x, y) = (1 - \alpha_x)(1 - \alpha_y)I(x_0, y_0) + \alpha_x(1 - \alpha_y)I(x_0 + 1, y_0) + \\ (1 - \alpha_x)\alpha_yI(x_0, y_0 + 1) + \alpha_x\alpha_yI(x_0 + 1, y_0 + 1)$$

Cuando se aproximan las derivadas en las líneas 6 y 7 del algoritmo 1 en la sección 3.2.2, en la vecindad de \vec{p} , es necesario conocer los valores de los píxeles en I_L^{t-1} que pertenecen a esta región, tal que se consideran los $(x, y) \in [x - w_x, x + w_x] \times [y - w_y, y + w_y]$. Dado que las coordenadas de \vec{p} no son necesariamente enteras, la interpolación antes mencionada resulta útil. Siendo p_{x_0} y p_{y_0} las partes enteras de p_x y p_y , el parche $(x, y) \in [p_{x_0} - w_x - 1, p_{x_0} + w_x + 2] \times [p_{y_0} - w_y - 1, p_{y_0} + w_y + 2]$ es necesario para computar los valores de intensidad requeridos. Un caso similar ocurre cuando se desea computar $d(x, y)$ para obtener, finalmente, \vec{e} en la línea 11. Particularmente, se deben conocer los valores de intensidad de $I_L^t(x + g_{x_L} + v_{x_{k-1}}, y + g_{y_L} + v_{y_{k-1}})$ para todos los $(x, y) \in [x - w_x, x + w_x] \times [y - w_y, y + w_y]$.

Apéndice H

Regresión por mínimos cuadrados totales

El problema clásico de mínimos cuadrados supone que el error está confinado al vector b , tal como lo expresa la ecuación (23) y se muestra en la Figura 20. Sin embargo, ello usualmente no sucede, puesto que errores como aquellos que pueden surgir al momento de tomar la muestra de datos imposibilitan el registro correcto de los valores de la matriz A .

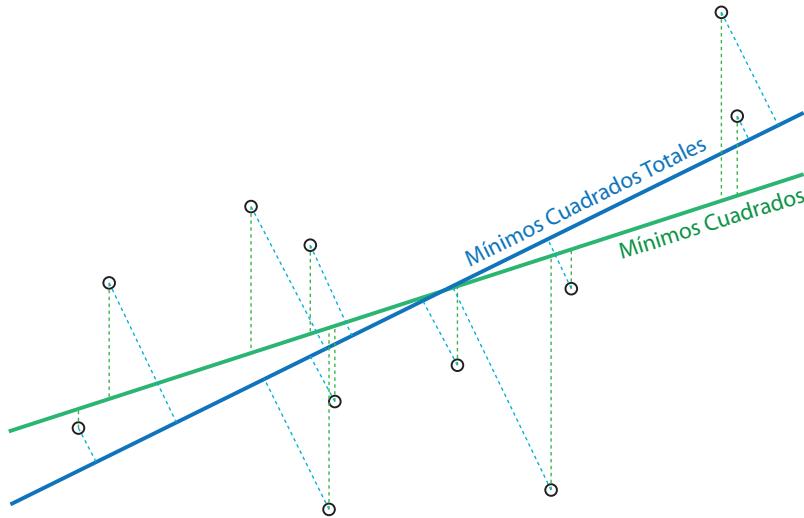


Figura 20: Regresión lineal por mínimos cuadrados y mínimos cuadrados ortogonales para un mismo conjunto de datos en 2D. Ejemplo tomado de [12].

En 2D, tal como explica P. de Groen[12], el objetivo de una regresión lineal por **mínimos cuadrados totales** es encontrar la recta l que minimiza la suma del cuadrado de las distancias “verdaderas” de m datos a ésta¹. Cuando la línea que minimiza la distancia ortogonal a los puntos es vertical, se dice que el problema no tiene solución[19].

En el caso multidimensional, se busca ajustar un hiperplano a un grupo de puntos en \mathbb{R}^m , tal que éste pasa por su centroide[12] y minimiza, de manera similar, las distancias ortogonales.

Una forma intuitiva de encontrar la solución a este problema, consiste en aplicar un análisis de componentes principales[2].

¹Tal como se muestra en la Figura 20, el método de mínimos cuadrados considera la distancia verticalmente; sin embargo, en mínimos cuadrados totales se considera la distancia ortogonal desde la recta l hasta los datos. Esta distancia ortogonal es, precisamente, la distancia “verdadera” a la que se hace referencia.

En particular, para encontrar el modelo afín que mejor se ajusta a un conjunto de n correspondencias de rasgos $(x_i, y_i) \rightarrow (\dot{x}_i, \dot{y}_i)$ entre dos imágenes, con $1 \leq i \leq n$, consideramos una descomposición de la transformación expresada según la ecuación (20). Esta simplificación del problema se fundamenta en ajustar dos planos en 3D, tal que se consideran dos regresiones lineales múltiples que buscan encontrar, respectivamente, el valor de 3 parámetros. De esta manera, el objetivo es resolver los sistemas sobredeterminados

$$a_1x + a_2y + a_5 \approx \dot{x} \quad (61)$$

$$a_3x + a_4y + a_6 \approx \dot{y} \quad (62)$$

La ecuación (61) representa la componente horizontal de la transformación, mientras que la 62 corresponde a la vertical.

Podemos ordenar los valores que se tienen del conjunto de correspondencias de la siguiente manera:

$$x = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T \quad (63)$$

$$y = \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix}^T \quad (64)$$

$$\dot{x} = \begin{bmatrix} \dot{x}_1 & \dot{x}_2 & \cdots & \dot{x}_n \end{bmatrix}^T \quad (65)$$

$$\dot{y} = \begin{bmatrix} \dot{y}_1 & \dot{y}_2 & \cdots & \dot{y}_n \end{bmatrix}^T \quad (66)$$

Para cada uno de los conjuntos de datos definidos en (63), (64), (65) y (66), se calcula su media, tal que se obtienen \bar{x} , \bar{y} , $\bar{\dot{x}}$ y $\bar{\dot{y}}$. Luego, se sustraen estos valores de sus conjuntos respectivos, tal que se obtienen 4 nuevas matrices: $x - \bar{x}1$, $y - \bar{y}1$, $\dot{x} - \bar{\dot{x}}1$ y $\dot{y} - \bar{\dot{y}}1$, siendo 1 una matriz de unos de tamaño $n \times 1$.

Para cada regresión lineal, según las variables que considera, se calcula una matriz de covarian-

za², tal que

$$cov_{hor} = \begin{bmatrix} Cov(x, x) & Cov(x, y) & Cov(x, \dot{x}) \\ Cov(y, x) & Cov(y, y) & Cov(y, \dot{x}) \\ Cov(\dot{x}, x) & Cov(\dot{x}, y) & Cov(\dot{x}, \dot{x}) \end{bmatrix} \quad (67)$$

$$cov_{ver} = \begin{bmatrix} Cov(x, x) & Cov(x, y) & Cov(x, \dot{y}) \\ Cov(y, x) & Cov(y, y) & Cov(y, \dot{y}) \\ Cov(\dot{y}, x) & Cov(\dot{y}, y) & Cov(\dot{y}, \dot{y}) \end{bmatrix} \quad (68)$$

Por último, se calculan los autovalores y autovectores asociados a cada matriz. La magnitud de los autovalores indican el orden de significancia de las componentes encontradas. De esta manera, el autovector asociado al autovalor de mayor magnitud representa la componente principal del conjunto de datos.

En el caso de los dos planos en 3D que se desean ajustar, los tres autovectores encontrados en cada caso son ortogonales y la tercera componente, el autovector con autovalor asociado de menor magnitud, representa el error de estimación de la regresión. Siendo este vector perpendicular al plano que mejor se ajusta al conjunto de datos dado y considerando que el centroide de los puntos pasa por esta solución, se puede encontrar, de manera sencilla, la ecuación general del plano

$$AX + BY + CZ + D = 0 \quad (69)$$

tal que en ambas regresiones $C = 1$ y X e Y representan las variables a las cuales se les asignan los valores de x e y , respectivamente. En el caso particular de la regresión que describe la componente horizontal de la transformación, Z representa la variable a la cual corresponden los valores \dot{x} , A es el parámetro a_1 , B es a_2 y D es a_5 . En el otro caso, a Z le corresponden los valores de \dot{y} , mientras que A representa a a_3 , B a a_4 y, por último, D a a_6 .

Sea $\vec{p} = (p_1, p_2, p_3)$ el vector perpendicular al plano, dado por el autovector asociado al autovalor de menor magnitud, y sea $\vec{c} = (c_1, c_2, c_3)$ el centroide del conjunto de puntos considerados. Entonces,

²En el Apéndice A se ofrece la formulación matemática de covarianza entre dos variables.

para cualquier punto (X, Y, Z) en el plano se cumple que

$$((X, Y, Z) - \vec{c}) \cdot \vec{p} = 0 \quad (70)$$

Aplicando la definición de producto escalar al planteamiento anterior, se tiene

$$((X - c_1, Y - c_2, Z - c_3) \cdot (p_1, p_2, p_3)) = 0 \quad (71)$$

$$(X - c_1)p_1 + (Y - c_2)p_2 + (Z - c_3)p_3 = 0 \quad (72)$$

$$Xp_1 - c_1p_1 + Yp_2 - c_2p_2 + Zp_3 - c_3p_3 = 0 \quad (73)$$

$$-p_1X - p_2Y + (c_1p_1 + c_2p_2 + c_3p_3) = p_3Z \quad (74)$$

$$-\frac{p_1}{p_3}X - \frac{p_2}{p_3}Y + \frac{c_1p_1 + c_2p_2 + c_3p_3}{p_3} = Z \quad (75)$$

De esta manera, siendo \vec{p}_{hor} y \vec{p}_{ver} los vectores perpendiculares a los planos que dan solución a cada una de las regresiones lineales planteadas, y $\vec{c}_{hor} = (\bar{x}, \bar{y}, \bar{x})$ y $\vec{c}_{ver} = (\bar{x}, \bar{y}, \bar{y})$ sus centroides respectivos, los 6 parámetros que conforman la transformación afín, según (20), pueden calcularse como

$$a_1 = -\frac{p_{hor1}}{p_{hor3}} \quad (76)$$

$$a_2 = -\frac{p_{hor2}}{p_{hor3}} \quad (77)$$

$$a_3 = -\frac{p_{ver1}}{p_{ver3}} \quad (78)$$

$$a_4 = -\frac{p_{ver2}}{p_{ver3}} \quad (79)$$

$$a_5 = \frac{\bar{x}p_{hor1} + \bar{y}p_{hor2} + \bar{x}p_{hor3}}{p_{hor3}} \quad (80)$$

$$a_6 = \frac{\bar{x}p_{ver1} + \bar{y}p_{ver2} + \bar{y}p_{ver3}}{p_{ver3}} \quad (81)$$

Apéndice I

Descomposición de una transformación lineal

Sea M una transformación lineal,

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad (82)$$

Si el determinante de M no es nulo, entonces M es una matriz no singular (invertible). En este caso, tal como explican Shoemake y Duff[40], su descomposición polar única

$$M = QS \quad (83)$$

cumple que Q es una matriz ortogonal y S es simétrica y definida positiva.¹

Cuando el determinante de Q es positivo, ésta representa una operación de rotación; mientras que cuando en negativo se trata de una reflexión. En este último caso, Q se puede plantear como

$$Q = \begin{bmatrix} c & s \\ s & -c \end{bmatrix} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = RN \quad (84)$$

siendo R una matriz de rotación.

Siguiendo la descripción de (83), S expresa una operación de *stretch*, que en algún sistema de coordenadas rotado representa un escalamiento.

Esta descomposición polar no describe explícitamente la operación de inclinación. Shoemake y Duff explican que la inclinación se obtiene como una composición de rotación y escalamiento. Por ejemplo, considerando H una transformación de inclinación o *shear*, su descomposición puede

¹Las definiciones de matriz ortogonal y definida positiva se presentan en el Apéndice A.

expresarse como

$$\begin{aligned}
 H &= \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \\
 &= \frac{1}{\sqrt{4+h^2}} \begin{bmatrix} 2 & h \\ -h & 2 \end{bmatrix} \begin{bmatrix} 2 & h \\ h & 2+h^2 \end{bmatrix} \frac{1}{\sqrt{4+h^2}} \\
 &= QS
 \end{aligned}$$

La matriz que representa la operación de *stretch*, puede factorizarse, a su vez, según una descomposición espectral. De esta manera,

$$S = UKU^T \quad (85)$$

donde la matriz U representa una rotación y K , siendo diagonal, expresa un escalamiento, usualmente no simétrico a lo largo de los ejes de coordenadas cartesianas[39]. Un problema con esta descomposición es que pudiese no ser única.

Dados los aspectos antes considerados, la matriz inicial M puede factorizarse utilizando una descomposición en valores singulares (SVD), tal que

$$M = VKU^T = (VU^T)(UKU^T) = QS \quad (86)$$

donde V y U son matrices ortogonales y K es diagonal de números no negativos.

Apéndice J

Aplicación de una transformación geométrica

Compensar una imagen f , de tamaño $m \times n$, equivale a aplicarle una transformación geométrica $\mathcal{T}(x, y) = (\dot{x}, \dot{y})$ en 2D. Siendo g la imagen compensada, intuitivamente pareciera que dada esta transformación se busca mover el valor de cada pixel (x, y) , con $0 \leq x < m$ y $0 \leq y < n$, a (\dot{x}, \dot{y}) . Sin embargo, fácilmente sucede que el punto (\dot{x}, \dot{y}) no está formado por componentes enteras, por lo cual habría que decidir de alguna manera dónde posicionar el valor de (x, y) . Por ejemplo, si se utiliza como estrategia llevar (\dot{x}, \dot{y}) a su parte entera, entonces cabe la posibilidad que varios puntos iniciales (x, y) sean asignados en una misma casilla de g y que algunas posiciones en ésta queden vacías.

Sin embargo, considerando el siguiente planteamiento,

$$g(i, j) = f(\mathcal{T}^{-1}(i, j)) \quad (87)$$

se pueden evitar estos problemas. Para cada punto (i, j) en g , con $0 \leq i < m$ y $0 \leq j < n$, se encuentra su mejor correspondencia en f , aplicando algún tipo de interpolación en el caso en que $\mathcal{T}^{-1}(i, j)$ no posea valores enteros.

Apéndice K

Listado de videos extraídos de *YouTube.com* para la fase experimental

A continuación se describen los aspectos más relevantes de los videos extraídos de *YouTube.com* que fueron utilizados en la fase experimental. Dado que generalmente son composiciones de varias grabaciones y les fueron agregadas transiciones al editarlos, éstos fueron divididos en secuencias de imágenes en las cuales el movimiento global que se aprecia es continuo.

Aerial video from Iceland. Este video presenta paisajes montañosos. Durante un período corto se aprecian dos motociclistas que atraviesan un puente a una velocidad considerable.

- Cantidad de secuencias extraídas: 7
- Número total de imágenes consideradas: 1317
- Dirección: <http://www.youtube.com/watch?v=ugwK5lA6qzQ>

Aerial footage. En este caso el helicóptero se mueve manteniendo una velocidad relativamente constante. En general, los movimientos son amplios, aunque a veces se perciben ciertas vibraciones. El reflejo de la luz del sol sobre el lente de la cámara en algunas ocasiones modifica significativamente el nivel de luminosidad de las imágenes.

- Cantidad de secuencias extraídas: 4
- Número total de imágenes consideradas: 1397
- Dirección: <http://www.youtube.com/watch?v=zg6okXl9Mbo>

Flying over stend a nice winterday in bergen (helicam). El helicóptero a veces se mueve a altas velocidades en medio de un campo lleno de nieve. Se observa el movimiento de otros vehículos aéreos a control remoto en el plano de las imágenes.

- Cantidad de secuencias extraídas: 13
- Número total de imágenes consideradas: 6723
- Dirección: <http://www.youtube.com/watch?v=pmOGd7-KkWI>

Heli cam in Bali. En esta secuencia se aprecia un conjunto residencial costero. La calidad del video es baja y fácilmente se evidencia la presencia de ruido.

- Cantidad de secuencias extraídas: 11
- Número total de imágenes consideradas: 2045
- Dirección: <http://www.youtube.com/watch?v=Q64mViTVZMY>

T-Rex 600 Heli Cam. El helicóptero lleva a cabo maniobras acrobáticas y cuando la magnitud de su velocidad es elevada se aprecia una disminución considerable en la calidad de las imágenes. La escena se compone de un campo amplio y no se evidencian fácilmente movimientos independientes de cuerpos u objetos durante la secuencia. Una gran cantidad de vibraciones resultan molestas durante su reproducción.

- Cantidad de secuencias extraídas: 5
- Número total de imágenes consideradas: 3427
- Dirección: <http://www.youtube.com/watch?v=JNhb0M570Ik>

Apéndice L

Detalles sobre los Experimentos I, II y III

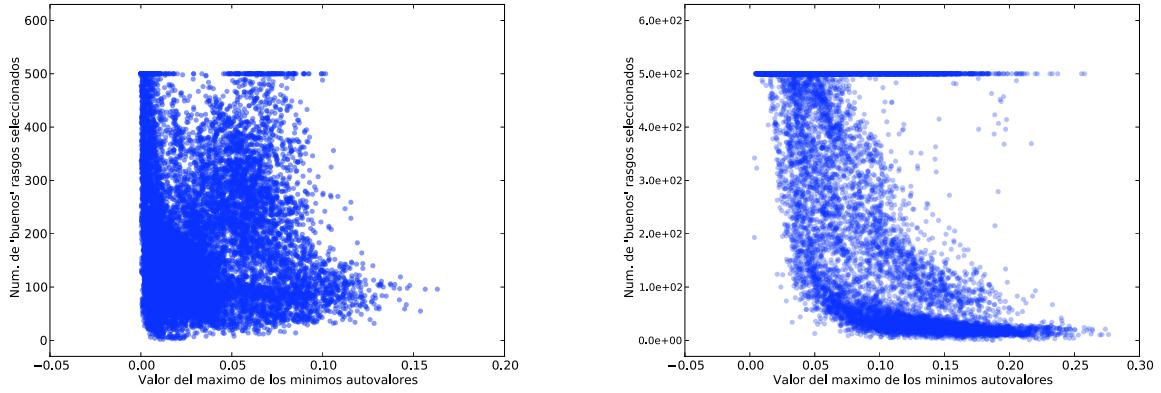
En este Apéndice se presentan detalles sobre los resultados obtenidos en la fase de evaluación del proceso de estimación de flujo óptico.

L.1. Detalles sobre resultados del Experimento I

A continuación se presentan de manera detallada varios de los resultados obtenidos en el Experimento I, sección 4.1.1.

Origen	$T(W_S)$	Prom. Mín.	S.E.M. Mín.	Prom. Calidad	S.E.M. Calidad
<i>YouTube.com</i>	3	1.295×10^{-09}	2.190×10^{-10}	1.648×10^{-03}	1.119×10^{-05}
	5	2.632×10^{-08}	1.618×10^{-09}	1.458×10^{-03}	9.863×10^{-06}
	7	1.437×10^{-07}	4.986×10^{-09}	1.189×10^{-03}	8.214×10^{-06}
	9	4.074×10^{-07}	1.033×10^{-08}	1.012×10^{-03}	7.133×10^{-06}
	11	8.048×10^{-07}	1.744×10^{-08}	8.817×10^{-04}	6.339×10^{-06}
	13	1.304×10^{-06}	2.672×10^{-08}	7.833×10^{-04}	5.794×10^{-06}
<i>ChocoCam</i>	3	6.514×10^{-07}	1.171×10^{-08}	3.671×10^{-03}	2.029×10^{-05}
	5	6.159×10^{-06}	8.878×10^{-08}	2.862×10^{-03}	1.585×10^{-05}
	7	1.706×10^{-05}	2.143×10^{-07}	2.277×10^{-03}	1.265×10^{-05}
	9	3.097×10^{-05}	3.455×10^{-07}	1.881×10^{-03}	1.034×10^{-05}
	11	4.767×10^{-05}	4.778×10^{-07}	1.603×10^{-03}	8.802×10^{-06}
	13	6.390×10^{-05}	6.042×10^{-07}	1.387×10^{-03}	7.505×10^{-06}

Cuadro 3: Promedios del mínimo valor de los mínimos autovalores encontrados y nivel de calidad establecido para calificar un rasgo como “bueno”. La columna siguiente a los valores promedio indica el error estándar de esta media (S.E.M.). Los resultados se agrupan según el origen de los pares de imágenes procesadas y las dimensiones de W_s (en píxeles).



(a) Procesando pares de imágenes provenientes de videos de *YouTube.com*. (b) Procesando pares de imágenes capturadas desde ChocoCam.

Figura 21: Máximo de los mínimos autovalores encontrado vs. número de rasgos seleccionados en promedio, por par de imágenes procesadas en el Experimento I utilizando una ventana W_s de 3×3 .

Origen	$T(W_S)$	$T(W_e) = 3$	$T(W_e) = 5$	$T(W_e) = 7$	$T(W_e) = 9$	$T(W_e) = 11$	$T(W_e) = 13$
<i>YouTube.com</i>	3	24.89	42.05	59.64	77.67	96.42	115.55
	$T(W_e)$	24.89	40.61	56.43	72.11	88.18	104.30
<i>ChocoCam</i>	3	48.09	78.86	109.96	141.36	173.20	205.39
	$T(W_e)$	48.09	78.22	108.84	139.41	169.74	200.07

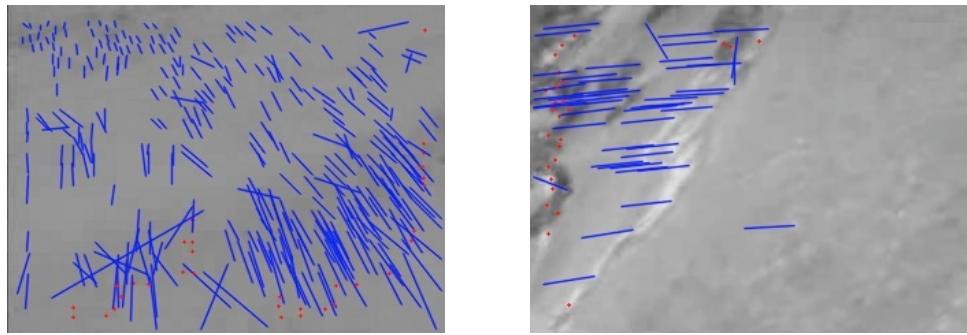
Cuadro 4: Error promedio de estimación de movimiento obtenido en el Experimento I, considerando diferentes valores (en píxeles) para el tamaño de las ventanas W_s y W_e . Los resultados se agrupan según el origen de los pares de imágenes procesadas y las dimensiones de W_s y W_e .

Origen	$T(W_S)$	$T(W_e) = 3$	$T(W_e) = 5$	$T(W_e) = 7$	$T(W_e) = 9$	$T(W_e) = 11$	$T(W_e) = 13$
<i>YouTube.com</i>	3	0.013904	0.022066	0.030524	0.039313	0.048721	0.058480
	$T(W_e)$	0.013904	0.021687	0.029879	0.038182	0.047140	0.056354
<i>ChocoCam</i>	3	0.009583	0.014162	0.019104	0.024121	0.029285	0.034669
	$T(W_e)$	0.009583	0.013934	0.018416	0.022696	0.026986	0.031245

Cuadro 5: Error estándar del promedio de error obtenido en la estimación de movimiento del Experimento I, considerando diferentes valores (en píxeles) para el tamaño de las ventanas W_s y W_e . Los resultados se agrupan según el origen de los pares de imágenes procesadas y las dimensiones de W_s y W_e .

Origen	$T(W_S)$	$T(W_e) = 3$	$T(W_e) = 5$	$T(W_e) = 7$	$T(W_e) = 9$	$T(W_e) = 11$	$T(W_e) = 13$
YouTube.com	3	0.008082	0.004979	0.003606	0.002618	0.002003	0.001524
	$T(W_e)$	0.008082	0.004773	0.003220	0.002510	0.002048	0.001595
ChocoCam	3	0.001246	0.000748	0.000575	0.000483	0.000426	0.000377
	$T(W_e)$	0.001246	0.000714	0.000559	0.000463	0.000401	0.000330

Cuadro 6: Porcentaje de rasgos “perdidos” obtenido en el Experimento I, considerando diferentes valores (en píxeles) para el tamaño de las ventanas W_s y W_e . Los resultados se agrupan según el origen de los pares de imágenes procesadas y las dimensiones de W_s y W_e .



(a) Estimación del campo de movimiento para par de imágenes del segmento de video *FlyingWinter3*.

(b) Estimación del campo de movimiento para par de imágenes del segmento de video *TRex6002*.

Figura 22: Ejemplos de rasgos “perdidos”, utilizando una pirámide de 4 niveles de profundidad máximo, en imágenes de segmentos de videos de *YouTube.com*. En la imagen a la izquierda, particularmente, el flujo óptico resulta difícil de estimar, dada la alta velocidad del vehículo aéreo y el elevado nivel de uniformidad en el valor de intensidad de algunas regiones de la imagen.

L.2. Detalles sobre resultados del Experimento II

A continuación se presentan de manera detallada varios de los resultados obtenidos en el Experimento II, sección 4.1.2.

Origen	Tam. W (píx.)	Máx. L_0	Máx. L_1	Máx. L_2	Máx. L_3	Máx. L_4
<i>YouTube.com</i>	7×7	87.43	66.48	57.87	56.43	56.27
	13×13	140.36	115.38	105.90	104.30	103.98
<i>ChocoCam</i>	7×7	128.74	111.71	108.83	108.84	108.82
	13×13	225.94	204.22	200.08	200.07	200.04

Cuadro 7: Error promedio de estimación de movimiento considerando diferentes profundidades para la pirámide utilizada en la estimación de flujo óptico en el Experimento II. Los resultados se agrupan según el origen de los pares de imágenes procesadas y el tamaño de la vecindad W .

Origen	Tam. W (píx.)	Máx. L_0	Máx. L_1	Máx. L_2	Máx. L_3	Máx. L_4
<i>YouTube.com</i>	7×7	0.083146	0.051414	0.033806	0.029879	0.029059
	13×13	0.124934	0.081493	0.060341	0.056354	0.054535
<i>ChocoCam</i>	7×7	0.040951	0.023806	0.018459	0.018416	0.018323
	13×13	0.063595	0.039412	0.031270	0.031245	0.031110

Cuadro 8: Error estándar del promedio de error obtenido en la estimación de movimiento del Experimento II, considerando diferentes profundidades para la pirámide utilizada. Los resultados se agrupan según el origen de los pares de imágenes procesadas y el tamaño de la vecindad W .

Origen	Tam. W (píx.)	Máx. L_0	Máx. L_1	Máx. L_2	Máx. L_3	Máx. L_4
<i>YouTube.com</i>	7×7	0.001715	0.002055	0.002461	0.003220	0.003714
	13×13	0.000293	0.000588	0.001165	0.001595	0.001734
<i>ChocoCam</i>	7×7	0.000045	0.000155	0.000388	0.000559	0.000696
	13×13	0.000006	0.000065	0.000205	0.000330	0.000383

Cuadro 9: Proporción de rasgos “perdidos” considerando diferentes profundidades para la pirámide utilizada en la estimación de flujo óptico en el Experimento II. Los resultados se agrupan según el origen de los pares de imágenes procesadas y el tamaño de la vecindad W .

L.3. Detalles sobre resultados del Experimento III

A continuación se presentan de manera detallada varios de los resultados obtenidos en el Experimento III, sección 4.1.3.

Origen	Tam. W (píx.)	$\beta = 1.5$	$\beta = 0.7$	$\beta = 0.3$	$\beta = 0.03$	$\beta = 0.003$
<i>YouTube.com</i>	7×7	76.30	63.09	57.87	54.80	54.75
	13×13	134.81	114.23	105.90	101.97	101.92
<i>ChocoCam</i>	7×7	120.47	114.91	108.83	105.65	105.63
	13×13	215.00	207.05	200.08	195.15	195.11

Cuadro 10: Error promedio de estimación de movimiento obtenido en el Experimento III, considerando diferentes valores para la cota β . Los resultados se agrupan según el origen de los pares de imágenes procesadas y las dimensiones de W .

Origen	Tam. W (píx.)	$\beta = 1.5$	$\beta = 0.7$	$\beta = 0.3$	$\beta = 0.03$	$\beta = 0.003$
<i>YouTube.com</i>	7×7	0.060369	0.040547	0.033806	0.032023	0.032000
	13×13	0.106079	0.072542	0.060341	0.057580	0.057570
<i>ChocoCam</i>	7×7	0.024328	0.020116	0.018459	0.018181	0.018184
	13×13	0.039655	0.033262	0.031270	0.030771	0.030772

Cuadro 11: Error estándar del promedio de error obtenido en la estimación de movimiento del Experimento III, considerando diferentes valores para la cota β . Los resultados se agrupan según el origen de los pares de imágenes procesadas y el tamaño de W .

Origen	Tam. W (píx.)	$\beta = 1.5$	$\beta = 0.7$	$\beta = 0.3$	$\beta = 0.03$	$\beta = 0.003$
<i>YouTube.com</i>	7×7	0.001124	0.001855	0.002461	0.002875	0.002879
	13×13	0.000289	0.000711	0.001165	0.001408	0.001417
<i>ChocoCam</i>	7×7	0.000246	0.000341	0.000388	0.000406	0.000406
	13×13	0.000129	0.000182	0.000205	0.000216	0.000216

Cuadro 12: Proporción de rasgos “perdidos” obtenido en el Experimento III, considerando diferentes valores para la cota β . Los resultados se agrupan según el origen de los pares de imágenes procesadas y el tamaño de W .

Apéndice M

Detalles sobre los Experimentos IV,V,VI

En este Apéndice se presentan detalles sobre los resultados obtenidos en la fase de evaluación del proceso de compensación.

M.1. Detalles sobre resultados del Experimento IV

A continuación se presentan de manera detalla varios de los resultados obtenidos en el Experimento IV, sección 4.2.1.

M.1.1. Ganancia con respecto a la diferencia entre imágenes

Ganancia promedio

A continuación se presenta la ganancia promedio obtenida al compensar con respecto a la diferencia de imágenes originales. La definición matemática de esta ganancia se presenta en la sección 4.2.1. Los mejores valores obtenidos se resaltan en negrillas para los casos en los cuales se refina el modelo de transformación iterativamente.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7 × 7	1	10.419	10.419	10.419	10.419	10.419	10.419
		2	10.699	10.782	10.787	10.756	10.535	10.168
		3	10.749	10.814	10.804	10.764	10.520	10.151
		4	10.755	10.822	10.813	10.763	10.515	10.144
	13 × 13	1	10.657	10.657	10.657	10.657	10.657	10.657
		2	10.730	10.764	10.794	10.798	10.678	10.426
		3	10.745	10.781	10.812	10.803	10.663	10.405
		4	10.753	10.785	10.816	10.803	10.657	10.396
<i>ChocoCam</i>	7 × 7	1	4.890	4.890	4.890	4.890	4.890	4.890
		2	4.923	4.948	4.954	4.955	4.922	4.836
		3	4.924	4.955	4.962	4.963	4.925	4.827
		4	4.925	4.959	4.964	4.965	4.925	4.824
	13 × 13	1	5.010	5.010	5.010	5.010	5.010	5.010
		2	5.017	5.025	5.031	5.020	4.999	4.962
		3	5.019	5.034	5.036	5.024	5.001	4.951
		4	5.020	5.037	5.039	5.026	5.001	4.948

Cuadro 13: Ganancia promedio sobre la diferencia entre imágenes consecutivas, compensando con un modelo afín por mínimos cuadrados y máximo 500 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	10.409	10.409	10.409	10.409	10.409	10.409
		2	10.690	10.776	10.782	10.752	10.520	10.136
		3	10.737	10.808	10.798	10.759	10.503	10.117
		4	10.749	10.815	10.809	10.758	10.498	10.110
	13×13	1	10.652	10.652	10.652	10.652	10.652	10.652
		2	10.723	10.755	10.787	10.788	10.680	10.422
		3	10.738	10.777	10.811	10.800	10.662	10.399
		4	10.747	10.782	10.814	10.801	10.657	10.388
<i>ChocoCam</i>	7×7	1	4.885	4.885	4.885	4.885	4.885	4.885
		2	4.914	4.937	4.948	4.946	4.915	4.817
		3	4.914	4.945	4.957	4.954	4.917	4.802
		4	4.916	4.948	4.958	4.955	4.917	4.796
	13×13	1	4.998	4.998	4.998	4.998	4.998	4.998
		2	5.007	5.010	5.019	5.007	4.980	4.933
		3	5.009	5.020	5.024	5.012	4.981	4.914
		4	5.010	5.023	5.026	5.013	4.981	4.903

Cuadro 14: Ganancia promedio sobre la diferencia entre imágenes consecutivas, compensando con un modelo afín por mínimos cuadrados y máximo 300 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	10.350	10.350	10.350	10.350	10.350	10.350
		2	10.617	10.705	10.702	10.682	10.463	10.058
		3	10.663	10.724	10.710	10.696	10.445	10.035
		4	10.673	10.730	10.716	10.698	10.441	10.028
	13×13	1	10.595	10.595	10.595	10.595	10.595	10.595
		2	10.654	10.678	10.705	10.721	10.603	10.319
		3	10.660	10.697	10.730	10.727	10.588	10.296
		4	10.664	10.702	10.736	10.728	10.582	10.289
<i>ChocoCam</i>	7×7	1	4.832	4.832	4.832	4.832	4.832	4.832
		2	4.863	4.887	4.896	4.921	4.903	4.735
		3	4.866	4.893	4.901	4.925	4.893	4.705
		4	4.867	4.896	4.904	4.926	4.890	4.690
	13×13	1	4.929	4.929	4.929	4.929	4.929	4.929
		2	4.933	4.937	4.950	4.945	4.932	4.800
		3	4.934	4.946	4.957	4.950	4.925	4.763
		4	4.935	4.949	4.958	4.951	4.921	4.749

Cuadro 15: Ganancia promedio sobre la diferencia entre imágenes consecutivas, compensando con un modelo afín por mínimos cuadrados y máximo 100 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	10.408	10.408	10.408	10.408	10.408	10.408
		2	10.702	10.785	10.791	10.755	10.529	10.174
		3	10.752	10.813	10.808	10.766	10.512	10.155
		4	10.758	10.820	10.816	10.766	10.506	10.148
	13×13	1	10.653	10.653	10.653	10.653	10.653	10.653
		2	10.731	10.762	10.793	10.801	10.683	10.423
		3	10.746	10.782	10.811	10.806	10.667	10.402
		4	10.751	10.785	10.814	10.807	10.661	10.392
<i>ChocoCam</i>	7×7	1	4.870	4.870	4.870	4.870	4.870	4.870
		2	4.904	4.934	4.943	4.948	4.914	4.830
		3	4.910	4.944	4.954	4.957	4.918	4.820
		4	4.909	4.947	4.956	4.958	4.919	4.818
	13×13	1	5.001	5.001	5.001	5.001	5.001	5.001
		2	5.012	5.022	5.023	5.019	5.000	4.962
		3	5.013	5.028	5.029	5.025	5.001	4.952
		4	5.014	5.032	5.030	5.026	5.001	4.949

Cuadro 16: Ganancia promedio sobre la diferencia entre imágenes consecutivas, compensando con un modelo afín por mínimos cuadrados totales y máximo 500 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	10.398	10.398	10.398	10.398	10.398	10.398
		2	10.692	10.777	10.785	10.750	10.516	10.153
		3	10.742	10.806	10.803	10.760	10.498	10.134
		4	10.750	10.814	10.810	10.760	10.493	10.126
	13×13	1	10.649	10.649	10.649	10.649	10.649	10.649
		2	10.724	10.754	10.787	10.790	10.675	10.419
		3	10.739	10.779	10.809	10.802	10.657	10.397
		4	10.743	10.783	10.813	10.803	10.651	10.385
<i>ChocoCam</i>	7×7	1	4.863	4.863	4.863	4.863	4.863	4.863
		2	4.888	4.927	4.931	4.940	4.902	4.813
		3	4.893	4.939	4.941	4.948	4.904	4.798
		4	4.895	4.942	4.943	4.950	4.904	4.792
	13×13	1	4.991	4.991	4.991	4.991	4.991	4.991
		2	4.995	5.007	5.012	5.009	4.981	4.930
		3	4.996	5.013	5.017	5.013	4.982	4.911
		4	4.996	5.015	5.018	5.014	4.982	4.900

Cuadro 17: Ganancia promedio sobre la diferencia entre imágenes consecutivas, compensando con un modelo afín por mínimos cuadrados totales y máximo 300 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	10.336	10.336	10.336	10.336	10.336	10.336
		2	10.612	10.702	10.713	10.679	10.441	10.042
		3	10.667	10.721	10.721	10.693	10.425	10.020
		4	10.672	10.726	10.727	10.696	10.421	10.014
	13×13	1	10.592	10.592	10.592	10.592	10.592	10.592
		2	10.656	10.676	10.703	10.725	10.605	10.322
		3	10.660	10.697	10.729	10.730	10.590	10.299
		4	10.665	10.702	10.739	10.731	10.584	10.291
<i>ChocoCam</i>	7×7	1	4.806	4.806	4.806	4.806	4.806	4.806
		2	4.844	4.875	4.895	4.904	4.882	4.714
		3	4.851	4.881	4.901	4.907	4.871	4.684
		4	4.854	4.882	4.903	4.910	4.867	4.669
	13×13	1	4.926	4.926	4.926	4.926	4.926	4.926
		2	4.927	4.939	4.946	4.944	4.933	4.799
		3	4.928	4.945	4.953	4.949	4.927	4.762
		4	4.929	4.950	4.956	4.951	4.923	4.749

Cuadro 18: Ganancia promedio sobre la diferencia entre imágenes consecutivas, compensando con un modelo afín por mínimos cuadrados totales y máximo 100 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	8.545	8.545	8.545	8.545	8.545	8.545
		2	8.764	8.542	7.917	7.460	7.794	8.279
		3	8.782	8.522	7.879	7.404	7.718	8.286
		4	8.785	8.519	7.877	7.391	7.703	8.285
	13×13	1	8.714	8.714	8.714	8.714	8.714	8.714
		2	8.762	8.496	7.907	7.525	7.987	8.493
		3	8.763	8.481	7.891	7.471	7.950	8.483
		4	8.764	8.480	7.875	7.463	7.942	8.483
<i>ChocoCam</i>	7×7	1	4.784	4.784	4.784	4.784	4.784	4.784
		2	4.826	4.823	4.810	4.676	4.256	4.348
		3	4.843	4.836	4.819	4.654	4.266	4.369
		4	4.845	4.838	4.819	4.653	4.260	4.365
	13×13	1	4.858	4.858	4.858	4.858	4.858	4.858
		2	4.867	4.870	4.846	4.753	4.396	4.519
		3	4.870	4.880	4.860	4.747	4.396	4.541
		4	4.872	4.881	4.861	4.745	4.389	4.534

Cuadro 19: Ganancia promedio sobre la diferencia entre imágenes consecutivas, compensando con un modelo similar por mínimos cuadrados y máximo 500 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	8.557	8.557	8.557	8.557	8.557	8.557
		2	8.785	8.585	8.006	7.552	7.850	8.334
		3	8.801	8.561	7.955	7.474	7.768	8.339
		4	8.805	8.558	7.946	7.461	7.750	8.338
	13×13	1	8.756	8.756	8.756	8.756	8.756	8.756
		2	8.807	8.580	8.019	7.677	8.134	8.566
		3	8.809	8.562	8.001	7.622	8.103	8.560
		4	8.809	8.558	7.979	7.617	8.095	8.560
<i>ChocoCam</i>	7×7	1	4.777	4.777	4.777	4.777	4.777	4.777
		2	4.816	4.814	4.806	4.667	4.297	4.455
		3	4.832	4.827	4.817	4.643	4.298	4.471
		4	4.834	4.828	4.818	4.644	4.294	4.468
	13×13	1	4.843	4.843	4.843	4.843	4.843	4.843
		2	4.854	4.857	4.836	4.728	4.464	4.597
		3	4.858	4.865	4.847	4.728	4.465	4.606
		4	4.861	4.864	4.843	4.722	4.459	4.601

Cuadro 20: Ganancia promedio sobre la diferencia entre imágenes consecutivas, compensando con un modelo similar por mínimos cuadrados y máximo 300 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	8.519	8.519	8.519	8.519	8.519	8.519
		2	8.743	8.563	8.074	7.831	8.011	8.353
		3	8.766	8.536	8.014	7.756	7.949	8.358
		4	8.769	8.533	8.012	7.741	7.934	8.358
	13×13	1	8.744	8.744	8.744	8.744	8.744	8.744
		2	8.788	8.608	8.200	7.995	8.292	8.595
		3	8.791	8.577	8.163	7.959	8.254	8.590
		4	8.791	8.571	8.151	7.953	8.247	8.590
<i>ChocoCam</i>	7×7	1	4.735	4.735	4.735	4.735	4.735	4.735
		2	4.788	4.793	4.767	4.612	4.382	4.540
		3	4.794	4.796	4.770	4.582	4.384	4.538
		4	4.794	4.796	4.770	4.577	4.383	4.531
	13×13	1	4.792	4.792	4.792	4.792	4.792	4.792
		2	4.808	4.810	4.794	4.650	4.490	4.639
		3	4.811	4.817	4.794	4.641	4.482	4.635
		4	4.811	4.817	4.793	4.639	4.477	4.634

Cuadro 21: Ganancia promedio sobre la diferencia entre imágenes consecutivas, compensando con un modelo similar por mínimos cuadrados y máximo 100 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	10.592	10.592	10.592	10.592	10.592	10.592
		2	10.861	10.927	10.947	10.930	10.721	10.359
		3	10.907	10.991	10.975	10.959	10.711	10.341
		4	10.925	11.000	10.987	10.962	10.704	10.336
	13×13	1	10.821	10.821	10.821	10.821	10.821	10.821
		2	10.898	10.921	10.944	10.956	10.885	10.679
		3	10.916	10.944	10.966	10.972	10.872	10.658
		4	10.924	10.949	10.972	10.977	10.866	10.653
<i>ChocoCam</i>	7×7	1	4.719	4.719	4.719	4.719	4.719	4.719
		2	4.749	4.787	4.793	4.799	4.783	4.636
		3	4.753	4.795	4.801	4.812	4.775	4.623
		4	4.753	4.798	4.803	4.816	4.771	4.619
	13×13	1	4.932	4.932	4.932	4.932	4.932	4.932
		2	4.947	4.962	4.964	4.962	4.927	4.877
		3	4.947	4.966	4.967	4.968	4.926	4.860
		4	4.947	4.968	4.969	4.968	4.926	4.855

Cuadro 22: Ganancia promedio sobre la diferencia entre imágenes consecutivas, compensando con un modelo bilineal por mínimos cuadrados y máximo 500 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	10.579	10.579	10.579	10.579	10.579	10.579
		2	10.845	10.919	10.946	10.914	10.700	10.354
		3	10.888	10.972	10.971	10.941	10.689	10.336
		4	10.905	10.985	10.976	10.944	10.683	10.330
	13×13	1	10.808	10.808	10.808	10.808	10.808	10.808
		2	10.882	10.905	10.926	10.936	10.871	10.651
		3	10.897	10.925	10.944	10.960	10.857	10.629
		4	10.907	10.933	10.961	10.965	10.851	10.622
<i>ChocoCam</i>	7×7	1	4.697	4.697	4.697	4.697	4.697	4.697
		2	4.732	4.761	4.783	4.800	4.783	4.626
		3	4.737	4.771	4.793	4.807	4.777	4.608
		4	4.738	4.775	4.795	4.811	4.773	4.599
	13×13	1	4.916	4.916	4.916	4.916	4.916	4.916
		2	4.924	4.942	4.944	4.949	4.930	4.840
		3	4.923	4.944	4.949	4.957	4.912	4.808
		4	4.924	4.946	4.953	4.958	4.904	4.795

Cuadro 23: Ganancia promedio sobre la diferencia entre imágenes consecutivas, compensando con un modelo bilineal por mínimos cuadrados y máximo 300 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7 × 7	1	10.477	10.477	10.477	10.477	10.477	10.477
		2	10.725	10.816	10.826	10.796	10.582	10.188
		3	10.768	10.863	10.838	10.812	10.569	10.160
		4	10.794	10.866	10.840	10.816	10.564	10.156
	13 × 13	1	10.718	10.718	10.718	10.718	10.718	10.718
		2	10.781	10.796	10.812	10.829	10.776	10.479
		3	10.788	10.805	10.819	10.839	10.762	10.455
		4	10.792	10.811	10.824	10.844	10.758	10.448
<i>ChocoCam</i>	7 × 7	1	4.602	4.602	4.602	4.602	4.602	4.602
		2	4.625	4.662	4.705	4.710	4.697	4.451
		3	4.630	4.676	4.714	4.715	4.695	4.409
		4	4.632	4.676	4.715	4.715	4.695	4.392
	13 × 13	1	4.771	4.771	4.771	4.771	4.771	4.771
		2	4.788	4.797	4.803	4.822	4.803	4.614
		3	4.789	4.803	4.811	4.830	4.804	4.560
		4	4.789	4.805	4.813	4.833	4.804	4.545

Cuadro 24: Ganancia promedio sobre la diferencia entre imágenes consecutivas, compensando con un modelo bilineal por mínimos cuadrados y máximo 100 rasgos “buenos”.

Error estándar de la ganancia promedio

A continuación se presenta el error estándar de la ganancia promedio obtenido en las pruebas del Experimento IV, sección 4.2.1.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	0.101	0.101	0.101	0.101	0.101	0.101
		2	0.099	0.099	0.099	0.099	0.101	0.106
		3	0.099	0.098	0.098	0.099	0.101	0.106
		4	0.099	0.098	0.098	0.099	0.101	0.106
	13×13	1	0.070	0.100	0.100	0.100	0.100	0.100
		2	0.099	0.098	0.097	0.097	0.099	0.105
		3	0.099	0.098	0.097	0.097	0.099	0.104
		4	0.098	0.098	0.097	0.097	0.099	0.104
<i>ChocoCam</i>	7×7	1	0.067	0.067	0.067	0.067	0.067	0.067
		2	0.066	0.066	0.066	0.066	0.066	0.068
		3	0.066	0.065	0.065	0.065	0.066	0.068
		4	0.066	0.065	0.065	0.065	0.066	0.067
	13×13	1	0.064	0.064	0.064	0.064	0.064	0.064
		2	0.064	0.063	0.063	0.063	0.064	0.064
		3	0.063	0.063	0.063	0.063	0.063	0.064
		4	0.063	0.063	0.063	0.063	0.063	0.064

Cuadro 25: Error estándar de la ganancia promedio compensando con un modelo afín por mínimos cuadrados y máximo 500 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	0.100	0.100	0.100	0.100	0.100	0.100
		2	0.099	0.099	0.099	0.099	0.101	0.107
		3	0.099	0.098	0.098	0.099	0.101	0.107
		4	0.099	0.098	0.098	0.099	0.101	0.107
	13×13	1	0.099	0.098	0.098	0.099	0.101	0.107
		2	0.099	0.098	0.097	0.097	0.099	0.104
		3	0.099	0.098	0.097	0.097	0.099	0.104
		4	0.099	0.098	0.097	0.097	0.099	0.104
<i>ChocoCam</i>	7×7	1	0.099	0.098	0.097	0.097	0.099	0.104
		2	0.066	0.066	0.065	0.066	0.065	0.068
		3	0.066	0.065	0.065	0.065	0.065	0.067
		4	0.066	0.065	0.065	0.065	0.065	0.067
	13×13	1	0.066	0.065	0.065	0.065	0.065	0.067
		2	0.063	0.063	0.063	0.063	0.063	0.063
		3	0.063	0.063	0.063	0.063	0.063	0.063
		4	0.063	0.063	0.063	0.063	0.063	0.063

Cuadro 26: Error estándar de la ganancia promedio compensando con un modelo afín por mínimos cuadrados y máximo 300 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	0.101	0.101	0.101	0.101	0.101	0.101
		2	0.100	0.100	0.100	0.099	0.101	0.107
		3	0.100	0.099	0.099	0.099	0.101	0.107
		4	0.100	0.099	0.099	0.099	0.101	0.107
	13×13	1	0.070	0.100	0.100	0.100	0.100	0.100
		2	0.099	0.099	0.098	0.098	0.099	0.105
		3	0.099	0.099	0.098	0.098	0.099	0.105
		4	0.099	0.098	0.097	0.098	0.099	0.105
<i>ChocoCam</i>	7×7	1	0.066	0.066	0.066	0.066	0.066	0.066
		2	0.066	0.065	0.065	0.064	0.064	0.066
		3	0.066	0.065	0.065	0.064	0.064	0.065
		4	0.066	0.065	0.065	0.064	0.064	0.065
	13×13	1	0.063	0.063	0.063	0.063	0.063	0.063
		2	0.063	0.063	0.062	0.062	0.062	0.062
		3	0.063	0.062	0.062	0.062	0.062	0.061
		4	0.063	0.062	0.062	0.062	0.062	0.061

Cuadro 27: Error estándar de la ganancia promedio compensando con un modelo afín por mínimos cuadrados y máximo 100 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	0.100	0.100	0.100	0.100	0.100	0.100
		2	0.099	0.099	0.099	0.099	0.101	0.105
		3	0.099	0.098	0.098	0.099	0.101	0.106
		4	0.099	0.098	0.098	0.099	0.101	0.106
	13×13	1	0.100	0.100	0.100	0.100	0.100	0.100
		2	0.099	0.098	0.097	0.097	0.099	0.105
		3	0.098	0.098	0.097	0.097	0.099	0.105
		4	0.098	0.098	0.097	0.097	0.099	0.105
<i>ChocoCam</i>	7×7	1	0.068	0.068	0.068	0.068	0.068	0.068
		2	0.067	0.066	0.066	0.066	0.066	0.068
		3	0.067	0.066	0.066	0.066	0.066	0.068
		4	0.067	0.066	0.066	0.066	0.066	0.068
	13×13	1	0.064	0.064	0.064	0.064	0.064	0.064
		2	0.064	0.064	0.064	0.064	0.064	0.064
		3	0.064	0.063	0.064	0.063	0.064	0.064
		4	0.064	0.063	0.064	0.063	0.064	0.064

Cuadro 28: Error estándar de la ganancia promedio compensando con un modelo afín por mínimos cuadrados totales y máximo 500 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	0.100	0.100	0.100	0.100	0.100	0.100
		2	0.099	0.099	0.099	0.099	0.101	0.106
		3	0.099	0.098	0.098	0.098	0.101	0.106
		4	0.099	0.098	0.098	0.098	0.101	0.106
	13×13	1	0.099	0.098	0.098	0.098	0.101	0.106
		2	0.099	0.098	0.097	0.097	0.099	0.104
		3	0.098	0.098	0.097	0.097	0.099	0.104
		4	0.098	0.098	0.097	0.097	0.099	0.104
<i>ChocoCam</i>	7×7	1	0.098	0.098	0.097	0.097	0.099	0.104
		2	0.068	0.068	0.068	0.068	0.068	0.068
		3	0.067	0.066	0.066	0.066	0.066	0.068
		4	0.067	0.066	0.066	0.066	0.066	0.067
	13×13	1	0.067	0.066	0.066	0.066	0.066	0.067
		2	0.064	0.064	0.064	0.063	0.064	0.064
		3	0.064	0.063	0.063	0.063	0.063	0.064
		4	0.064	0.063	0.063	0.063	0.063	0.063

Cuadro 29: Error estándar de la ganancia promedio compensando con un modelo afín por mínimos cuadrados totales y máximo 300 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	0.101	0.101	0.101	0.101	0.101	0.101
		2	0.100	0.100	0.099	0.099	0.102	0.107
		3	0.100	0.099	0.099	0.099	0.102	0.107
		4	0.100	0.099	0.099	0.099	0.102	0.107
	13×13	1	0.100	0.100	0.100	0.100	0.100	0.100
		2	0.099	0.099	0.098	0.098	0.099	0.105
		3	0.099	0.099	0.098	0.098	0.099	0.105
		4	0.099	0.098	0.097	0.098	0.099	0.105
<i>ChocoCam</i>	7×7	1	0.068	0.068	0.068	0.068	0.068	0.068
		2	0.066	0.066	0.065	0.065	0.065	0.067
		3	0.066	0.066	0.065	0.065	0.065	0.066
		4	0.066	0.066	0.065	0.065	0.065	0.066
	13×13	1	0.063	0.063	0.063	0.063	0.063	0.063
		2	0.063	0.063	0.062	0.063	0.062	0.062
		3	0.063	0.062	0.062	0.062	0.062	0.061
		4	0.063	0.062	0.062	0.062	0.062	0.061

Cuadro 30: Error estándar de la ganancia promedio compensando con un modelo afín por mínimos cuadrados totales y máximo 100 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	0.103	0.103	0.103	0.103	0.103	0.103
		2	0.105	0.106	0.112	0.113	0.106	0.107
		3	0.105	0.107	0.112	0.113	0.106	0.107
		4	0.105	0.107	0.112	0.113	0.107	0.107
	13×13	1	0.104	0.104	0.104	0.104	0.104	0.104
		2	0.104	0.107	0.113	0.113	0.106	0.106
		3	0.104	0.107	0.112	0.114	0.107	0.107
		4	0.104	0.107	0.112	0.114	0.107	0.107
<i>ChocoCam</i>	7×7	1	0.062	0.062	0.062	0.062	0.062	0.062
		2	0.061	0.061	0.062	0.063	0.065	0.062
		3	0.060	0.061	0.061	0.064	0.065	0.062
		4	0.060	0.061	0.061	0.064	0.065	0.062
	13×13	1	0.060	0.060	0.060	0.060	0.060	0.060
		2	0.060	0.060	0.061	0.062	0.064	0.060
		3	0.060	0.059	0.060	0.061	0.064	0.060
		4	0.060	0.059	0.060	0.061	0.064	0.060

Cuadro 31: Error estándar de la ganancia promedio compensando con un modelo similar por mínimos cuadrados y máximo 500 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	0.103	0.103	0.103	0.103	0.103	0.103
		2	0.104	0.106	0.111	0.112	0.106	0.106
		3	0.105	0.106	0.111	0.113	0.106	0.106
		4	0.105	0.106	0.111	0.113	0.107	0.106
	13×13	1	0.105	0.106	0.111	0.113	0.107	0.106
		2	0.104	0.105	0.111	0.110	0.105	0.106
		3	0.104	0.106	0.111	0.111	0.106	0.106
		4	0.104	0.106	0.111	0.111	0.106	0.106
<i>ChocoCam</i>	7×7	1	0.104	0.106	0.111	0.111	0.106	0.106
		2	0.061	0.061	0.061	0.063	0.065	0.061
		3	0.060	0.061	0.061	0.063	0.065	0.062
		4	0.060	0.061	0.061	0.063	0.065	0.061
	13×13	1	0.060	0.061	0.061	0.063	0.065	0.061
		2	0.060	0.060	0.060	0.062	0.063	0.060
		3	0.060	0.059	0.060	0.061	0.063	0.060
		4	0.059	0.059	0.060	0.061	0.063	0.060

Cuadro 32: Error estándar de la ganancia promedio compensando con un modelo similar por mínimos cuadrados y máximo 300 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	0.104	0.104	0.104	0.104	0.104	0.104
		2	0.105	0.107	0.111	0.111	0.104	0.106
		3	0.106	0.108	0.112	0.111	0.105	0.106
		4	0.106	0.108	0.112	0.111	0.105	0.106
	13×13	1	0.105	0.105	0.105	0.105	0.105	0.105
		2	0.104	0.106	0.109	0.108	0.104	0.106
		3	0.104	0.107	0.109	0.108	0.105	0.106
		4	0.104	0.107	0.110	0.108	0.105	0.106
<i>ChocoCam</i>	7×7	1	0.062	0.062	0.062	0.062	0.062	0.062
		2	0.060	0.060	0.060	0.063	0.064	0.061
		3	0.060	0.060	0.060	0.064	0.064	0.061
		4	0.060	0.060	0.060	0.064	0.064	0.061
	13×13	1	0.059	0.059	0.059	0.059	0.059	0.059
		2	0.059	0.059	0.059	0.061	0.063	0.060
		3	0.059	0.059	0.059	0.061	0.063	0.060
		4	0.059	0.059	0.059	0.061	0.063	0.060

Cuadro 33: Error estándar de la ganancia promedio compensando con un modelo similar por mínimos cuadrados y máximo 100 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	0.103	0.103	0.103	0.103	0.103	0.103
		2	0.102	0.103	0.103	0.102	0.105	0.109
		3	0.102	0.103	0.103	0.102	0.104	0.109
		4	0.102	0.103	0.102	0.101	0.104	0.109
	13×13	1	0.102	0.103	0.102	0.101	0.104	0.109
		2	0.104	0.103	0.103	0.103	0.103	0.106
		3	0.103	0.103	0.103	0.102	0.103	0.106
		4	0.103	0.103	0.102	0.102	0.103	0.106
<i>ChocoCam</i>	7×7	1	0.074	0.074	0.074	0.074	0.074	0.074
		2	0.074	0.073	0.073	0.073	0.072	0.076
		3	0.074	0.073	0.073	0.073	0.072	0.076
		4	0.074	0.073	0.073	0.072	0.072	0.075
	13×13	1	0.068	0.068	0.068	0.068	0.068	0.068
		2	0.067	0.067	0.067	0.067	0.067	0.068
		3	0.067	0.067	0.067	0.067	0.066	0.068
		4	0.067	0.067	0.067	0.067	0.066	0.067

Cuadro 34: Error estándar de la ganancia promedio compensando con un modelo bilineal por mínimos cuadrados y máximo 500 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	0.103	0.103	0.103	0.103	0.103	0.103
		2	0.102	0.103	0.103	0.103	0.105	0.108
		3	0.102	0.103	0.102	0.102	0.105	0.108
		4	0.102	0.103	0.102	0.102	0.105	0.108
	13×13	1	0.104	0.104	0.104	0.104	0.104	0.104
		2	0.104	0.104	0.104	0.104	0.104	0.104
		3	0.104	0.103	0.103	0.103	0.103	0.106
		4	0.103	0.103	0.103	0.102	0.103	0.106
<i>ChocoCam</i>	7×7	1	0.075	0.075	0.075	0.075	0.075	0.075
		2	0.074	0.074	0.073	0.072	0.072	0.075
		3	0.074	0.074	0.073	0.072	0.072	0.075
		4	0.074	0.074	0.073	0.072	0.072	0.075
	13×13	1	0.074	0.073	0.073	0.072	0.072	0.075
		2	0.067	0.067	0.067	0.067	0.066	0.066
		3	0.068	0.067	0.067	0.066	0.066	0.066
		4	0.067	0.067	0.067	0.066	0.066	0.066

Cuadro 35: Error estándar de la ganancia promedio compensando con un modelo bilineal por mínimos cuadrados y máximo 300 rasgos “buenos”.

Origen	Tam. W	Máx. #It.	$\varepsilon = 15.0$	$\varepsilon = 8.0$	$\varepsilon = 5.0$	$\varepsilon = 3.0$	$\varepsilon = 1.0$	$\varepsilon = 0.25$
<i>YouTube.com</i>	7×7	1	0.104	0.104	0.104	0.104	0.104	0.104
		2	0.103	0.104	0.104	0.104	0.106	0.110
		3	0.103	0.104	0.104	0.104	0.106	0.110
		4	0.103	0.104	0.104	0.103	0.106	0.110
	13×13	1	0.105	0.105	0.105	0.105	0.105	0.105
		2	0.104	0.104	0.104	0.104	0.104	0.108
		3	0.104	0.104	0.104	0.103	0.104	0.108
		4	0.104	0.104	0.104	0.103	0.104	0.107
<i>ChocoCam</i>	7×7	1	0.104	0.104	0.104	0.103	0.104	0.107
		2	0.074	0.073	0.072	0.072	0.073	0.075
		3	0.074	0.073	0.072	0.072	0.073	0.074
		4	0.074	0.073	0.072	0.072	0.073	0.074
	13×13	1	0.067	0.067	0.067	0.067	0.067	0.067
		2	0.067	0.066	0.066	0.065	0.066	0.066
		3	0.066	0.066	0.066	0.065	0.066	0.065
		4	0.066	0.066	0.066	0.065	0.066	0.065

Cuadro 36: Error estándar de la ganancia promedio compensando con un modelo bilineal por mínimos cuadrados y máximo 100 rasgos “buenos”.

Comparación de las mejoras ganacias promedio

De forma resumida, en los siguientes cuadros se muestran las mejores ganancias obtenidas para cierto número máximo de iteraciones y modelo y método de estimación correspondiente, bajo un máximo de 500 o 300 rasgos “buenos”¹. M.C. indica mínimos cuadrados y M.C.T. se refiere a mínimos cuadrados totales. En los resultados correspondientes a los casos en los que se llevó a cabo sólo una iteración, no es relevante el valor de ε .

Origen	$T(W)$	Modelo/Método	Máx. 1 It.	Máx. 2 Its.	Máx. 3 Its.	Máx. 4 Its.
<i>You Tube</i>	7	Afín/M.C.	10.419	10.787 ($\varepsilon = 5.0$)	10.814 ($\varepsilon = 8.0$)	10.822 ($\varepsilon = 8.0$)
		Afín/M.C.T	10.408	10.791 ($\varepsilon = 5.0$)	10.813 ($\varepsilon = 8.0$)	10.820 ($\varepsilon = 8.0$)
		Similar/M.C.	8.545	8.764 ($\varepsilon = 15.0$)	8.782 ($\varepsilon = 15.0$)	8.785 ($\varepsilon = 15.0$)
		Bilineal/M.C	10.592	10.497 ($\varepsilon = 5.0$)	10.991 ($\varepsilon = 8.0$)	11.000 ($\varepsilon = 8.0$)
	13	Afín/M.C	10.657	10.798 ($\varepsilon = 3.0$)	10.812 ($\varepsilon = 5.0$)	10.816 ($\varepsilon = 5.0$)
		Afín/M.C.T	10.653	10.801 ($\varepsilon = 3.0$)	10.811 ($\varepsilon = 5.0$)	10.814 ($\varepsilon = 5.0$)
		Similar/M.C.	8.714	8.762 ($\varepsilon = 15.0$)	8.763 ($\varepsilon = 15.0$)	8.764 ($\varepsilon = 15.0$)
		Bilineal/M.C	10.821	10.956 ($\varepsilon = 3.0$)	10.972 ($\varepsilon = 3.0$)	10.977 ($\varepsilon = 3.0$)
<i>Choco Cam</i>	7	Afín/M.C	4.890	4.955 ($\varepsilon = 3.0$)	4.963 ($\varepsilon = 3.0$)	4.965 ($\varepsilon = 3.0$)
		Afín/M.C.T	4.870	4.948 ($\varepsilon = 3.0$)	4.957 ($\varepsilon = 3.0$)	4.958 ($\varepsilon = 3.0$)
		Similar/M.C.	4.784	4.826 ($\varepsilon = 15.0$)	4.843 ($\varepsilon = 15.0$)	4.845 ($\varepsilon = 15.0$)
		Bilineal/M.C	4.719	4.799 ($\varepsilon = 3.0$)	4.812 ($\varepsilon = 3.0$)	4.816 ($\varepsilon = 3.0$)
	13	Afín/M.C	5.010	5.031 ($\varepsilon = 5.0$)	5.036 ($\varepsilon = 5.0$)	5.039 ($\varepsilon = 5.0$)
		Afín/M.C.T	5.001	5.023 ($\varepsilon = 5.0$)	5.029 ($\varepsilon = 5.0$)	5.032 ($\varepsilon = 8.0$)
		Similar/M.C.	4.858	4.870 ($\varepsilon = 8.0$)	4.880 ($\varepsilon = 8.0$)	4.881 ($\varepsilon = 8.0$)
		Bilineal/M.C	4.932	4.964 ($\varepsilon = 5.0$)	4.968 ($\varepsilon = 3.0$)	4.969 ($\varepsilon = 5.0$)

Cuadro 37: Mejores ganancias obtenidas en promedio para un máximo de 500 rasgos.

¹En la sección 4.2.1 se detallan las mejores ganancias obtenidas en promedio para un máximo de 100 rasgos.

Origen	$T(W)$	Modelo/Método	Máx. 1 It.	Máx. 2 Its.	Máx. 3 Its.	Máx. 4 Its.
<i>You Tube</i>	7	Afín/M.C.	10.409	10.782 ($\varepsilon = 5.0$)	10.808 ($\varepsilon = 8.0$)	10.815 ($\varepsilon = 8.0$)
		Afín/M.C.T	10.398	10.785 ($\varepsilon = 5.0$)	10.806 ($\varepsilon = 8.0$)	10.814 ($\varepsilon = 8.0$)
		Similar/M.C.	8.557	8.785 ($\varepsilon = 15.0$)	8.801 ($\varepsilon = 15.0$)	8.805 ($\varepsilon = 15.0$)
		Bilineal/M.C	10.579	10.946 ($\varepsilon = 5.0$)	10.972 ($\varepsilon = 8.0$)	10.985 ($\varepsilon = 8.0$)
	13	Afín/M.C	10.652	10.788 ($\varepsilon = 5.0$)	10.811 ($\varepsilon = 5.0$)	10.814 ($\varepsilon = 5.0$)
		Afín/M.C.T	10.649	10.790 ($\varepsilon = 3.0$)	10.809 ($\varepsilon = 5.0$)	10.813 ($\varepsilon = 5.0$)
		Similar/M.C.	8.756	8.807 ($\varepsilon = 15.0$)	8.809 ($\varepsilon = 15.0$)	8.809 ($\varepsilon = 15.0$)
		Bilineal/M.C	10.808	10.936 ($\varepsilon = 3.0$)	10.960 ($\varepsilon = 3.0$)	10.965 ($\varepsilon = 3.0$)
<i>Choco Cam</i>	7	Afín/M.C	4.885	4.984 ($\varepsilon = 5.0$)	4.957 ($\varepsilon = 5.0$)	4.958 ($\varepsilon = 5.0$)
		Afín/M.C.T	4.863	4.940 ($\varepsilon = 3.0$)	4.948 ($\varepsilon = 3.0$)	4.950 ($\varepsilon = 3.0$)
		Similar/M.C.	4.777	4.816 ($\varepsilon = 15.0$)	4.832 ($\varepsilon = 15.0$)	4.834 ($\varepsilon = 15.0$)
		Bilineal/M.C	4.697	4.783 ($\varepsilon = 5.0$)	4.793 ($\varepsilon = 5.0$)	4.811 ($\varepsilon = 3.0$)
	13	Afín/M.C	4.998	5.019 ($\varepsilon = 5.0$)	5.024 ($\varepsilon = 5.0$)	5.026 ($\varepsilon = 5.0$)
		Afín/M.C.T	4.991	5.012 ($\varepsilon = 5.0$)	5.017 ($\varepsilon = 5.0$)	5.018 ($\varepsilon = 5.0$)
		Similar/M.C.	4.843	4.857 ($\varepsilon = 8.0$)	4.865 ($\varepsilon = 8.0$)	4.864 ($\varepsilon = 8.0$)
		Bilineal/M.C	4.916	4.949 ($\varepsilon = 3.0$)	4.957 ($\varepsilon = 3.0$)	4.958 ($\varepsilon = 3.0$)

Cuadro 38: Mejores ganancias obtenidas en promedio para un máximo de 300 rasgos.

M.1.2. Número de iteraciones

A continuación se presenta el promedio de iteraciones llevadas a cabo para estimar, de manera iterativa, la transformación que modela el movimiento global. M.C. indica mínimos cuadrados y M.C.T. se refiere a mínimos cuadrados totales.

Origen	$T(W)$	Modelo/Método	Máx. 2 Its.	Máx. 3 Its.	Máx. 4 Its.
<i>YouTube.com</i>	7	Afín/M.C.	1.240 ($\varepsilon = 5.0$)	1.273 ($\varepsilon = 8.0$)	1.302 ($\varepsilon = 8.0$)
		Afín/M.C.T	1.240 ($\varepsilon = 5.0$)	1.271 ($\varepsilon = 8.0$)	1.296 ($\varepsilon = 8.0$)
		Similar/M.C.	1.231 ($\varepsilon = 15.0$)	1.306 ($\varepsilon = 15.0$)	1.332 ($\varepsilon = 15.0$)
		Bilineal/M.C	1.237 ($\varepsilon = 5.0$)	1.275 ($\varepsilon = 8.0$)	1.298 ($\varepsilon = 8.0$)
	13	Afín/M.C	1.201 ($\varepsilon = 3.0$)	1.228 ($\varepsilon = 5.0$)	1.258 ($\varepsilon = 5.0$)
		Afin/M.C.T	1.201 ($\varepsilon = 3.0$)	1.228 ($\varepsilon = 5.0$)	1.259 ($\varepsilon = 5.0$)
		Similar/M.C.	1.165 ($\varepsilon = 15.0$)	1.209 ($\varepsilon = 15.0$)	1.225 ($\varepsilon = 15.0$)
		Bilineal/M.C	1.188 ($\varepsilon = 3.0$)	1.267 ($\varepsilon = 3.0$)	1.301 ($\varepsilon = 3.0$)
<i>ChocoCam</i>	7	Afín/M.C	1.282 ($\varepsilon = 3.0$)	1.364 ($\varepsilon = 3.0$)	1.391 ($\varepsilon = 3.0$)
		Afín/M.C.T	1.281 ($\varepsilon = 3.0$)	1.362 ($\varepsilon = 3.0$)	1.389 ($\varepsilon = 3.0$)
		Similar/M.C.	1.093 ($\varepsilon = 15.0$)	1.126 ($\varepsilon = 15.0$)	1.139 ($\varepsilon = 15.0$)
		Bilineal/M.C	1.275 ($\varepsilon = 3.0$)	1.355 ($\varepsilon = 3.0$)	1.380 ($\varepsilon = 3.0$)
	13	Afín/M.C	1.096 ($\varepsilon = 5.0$)	1.138 ($\varepsilon = 5.0$)	1.154 ($\varepsilon = 5.0$)
		Afn/M.C.T	1.097 ($\varepsilon = 5.0$)	1.242 ($\varepsilon = 5.0$)	1.265 ($\varepsilon = 8.0$)
		Similar/M.C.	1.092 ($\varepsilon = 8.0$)	1.120 ($\varepsilon = 8.0$)	1.128 ($\varepsilon = 8.0$)
		Bilineal/M.C	1.095 ($\varepsilon = 5.0$)	1.236 ($\varepsilon = 3.0$)	1.152 ($\varepsilon = 5.0$)

Cuadro 39: Promedio de iteraciones para estimar el modelo de transformación global con un máximo de 500 rasgos. Los resultados se agrupan considerando el máximo de iteraciones permitido, el modelo y el método de estimación utilizado, el tamaño de W y el origen de los videos.

Origen	$T(W)$	Modelo/Método	Máx. 2 Its.	Máx. 3 Its.	Máx. 4 Its.
<i>YouTube.com</i>	7	Afín/M.C.	0.010 ($\varepsilon = 5.0$)	0.013 ($\varepsilon = 8.0$)	0.016 ($\varepsilon = 8.0$)
		Afín/M.C.T	0.010 ($\varepsilon = 5.0$)	0.013 ($\varepsilon = 8.0$)	0.015 ($\varepsilon = 8.0$)
		Similar/M.C.	0.009 ($\varepsilon = 15.0$)	0.014 ($\varepsilon = 15.0$)	0.016 ($\varepsilon = 15.0$)
		Bilineal/M.C	0.010 ($\varepsilon = 5.0$)	0.013 ($\varepsilon = 8.0$)	0.015 ($\varepsilon = 8.0$)
	13	Afín/M.C	0.009 ($\varepsilon = 3.0$)	0.013 ($\varepsilon = 5.0$)	0.015 ($\varepsilon = 5.0$)
		Afín/M.C.T	0.009 ($\varepsilon = 3.0$)	0.013 ($\varepsilon = 5.0$)	0.015 ($\varepsilon = 5.0$)
		Similar/M.C.	0.008 ($\varepsilon = 15.0$)	0.011 ($\varepsilon = 15.0$)	0.013 ($\varepsilon = 15.0$)
		Bilineal/M.C	0.009 ($\varepsilon = 3.0$)	0.014 ($\varepsilon = 3.0$)	0.016 ($\varepsilon = 3.0$)
<i>ChocoCam</i>	7	Afín/M.C	0.010 ($\varepsilon = 3.0$)	0.014 ($\varepsilon = 3.0$)	0.016 ($\varepsilon = 3.0$)
		Afín/M.C.T	0.010 ($\varepsilon = 3.0$)	0.014 ($\varepsilon = 3.0$)	0.016 ($\varepsilon = 3.0$)
		Similar/M.C.	0.006 ($\varepsilon = 15.0$)	0.009 ($\varepsilon = 15.0$)	0.011 ($\varepsilon = 15.0$)
		Bilineal/M.C	0.010 ($\varepsilon = 3.0$)	0.014 ($\varepsilon = 3.0$)	0.016 ($\varepsilon = 3.0$)
	13	Afín/M.C	0.006 ($\varepsilon = 5.0$)	0.010 ($\varepsilon = 5.0$)	0.012 ($\varepsilon = 5.0$)
		Afín/M.C.T	0.006 ($\varepsilon = 5.0$)	0.010 ($\varepsilon = 5.0$)	0.011 ($\varepsilon = 8.0$)
		Similar/M.C.	0.006 ($\varepsilon = 8.0$)	0.009 ($\varepsilon = 8.0$)	0.010 ($\varepsilon = 8.0$)
		Bilineal/M.C	0.006 ($\varepsilon = 5.0$)	0.012 ($\varepsilon = 3.0$)	0.0121 ($\varepsilon = 5.0$)

Cuadro 40: Error estándar del promedio de iteraciones llevadas a cabo para estimar el modelo de transformación global con un máximo de 500 rasgos. Los resultados se agrupan considerando el máximo de iteraciones permitido, el modelo y el método de estimación utilizado, el tamaño de W y el origen de los videos.

Origen	$T(W)$	Modelo/Método	Máx. 2 Its.	Máx. 3 Its.	Máx. 4 Its.
<i>YouTube.com</i>	7	Afín/M.C.	1.224 ($\varepsilon = 5.0$)	1.257 ($\varepsilon = 8.0$)	1.284 ($\varepsilon = 8.0$)
		Afín/M.C.T	1.224 ($\varepsilon = 5.0$)	1.258 ($\varepsilon = 8.0$)	1.280 ($\varepsilon = 8.0$)
		Similar/M.C.	1.219 ($\varepsilon = 15.0$)	1.290 ($\varepsilon = 15.0$)	1.315 ($\varepsilon = 15.0$)
		Bilineal/M.C	1.221 ($\varepsilon = 5.0$)	1.262 ($\varepsilon = 8.0$)	1.284 ($\varepsilon = 8.0$)
	13	Afín/M.C	1.144 ($\varepsilon = 5.0$)	1.211 ($\varepsilon = 5.0$)	1.236 ($\varepsilon = 5.0$)
		Afín/M.C.T	1.183 ($\varepsilon = 3.0$)	1.211 ($\varepsilon = 5.0$)	1.237 ($\varepsilon = 5.0$)
		Similar/M.C.	1.152 ($\varepsilon = 15.0$)	1.193 ($\varepsilon = 15.0$)	1.203 ($\varepsilon = 15.0$)
		Bilineal/M.C	1.173 ($\varepsilon = 3.0$)	1.245 ($\varepsilon = 3.0$)	1.276 ($\varepsilon = 3.0$)
<i>ChocoCam</i>	7	Afín/M.C	1.154 ($\varepsilon = 5.0$)	1.201 ($\varepsilon = 5.0$)	1.218 ($\varepsilon = 5.0$)
		Afín/M.C.T	1.226 ($\varepsilon = 3.0$)	1.284 ($\varepsilon = 3.0$)	1.303 ($\varepsilon = 3.0$)
		Similar/M.C.	1.083 ($\varepsilon = 15.0$)	1.111 ($\varepsilon = 15.0$)	1.119 ($\varepsilon = 15.0$)
		Bilineal/M.C	1.153 ($\varepsilon = 5.0$)	1.200 ($\varepsilon = 5.0$)	1.293 ($\varepsilon = 3.0$)
	13	Afín/M.C	1.081 ($\varepsilon = 5.0$)	1.114 ($\varepsilon = 5.0$)	1.127 ($\varepsilon = 5.0$)
		Afín/M.C.T	1.082 ($\varepsilon = 5.0$)	1.118 ($\varepsilon = 5.0$)	1.132 ($\varepsilon = 5.0$)
		Similar/M.C.	1.081 ($\varepsilon = 8.0$)	1.107 ($\varepsilon = 8.0$)	1.113 ($\varepsilon = 8.0$)
		Bilineal/M.C	1.125 ($\varepsilon = 3.0$)	1.163 ($\varepsilon = 3.0$)	1.174 ($\varepsilon = 3.0$)

Cuadro 41: Promedio de iteraciones para estimar el modelo de transformación global con un máximo de 300 rasgos. Los resultados se agrupan considerando el máximo de iteraciones permitido, el modelo y el método de estimación utilizado, el tamaño de W y el origen de los videos.

Origen	$T(W)$	Modelo/Método	Máx. 2 Its.	Máx. 3 Its.	Máx. 4 Its.
<i>YouTube.com</i>	7	Afín/M.C.	0.009 ($\varepsilon = 5.0$)	0.013 ($\varepsilon = 8.0$)	0.015 ($\varepsilon = 8.0$)
		Afín/M.C.T	0.009 ($\varepsilon = 5.0$)	0.013 ($\varepsilon = 8.0$)	0.015 ($\varepsilon = 8.0$)
		Similar/M.C.	0.009 ($\varepsilon = 15.0$)	0.013 ($\varepsilon = 15.0$)	0.016 ($\varepsilon = 15.0$)
		Bilineal/M.C	0.009 ($\varepsilon = 5.0$)	0.013 ($\varepsilon = 8.0$)	0.015 ($\varepsilon = 8.0$)
	13	Afín/M.C	0.008 ($\varepsilon = 5.0$)	0.012 ($\varepsilon = 5.0$)	0.015 ($\varepsilon = 5.0$)
		Afín/M.C.T	0.009 ($\varepsilon = 3.0$)	0.012 ($\varepsilon = 5.0$)	0.015 ($\varepsilon = 5.0$)
		Similar/M.C.	0.008 ($\varepsilon = 15.0$)	0.011 ($\varepsilon = 15.0$)	0.012 ($\varepsilon = 15.0$)
		Bilineal/M.C	0.008 ($\varepsilon = 3.0$)	0.013 ($\varepsilon = 3.0$)	0.016 ($\varepsilon = 3.0$)
<i>ChocoCam</i>	7	Afín/M.C	0.008 ($\varepsilon = 5.0$)	0.011 ($\varepsilon = 5.0$)	0.013 ($\varepsilon = 5.0$)
		Afín/M.C.T	0.009 ($\varepsilon = 3.0$)	0.013 ($\varepsilon = 3.0$)	0.014 ($\varepsilon = 3.0$)
		Similar/M.C.	0.006 ($\varepsilon = 15.0$)	0.009 ($\varepsilon = 15.0$)	0.010 ($\varepsilon = 15.0$)
		Bilineal/M.C	0.008 ($\varepsilon = 5.0$)	0.011 ($\varepsilon = 5.0$)	0.014 ($\varepsilon = 3.0$)
	13	Afín/M.C	0.006 ($\varepsilon = 5.0$)	0.009 ($\varepsilon = 5.0$)	0.011 ($\varepsilon = 5.0$)
		Afín/M.C.T	0.006 ($\varepsilon = 5.0$)	0.009 ($\varepsilon = 5.0$)	0.011 ($\varepsilon = 5.0$)
		Similar/M.C.	0.006 ($\varepsilon = 8.0$)	0.008 ($\varepsilon = 8.0$)	0.009 ($\varepsilon = 8.0$)
		Bilineal/M.C	0.007 ($\varepsilon = 3.0$)	0.010 ($\varepsilon = 3.0$)	0.011 ($\varepsilon = 3.0$)

Cuadro 42: Error estándar del promedio de iteraciones llevadas a cabo para estimar el modelo de transformación global con un máximo de 300 rasgos. Los resultados se agrupan considerando el máximo de iteraciones permitido, el modelo y el método de estimación utilizado, el tamaño de W y el origen de los videos.

Origen	$T(W)$	Modelo/Método	Máx. 2 Its.	Máx. 3 Its.	Máx. 4 Its.
<i>YouTube.com</i>	7	Afín/M.C.	1.166 ($\varepsilon = 8.0$)	1.218 ($\varepsilon = 8.0$)	1.231 ($\varepsilon = 8.0$)
		Afín/M.C.T	1.188 ($\varepsilon = 5.0$)	1.212 ($\varepsilon = 8.0$)	1.262 ($\varepsilon = 5.0$)
		Similar/M.C.	1.188 ($\varepsilon = 15.0$)	1.256 ($\varepsilon = 15.0$)	1.275 ($\varepsilon = 15.0$)
		Bilineal/M.C	1.187 ($\varepsilon = 5.0$)	1.220 ($\varepsilon = 8.0$)	1.233 ($\varepsilon = 8.0$)
	13	Afín/M.C	1.147 ($\varepsilon = 3.0$)	1.172 ($\varepsilon = 5.0$)	1.191 ($\varepsilon = 5.0$)
		Afín/M.C.T	1.147 ($\varepsilon = 3.0$)	1.182 ($\varepsilon = 3.0$)	1.192 ($\varepsilon = 5.0$)
		Similar/M.C.	1.131 ($\varepsilon = 15.0$)	1.164 ($\varepsilon = 15.0$)	1.173 ($\varepsilon = 15.0$)
		Bilineal/M.C	1.143 ($\varepsilon = 3.0$)	1.192 ($\varepsilon = 3.0$)	1.210 ($\varepsilon = 3.0$)
<i>ChocoCam</i>	7	Afín/M.C	1.139 ($\varepsilon = 3.0$)	1.169 ($\varepsilon = 3.0$)	1.174 ($\varepsilon = 3.0$)
		Afín/M.C.T	1.138 ($\varepsilon = 3.0$)	1.167 ($\varepsilon = 3.0$)	1.173 ($\varepsilon = 3.0$)
		Similar/M.C.	1.093 ($\varepsilon = 8.0$)	1.113 ($\varepsilon = 8.0$)	1.116 ($\varepsilon = 8.0$)
		Bilineal/M.C	1.102 ($\varepsilon = 5.0$)	1.162 ($\varepsilon = 3.0$)	1.137 ($\varepsilon = 5.0$)
	13	Afín/M.C	1.057 ($\varepsilon = 5.0$)	1.074 ($\varepsilon = 5.0$)	1.079 ($\varepsilon = 5.0$)
		Afín/M.C.T	1.056 ($\varepsilon = 5.0$)	1.072 ($\varepsilon = 5.0$)	1.077 ($\varepsilon = 5.0$)
		Similar/M.C.	1.056 ($\varepsilon = 8.0$)	1.070 ($\varepsilon = 8.0$)	1.072 ($\varepsilon = 8.0$)
		Bilineal/M.C	1.076 ($\varepsilon = 3.0$)	1.093 ($\varepsilon = 3.0$)	1.097 ($\varepsilon = 3.0$)

Cuadro 43: Promedio de iteraciones para estimar el modelo de transformación global con un máximo de 100 rasgos. Los resultados se agrupan considerando el máximo de iteraciones permitido, el modelo y el método de estimación utilizado, el tamaño de W y el origen de los videos.

Origen	$T(W)$	Modelo/Método	Máx. 2 Its.	Máx. 3 Its.	Máx. 4 Its.
<i>YouTube.com</i>	7	Afín/M.C.	0.008 ($\varepsilon = 8.0$)	0.012 ($\varepsilon = 8.0$)	0.013 ($\varepsilon = 8.0$)
		Afín/M.C.T	0.009 ($\varepsilon = 5.0$)	0.012 ($\varepsilon = 8.0$)	0.014 ($\varepsilon = 5.0$)
		Similar/M.C.	0.009 ($\varepsilon = 15.0$)	0.013 ($\varepsilon = 15.0$)	0.015 ($\varepsilon = 15.0$)
		Bilineal/M.C	0.009 ($\varepsilon = 5.0$)	0.012 ($\varepsilon = 8.0$)	0.013 ($\varepsilon = 8.0$)
	13	Afín/M.C	0.008 ($\varepsilon = 3.0$)	0.011 ($\varepsilon = 5.0$)	0.013 ($\varepsilon = 5.0$)
		Afín/M.C.T	0.008 ($\varepsilon = 3.0$)	0.011 ($\varepsilon = 3.0$)	0.013 ($\varepsilon = 5.0$)
		Similar/M.C.	0.007 ($\varepsilon = 15.0$)	0.010 ($\varepsilon = 15.0$)	0.011 ($\varepsilon = 15.0$)
		Bilineal/M.C	0.008 ($\varepsilon = 3.0$)	0.011 ($\varepsilon = 3.0$)	0.013 ($\varepsilon = 3.0$)
<i>ChocoCam</i>	7	Afín/M.C	0.007 ($\varepsilon = 3.0$)	0.010 ($\varepsilon = 3.0$)	0.010 ($\varepsilon = 3.0$)
		Afín/M.C.T	0.007 ($\varepsilon = 3.0$)	0.010 ($\varepsilon = 3.0$)	0.010 ($\varepsilon = 3.0$)
		Similar/M.C.	0.006 ($\varepsilon = 8.0$)	0.008 ($\varepsilon = 8.0$)	0.009 ($\varepsilon = 8.0$)
		Bilineal/M.C	0.006 ($\varepsilon = 5.0$)	0.010 ($\varepsilon = 3.0$)	0.010 ($\varepsilon = 5.0$)
	13	Afín/M.C	0.005 ($\varepsilon = 5.0$)	0.007 ($\varepsilon = 5.0$)	0.008 ($\varepsilon = 5.0$)
		Afín/M.C.T	0.005 ($\varepsilon = 5.0$)	0.007 ($\varepsilon = 5.0$)	0.008 ($\varepsilon = 5.0$)
		Similar/M.C.	0.005 ($\varepsilon = 8.0$)	0.007 ($\varepsilon = 8.0$)	0.007 ($\varepsilon = 8.0$)
		Bilineal/M.C	0.006 ($\varepsilon = 3.0$)	0.007 ($\varepsilon = 3.0$)	0.008 ($\varepsilon = 3.0$)

Cuadro 44: Error estándar del promedio de iteraciones llevadas a cabo para estimar el modelo de transformación global con un máximo de 100 rasgos. Los resultados se agrupan considerando el máximo de iteraciones permitido, el modelo y el método de estimación utilizado, el tamaño de W y el origen de los videos.

M.1.3. Porcentaje de región válida

A continuación se presenta el porcentaje de región válida, en promedio, que resulta luego de compensar las imágenes en función a un área máxima de 320×240 píxeles.

Origen	$T(W)$	Modelo/Método	Máx. 2 Its.	Máx. 3 Its.	Máx. 4 Its.
<i>YouTube.com</i>	7	Afín/M.C.	99.007 ($\varepsilon = 5.0$)	99.001 ($\varepsilon = 8.0$)	98.993 ($\varepsilon = 8.0$)
		Afín/M.C.T	99.083 ($\varepsilon = 5.0$)	99.090 ($\varepsilon = 8.0$)	99.086 ($\varepsilon = 8.0$)
		Similar/M.C.	98.873 ($\varepsilon = 15.0$)	98.884 ($\varepsilon = 15.0$)	98.881 ($\varepsilon = 15.0$)
		Bilineal/M.C	98.973 ($\varepsilon = 5.0$)	98.960 ($\varepsilon = 8.0$)	98.962 ($\varepsilon = 8.0$)
	13	Afín/M.C	98.973 ($\varepsilon = 3.0$)	98.967 ($\varepsilon = 5.0$)	98.965 ($\varepsilon = 5.0$)
		Afín/M.C.T	99.031 ($\varepsilon = 3.0$)	98.999 ($\varepsilon = 5.0$)	98.997 ($\varepsilon = 5.0$)
		Similar/M.C.	98.877 ($\varepsilon = 15.0$)	98.870 ($\varepsilon = 15.0$)	98.865 ($\varepsilon = 15.0$)
		Bilineal/M.C	98.913 ($\varepsilon = 3.0$)	98.906 ($\varepsilon = 3.0$)	98.905 ($\varepsilon = 3.0$)
<i>ChocoCam</i>	7	Afín/M.C	98.360 ($\varepsilon = 3.0$)	98.360 ($\varepsilon = 3.0$)	98.360 ($\varepsilon = 3.0$)
		Afín/M.C.T	98.495 ($\varepsilon = 3.0$)	98.496 ($\varepsilon = 3.0$)	98.495 ($\varepsilon = 3.0$)
		Similar/M.C.	98.338 ($\varepsilon = 15.0$)	98.378 ($\varepsilon = 15.0$)	98.375 ($\varepsilon = 15.0$)
		Bilineal/M.C	98.239 ($\varepsilon = 3.0$)	98.240 ($\varepsilon = 3.0$)	98.237 ($\varepsilon = 3.0$)
	13	Afín/M.C	98.208 ($\varepsilon = 5.0$)	98.216 ($\varepsilon = 5.0$)	98.220 ($\varepsilon = 5.0$)
		Afín/M.C.T	98.444 ($\varepsilon = 5.0$)	98.452 ($\varepsilon = 5.0$)	98.435 ($\varepsilon = 8.0$)
		Similar/M.C.	98.351 ($\varepsilon = 8.0$)	98.440 ($\varepsilon = 8.0$)	98.441 ($\varepsilon = 8.0$)
		Bilineal/M.C	98.310 ($\varepsilon = 5.0$)	98.373 ($\varepsilon = 3.0$)	98.322 ($\varepsilon = 5.0$)

Cuadro 45: Porcentaje de región válida promedio con un máximo de 500 rasgos. Los resultados se agrupan considerando el máximo de iteraciones permitido, el modelo y el método de estimación utilizado, el tamaño de W y el origen de los videos.

Origen	$T(W)$	Modelo/Método	Máx. 2 Its.	Máx. 3 Its.	Máx. 4 Its.
<i>YouTube.com</i>	7	Afín/M.C.	0.054 ($\varepsilon = 5.0$)	0.056 ($\varepsilon = 8.0$)	0.056 ($\varepsilon = 8.0$)
		Afín/M.C.T	0.051 ($\varepsilon = 5.0$)	0.052 ($\varepsilon = 8.0$)	0.052 ($\varepsilon = 8.0$)
		Similar/M.C.	0.062 ($\varepsilon = 15.0$)	0.061 ($\varepsilon = 15.0$)	0.061 ($\varepsilon = 15.0$)
		Bilineal/M.C	0.066 ($\varepsilon = 5.0$)	0.067 ($\varepsilon = 8.0$)	0.067 ($\varepsilon = 8.0$)
	13	Afín/M.C	0.057 ($\varepsilon = 3.0$)	0.057 ($\varepsilon = 5.0$)	0.057 ($\varepsilon = 5.0$)
		Afín/M.C.T	0.051 ($\varepsilon = 3.0$)	0.055 ($\varepsilon = 5.0$)	0.055 ($\varepsilon = 5.0$)
		Similar/M.C.	0.059 ($\varepsilon = 15.0$)	0.060 ($\varepsilon = 15.0$)	0.061 ($\varepsilon = 15.0$)
		Bilineal/M.C	0.070 ($\varepsilon = 3.0$)	0.070 ($\varepsilon = 3.0$)	0.070 ($\varepsilon = 3.0$)
<i>ChocoCam</i>	7	Afín/M.C	0.064 ($\varepsilon = 5.0$)	0.064 ($\varepsilon = 5.0$)	0.064 ($\varepsilon = 3.0$)
		Afín/M.C.T	0.043 ($\varepsilon = 3.0$)	0.042 ($\varepsilon = 3.0$)	0.042 ($\varepsilon = 3.0$)
		Similar/M.C.	0.079 ($\varepsilon = 15.0$)	0.075 ($\varepsilon = 15.0$)	0.075 ($\varepsilon = 15.0$)
		Bilineal/M.C	0.082 ($\varepsilon = 3.0$)	0.082 ($\varepsilon = 3.0$)	0.081 ($\varepsilon = 3.0$)
	13	Afín/M.C	0.088 ($\varepsilon = 5.0$)	0.088 ($\varepsilon = 5.0$)	0.087 ($\varepsilon = 5.0$)
		Afín/M.C.T	0.045 ($\varepsilon = 5.0$)	0.044 ($\varepsilon = 5.0$)	0.047 ($\varepsilon = 8.0$)
		Similar/M.C.	0.071 ($\varepsilon = 8.0$)	0.046 ($\varepsilon = 8.0$)	0.046 ($\varepsilon = 8.0$)
		Bilineal/M.C	0.071 ($\varepsilon = 5.0$)	0.065 ($\varepsilon = 3.0$)	0.069 ($\varepsilon = 5.0$)

Cuadro 46: Error estándar del promedio del porcentaje de región válida con un máximo de 500 rasgos. Los resultados se agrupan considerando el máximo de iteraciones permitido, el modelo y el método de estimación utilizado, el tamaño de W y el origen de los videos.

Origen	$T(W)$	Modelo/Método	Máx. 2 Its.	Máx. 3 Its.	Máx. 4 Its.
<i>YouTube.com</i>	7	Afín/M.C.	99.029 ($\varepsilon = 5.0$)	99.017 ($\varepsilon = 8.0$)	99.010 ($\varepsilon = 8.0$)
		Afín/M.C.T	99.079 ($\varepsilon = 5.0$)	99.104 ($\varepsilon = 8.0$)	99.101 ($\varepsilon = 8.0$)
		Similar/M.C.	98.890 ($\varepsilon = 15.0$)	98.898 ($\varepsilon = 15.0$)	98.901 ($\varepsilon = 15.0$)
		Bilineal/M.C	98.994 ($\varepsilon = 5.0$)	98.978 ($\varepsilon = 8.0$)	98.980 ($\varepsilon = 8.0$)
	13	Afín/M.C	98.986 ($\varepsilon = 5.0$)	98.983 ($\varepsilon = 5.0$)	98.982 ($\varepsilon = 5.0$)
		Afín/M.C.T	99.032 ($\varepsilon = 3.0$)	99.025 ($\varepsilon = 5.0$)	99.023 ($\varepsilon = 5.0$)
		Similar/M.C.	98.906 ($\varepsilon = 15.0$)	98.897 ($\varepsilon = 15.0$)	98.894 ($\varepsilon = 15.0$)
		Bilineal/M.C	98.924 ($\varepsilon = 3.0$)	98.916 ($\varepsilon = 3.0$)	98.916 ($\varepsilon = 3.0$)
<i>ChocoCam</i>	7	Afín/M.C	98.355 ($\varepsilon = 5.0$)	98.358 ($\varepsilon = 5.0$)	98.366 ($\varepsilon = 5.0$)
		Afín/M.C.T	98.498 ($\varepsilon = 3.0$)	98.495 ($\varepsilon = 3.0$)	98.496 ($\varepsilon = 3.0$)
		Similar/M.C.	98.334 ($\varepsilon = 15.0$)	98.416 ($\varepsilon = 15.0$)	98.419 ($\varepsilon = 15.0$)
		Bilineal/M.C	98.214 ($\varepsilon = 5.0$)	98.213 ($\varepsilon = 5.0$)	98.212 ($\varepsilon = 3.0$)
	13	Afín/M.C	98.187 ($\varepsilon = 5.0$)	98.194 ($\varepsilon = 5.0$)	98.197 ($\varepsilon = 5.0$)
		Afín/M.C.T	98.426 ($\varepsilon = 5.0$)	98.425 ($\varepsilon = 5.0$)	98.425 ($\varepsilon = 5.0$)
		Similar/M.C.	98.389 ($\varepsilon = 8.0$)	98.431 ($\varepsilon = 8.0$)	98.429 ($\varepsilon = 8.0$)
		Bilineal/M.C	98.347 ($\varepsilon = 3.0$)	98.347 ($\varepsilon = 3.0$)	98.349 ($\varepsilon = 3.0$)

Cuadro 47: Porcentaje de región válida promedio con un máximo de 300 rasgos. Los resultados se agrupan considerando el máximo de iteraciones permitido, el modelo y el método de estimación utilizado, el tamaño de W y el origen de los videos.

Origen	$T(W)$	Modelo/Método	Máx. 2 Its.	Máx. 3 Its.	Máx. 4 Its.
<i>YouTube.com</i>	7	Afín/M.C.	0.053 ($\varepsilon = 5.0$)	0.056 ($\varepsilon = 8.0$)	0.056 ($\varepsilon = 8.0$)
		Afín/M.C.T	0.053 ($\varepsilon = 5.0$)	0.052 ($\varepsilon = 8.0$)	0.052 ($\varepsilon = 8.0$)
		Similar/M.C.	0.062 ($\varepsilon = 15.0$)	0.061 ($\varepsilon = 15.0$)	0.061 ($\varepsilon = 15.0$)
		Bilineal/M.C	0.064 ($\varepsilon = 5.0$)	0.067 ($\varepsilon = 8.0$)	0.067 ($\varepsilon = 8.0$)
	13	Afín/M.C	0.058 ($\varepsilon = 5.0$)	0.058 ($\varepsilon = 5.0$)	0.058 ($\varepsilon = 5.0$)
		Afín/M.C.T	0.055 ($\varepsilon = 3.0$)	0.055 ($\varepsilon = 5.0$)	0.055 ($\varepsilon = 5.0$)
		Similar/M.C.	0.058 ($\varepsilon = 15.0$)	0.060 ($\varepsilon = 15.0$)	0.060 ($\varepsilon = 15.0$)
		Bilineal/M.C	0.071 ($\varepsilon = 3.0$)	0.071 ($\varepsilon = 3.0$)	0.071 ($\varepsilon = 3.0$)
<i>ChocoCam</i>	7	Afín/M.C	0.062 ($\varepsilon = 5.0$)	0.061 ($\varepsilon = 5.0$)	0.061 ($\varepsilon = 5.0$)
		Afín/M.C.T	0.046 ($\varepsilon = 3.0$)	0.046 ($\varepsilon = 3.0$)	0.046 ($\varepsilon = 3.0$)
		Similar/M.C.	0.078 ($\varepsilon = 15.0$)	0.059 ($\varepsilon = 15.0$)	0.059 ($\varepsilon = 15.0$)
		Bilineal/M.C	0.087 ($\varepsilon = 5.0$)	0.088 ($\varepsilon = 5.0$)	0.088 ($\varepsilon = 3.0$)
	13	Afín/M.C	0.091 ($\varepsilon = 5.0$)	0.091 ($\varepsilon = 5.0$)	0.091 ($\varepsilon = 5.0$)
		Afín/M.C.T	0.054 ($\varepsilon = 5.0$)	0.054 ($\varepsilon = 5.0$)	0.054 ($\varepsilon = 5.0$)
		Similar/M.C.	0.060 ($\varepsilon = 8.0$)	0.047 ($\varepsilon = 8.0$)	0.047 ($\varepsilon = 8.0$)
		Bilineal/M.C	0.066 ($\varepsilon = 3.0$)	0.066 ($\varepsilon = 3.0$)	0.066 ($\varepsilon = 3.0$)

Cuadro 48: Error estándar del promedio del porcentaje de región válida con un máximo de 300 rasgos. Los resultados se agrupan considerando el máximo de iteraciones permitido, el modelo y el método de estimación utilizado, el tamaño de W y el origen de los videos.

Origen	$T(W)$	Modelo/Método	Máx. 2 Its.	Máx. 3 Its.	Máx. 4 Its.
<i>YouTube.com</i>	7	Afín/M.C.	99.008 ($\varepsilon = 8.0$)	99.014 ($\varepsilon = 8.0$)	99.008 ($\varepsilon = 8.0$)
		Afín/M.C.T	99.089 ($\varepsilon = 5.0$)	99.098 ($\varepsilon = 8.0$)	99.083 ($\varepsilon = 5.0$)
		Similar/M.C.	98.932 ($\varepsilon = 15.0$)	98.931 ($\varepsilon = 15.0$)	98.932 ($\varepsilon = 15.0$)
		Bilineal/M.C	98.873 ($\varepsilon = 5.0$)	98.972 ($\varepsilon = 8.0$)	98.972 ($\varepsilon = 8.0$)
	13	Afín/M.C	98.941 ($\varepsilon = 3.0$)	98.967 ($\varepsilon = 5.0$)	98.962 ($\varepsilon = 5.0$)
		Afín/M.C.T	99.023 ($\varepsilon = 3.0$)	99.023 ($\varepsilon = 3.0$)	99.004 ($\varepsilon = 5.0$)
		Similar/M.C.	98.908 ($\varepsilon = 15.0$)	98.899 ($\varepsilon = 15.0$)	98.899 ($\varepsilon = 15.0$)
		Bilineal/M.C	98.908 ($\varepsilon = 3.0$)	98.900 ($\varepsilon = 3.0$)	98.898 ($\varepsilon = 3.0$)
<i>ChocoCam</i>	7	Afín/M.C	98.348 ($\varepsilon = 3.0$)	98.351 ($\varepsilon = 3.0$)	98.351 ($\varepsilon = 3.0$)
		Afín/M.C.T	98.432 ($\varepsilon = 3.0$)	98.431 ($\varepsilon = 3.0$)	98.433 ($\varepsilon = 3.0$)
		Similar/M.C.	98.344 ($\varepsilon = 8.0$)	98.360 ($\varepsilon = 8.0$)	98.359 ($\varepsilon = 8.0$)
		Bilineal/M.C	98.252 ($\varepsilon = 5.0$)	98.201 ($\varepsilon = 3.0$)	98.251 ($\varepsilon = 5.0$)
	13	Afín/M.C	98.169 ($\varepsilon = 5.0$)	98.170 ($\varepsilon = 5.0$)	98.169 ($\varepsilon = 5.0$)
		Afín/M.C.T	98.381 ($\varepsilon = 5.0$)	98.384 ($\varepsilon = 5.0$)	98.382 ($\varepsilon = 5.0$)
		Similar/M.C.	98.374 ($\varepsilon = 8.0$)	98.370 ($\varepsilon = 8.0$)	98.368 ($\varepsilon = 8.0$)
		Bilineal/M.C	98.295 ($\varepsilon = 3.0$)	98.296 ($\varepsilon = 3.0$)	98.297 ($\varepsilon = 3.0$)

Cuadro 49: Porcentaje de región válida promedio con un máximo de 100 rasgos. Los resultados se agrupan considerando el máximo de iteraciones permitido, el modelo y el método de estimación utilizado, el tamaño de W y el origen de los videos.

Origen	$T(W)$	Modelo/Método	Máx. 2 Its.	Máx. 3 Its.	Máx. 4 Its.
<i>YouTube.com</i>	7	Afín/M.C.	0.059 ($\varepsilon = 8.0$)	0.059 ($\varepsilon = 8.0$)	0.059 ($\varepsilon = 8.0$)
		Afín/M.C.T	0.053 ($\varepsilon = 5.0$)	0.055 ($\varepsilon = 8.0$)	0.053 ($\varepsilon = 5.0$)
		Similar/M.C.	0.064 ($\varepsilon = 15.0$)	0.064 ($\varepsilon = 15.0$)	0.064 ($\varepsilon = 15.0$)
		Bilineal/M.C	0.095 ($\varepsilon = 5.0$)	0.069 ($\varepsilon = 8.0$)	0.069 ($\varepsilon = 8.0$)
	13	Afín/M.C	0.069 ($\varepsilon = 3.0$)	0.062 ($\varepsilon = 5.0$)	0.062 ($\varepsilon = 5.0$)
		Afín/M.C.T	0.057 ($\varepsilon = 3.0$)	0.057 ($\varepsilon = 3.0$)	0.059 ($\varepsilon = 5.0$)
		Similar/M.C.	0.063 ($\varepsilon = 15.0$)	0.065 ($\varepsilon = 15.0$)	0.065 ($\varepsilon = 15.0$)
		Bilineal/M.C	0.077 ($\varepsilon = 3.0$)	0.077 ($\varepsilon = 3.0$)	0.077 ($\varepsilon = 3.0$)
<i>ChocoCam</i>	7	Afín/M.C	0.066 ($\varepsilon = 3.0$)	0.066 ($\varepsilon = 3.0$)	0.066 ($\varepsilon = 3.0$)
		Afín/M.C.T	0.063 ($\varepsilon = 3.0$)	0.062 ($\varepsilon = 3.0$)	0.062 ($\varepsilon = 3.0$)
		Similar/M.C.	0.068 ($\varepsilon = 8.0$)	0.068 ($\varepsilon = 8.0$)	0.068 ($\varepsilon = 8.0$)
		Bilineal/M.C	0.081 ($\varepsilon = 5.0$)	0.088 ($\varepsilon = 3.0$)	0.081 ($\varepsilon = 5.0$)
	13	Afín/M.C	0.104 ($\varepsilon = 5.0$)	0.104 ($\varepsilon = 5.0$)	0.104 ($\varepsilon = 5.0$)
		Afín/M.C.T	0.068 ($\varepsilon = 5.0$)	0.067 ($\varepsilon = 5.0$)	0.067 ($\varepsilon = 5.0$)
		Similar/M.C.	0.069 ($\varepsilon = 8.0$)	0.071 ($\varepsilon = 8.0$)	0.072 ($\varepsilon = 8.0$)
		Bilineal/M.C	0.079 ($\varepsilon = 3.0$)	0.079 ($\varepsilon = 3.0$)	0.079 ($\varepsilon = 3.0$)

Cuadro 50: Error estándar del promedio del porcentaje de región válida con un máximo de 100 rasgos. Los resultados se agrupan considerando el máximo de iteraciones permitido, el modelo y el método de estimación utilizado, el tamaño de W y el origen de los videos.

M.2. Detalles sobre resultados del Experimento V

Para llevar a cabo las pruebas del Experimento V, sección 4.2.2, se consideraron los siguientes valores para ε en función de los resultados del Experimento IV:

Máx. #Rasgos	Modelo/Método	$T(W_e) = 7$	$T(W_e) = 13$
500	Afín/M.C.	3.0	5.0
	Afín/M.C.T.	3.0	8.0
	Similar/M.C.	15.0	8.0
	Bilineal/M.C.	3.0	5.0
100	Afín/M.C.	3.0	5.0
	Afín/M.C.T.	3.0	5.0
	Similar/M.C.	8.0	8.0
	Bilineal/M.C.	5.0	3.0

Cuadro 51: Valor de ε en el Experimento V.

Los resultados obtenidos en cuanto al número aproximado, en promedio, de imágenes procesadas por segundo se detallan a continuación:

Máx. #Rasgos	Modelo/Método	$T(W_e) = 7$		$T(W_e) = 13$	
		$T(W_s) = 3$	$T(W_s) = T(W_e)$	$T(W_s) = 3$	$T(W_s) = T(W_e)$
100	Afín/M.C.	20.219 (0.204)	20.682 (0.186)	16.000 (0.250)	15.344 (0.239)
	Afín/M.C.T.	20.344 (0.179)	20.351 (0.162)	15.742 (0.276)	15.490 (0.269)
	Similar/M.C.	20.179 (0.162)	20.159 (0.179)	15.563 (0.241)	15.444 (0.244)
	Bilineal/M.C.	20.417 (0.209)	20.046 (0.187)	15.278 (0.253)	15.510 (0.242)
500	Afín/M.C.	15.596 (0.465)	14.914 (0.460)	9.020 (0.524)	8.013 (0.470)
	Afín/M.C.T.	15.305 (0.445)	15.073 (0.465)	9.318 (0.523)	8.219 (0.482)
	Similar/M.C.	15.219 (0.487)	14.616 (0.457)	9.397 (0.554)	8.046 (0.467)
	Bilineal/M.C.	14.285 (0.459)	14.185 (0.464)	9.371 (0.544)	8.113 (0.490)

Cuadro 52: Promedio del aproximado número de imágenes procesadas por segundo en la secuencia 1 del Experimento V. Entre paréntesis se muestra el error estándar de la media según el caso.

Máx. #Rasgos	Modelo/Método	$T(W_e) = 7$		$T(W_e) = 13$	
		$T(W_s) = 3$	$T(W_s) = T(W_e)$	$T(W_s) = 3$	$T(W_s) = T(W_e)$
100	Afín/M.C.	20.464 (0.162)	21.424 (0.160)	21.987 (0.154)	21.119 (0.166)
	Afín/M.C.T.	21.629 (0.382)	22.285 (0.350)	15.967 (0.265)	16.026 (0.324)
	Similar/M.C.	20.417 (0.143)	28.755 (0.182)	15.682 (0.330)	16.536 (0.219)
	Bilineal/M.C.	20.093 (0.282)	22.411 (0.392)	15.344 (0.261)	15.596 (0.222)
500	Afín/M.C.	11.358 (0.191)	17.020 (0.177)	5.477 (0.215)	5.040 (0.223)
	Afín/M.C.T.	11.556 (0.202)	14.450 (0.249)	6.185 (0.206)	6.954 (0.176)
	Similar/M.C.	11.086 (0.193)	16.689 (0.167)	6.338 (0.216)	6.781 (0.202)
	Bilineal/M.C.	10.781 (0.192)	11.728 (0.184)	6.126 (0.230)	6.927 (0.207)

Cuadro 53: Promedio del aproximado número de imágenes procesadas por segundo en la secuencia 2 del Experimento V. Entre paréntesis se muestra el error estándar de la media según el caso.

Máx. #Rasgos	Modelo/Método	$T(W_e) = 7$		$T(W_e) = 13$	
		$T(W_s) = 3$	$T(W_s) = T(W_e)$	$T(W_s) = 3$	$T(W_s) = T(W_e)$
100	Afín/M.C.	21.867 (0.355)	28.901 (0.489)	17.311 (0.368)	23.144 (0.435)
	Afín/M.C.T.	21.418 (0.329)	28.934 (0.400)	16.600 (0.335)	19.913 (0.441)
	Similar/M.C.	22.077 (0.269)	28.609 (0.473)	16.725 (0.391)	18.859 (0.426)
	Bilineal/M.C.	21.088 (0.397)	26.813 (0.642)	17.385 (0.354)	20.077 (0.471)
500	Afín/M.C.	15.407 (0.455)	14.209 (0.429)	7.835 (0.488)	10.815 (0.473)
	Afín/M.C.T.	15.791 (0.423)	14.078 (0.439)	8.044 (0.508)	11.511 (0.428)
	Similar/M.C.	14.717 (0.455)	13.989 (0.439)	8.120 (0.536)	10.484 (0.498)
	Bilineal/M.C.	13.630 (0.459)	13.956 (0.433)	8.802 (0.589)	10.011 (0.478)

Cuadro 54: Promedio del aproximado número de imágenes procesadas por segundo en la secuencia 3 del Experimento V. Entre paréntesis se muestra el error estándar de la media según el caso.

Máx. #Rasgos	Modelo/Método	$T(W_e) = 7$		$T(W_e) = 13$	
		$T(W_s) = 3$	$T(W_s) = T(W_e)$	$T(W_s) = 3$	$T(W_s) = T(W_e)$
100	Afín/M.C.	21.867 (0.355)	28.901 (0.489)	17.311 (0.368)	23.144 (0.435)
	Afín/M.C.T.	21.418 (0.329)	28.934 (0.400)	16.600 (0.335)	19.913 (0.441)
	Similar/M.C.	22.077 (0.269)	28.609 (0.473)	16.725 (0.391)	18.859 (0.426)
	Bilineal/M.C.	21.088 (0.397)	26.813 (0.642)	17.385 (0.354)	20.077 (0.471)
500	Afín/M.C.	15.407 (0.455)	14.209 (0.429)	7.835 (0.488)	10.815 (0.473)
	Afín/M.C.T.	15.791 (0.423)	14.078 (0.439)	8.044 (0.508)	11.511 (0.428)
	Similar/M.C.	14.717 (0.455)	13.989 (0.439)	8.120 (0.536)	10.484 (0.498)
	Bilineal/M.C.	13.630 (0.459)	13.956 (0.433)	8.802 (0.589)	10.011 (0.478)

Cuadro 55: Promedio del aproximado número de imágenes procesadas por segundo en la secuencia 4 del Experimento V. Entre paréntesis se muestra el error estándar de la media según el caso.

Máx. #Rasgos	Modelo/Método	$T(W_e) = 7$		$T(W_e) = 13$	
		$T(W_s) = 3$	$T(W_s) = T(W_e)$	$T(W_s) = 3$	$T(W_s) = T(W_e)$
100	Afín/M.C.	20.911 (0.129)	24.004 (0.230)	18.365 (0.176)	20.066 (0.210)
	Afín/M.C.T.	21.149 (0.160)	24.182 (0.235)	16.133 (0.150)	17.331 (0.199)
	Similar/M.C.	20.967 (0.106)	26.029 (0.233)	16.037 (0.166)	17.076 (0.167)
	Bilineal/M.C.	20.568 (0.153)	23.329 (0.254)	16.091 (0.154)	17.254 (0.190)
500	Afín/M.C.	14.202 (0.217)	15.306 (0.199)	7.469 (0.229)	8.150 (0.232)
	Afín/M.C.T.	14.318 (0.208)	14.506 (0.203)	7.862 (0.228)	9.052 (0.216)
	Similar/M.C.	13.745 (0.219)	15.023 (0.198)	7.963 (0.240)	8.568 (0.219)
	Bilineal/M.C.	12.949 (0.209)	13.333 (0.200)	8.145 (0.250)	8.457 (0.217)

Cuadro 56: Promedio del aproximado número de imágenes procesadas por segundo en total para las secuencias consideradas en el Experimento V. Entre paréntesis se muestra el error estándar de la media según el caso.

M.3. Detalles sobre resultados del Experimento VI

Para llevar a cabo las pruebas del Experimento VI, sección 4.2.3, se consideraron los siguientes valores para ε , según se trata de la secuencia extraída de *YouTube.com* o de aquella capturada desde el helicóptero del GIA, en función de los resultados del Experimento IV:

Máx. #Rasgos	Modelo/Método	<i>YouTube.com</i>	<i>ChocoCam</i>
100	Afín/M.C.	8.0	3.0
	Afín/M.C.T.	5.0	3.0
	Similar/M.C.	15.0	8.0
	Bilineal/M.C.	8.0	5.0

Cuadro 57: Valor de ε en el Experimento VI.

M.3.1. Detalles sobre secuencia de *YouTube.com*

En esta sección se muestran un grupo de imágenes del video procesado sin filtrar la transformación que modela el movimiento global, así como suprimiendo aquellas que se suponen erróneas. Para ellas se estima un modelo afín utilizando mínimos cuadrados. Se puede observar que para el grupo de imágenes que se muestran, resulta mejor no filtrar la transformación. Cuando se suprimen algunas, el movimiento vibratorio que se evidencia en la secuencia hace que la predicción según la transformación anterior aumente su magnitud, en vez de reducirla. Por lo tanto, los parámetros elegidos para clasificar las transformaciones parecen no ajustarse a las características de la secuencia y a este modelo.

A continuación también se presenta el acumulado del ángulo de rotación y los desplazamientos estimados en grupo de imágenes de la secuencia, así como la compensación de los mismos utilizando vecindades de 7 y 13 transformaciones, respectivamente.



Figura 23: Secuencia de imágenes sin filtrar la transformación en video de *YouTube.com*. Se presentan ordenadas de izquierda a derecha y de arriba a abajo. Se utiliza un modelo afín bajo mínimos cuadrados para estimar las transformaciones que modelan el movimiento global.



Figura 24: Secuencia de imágenes filtrando la transformación en video de *YouTube.com*. Se presentan ordenadas de izquierda a derecha y de arriba a abajo. Las imágenes que poseen un punto azul en la esquina superior izquierda son aquellas para las cuales se considera la transformación anterior al estimar la componente intencional del movimiento percibido entre pares. Se utiliza un modelo afín bajo mínimos cuadrados para estimar las transformaciones que modelan el movimiento global.

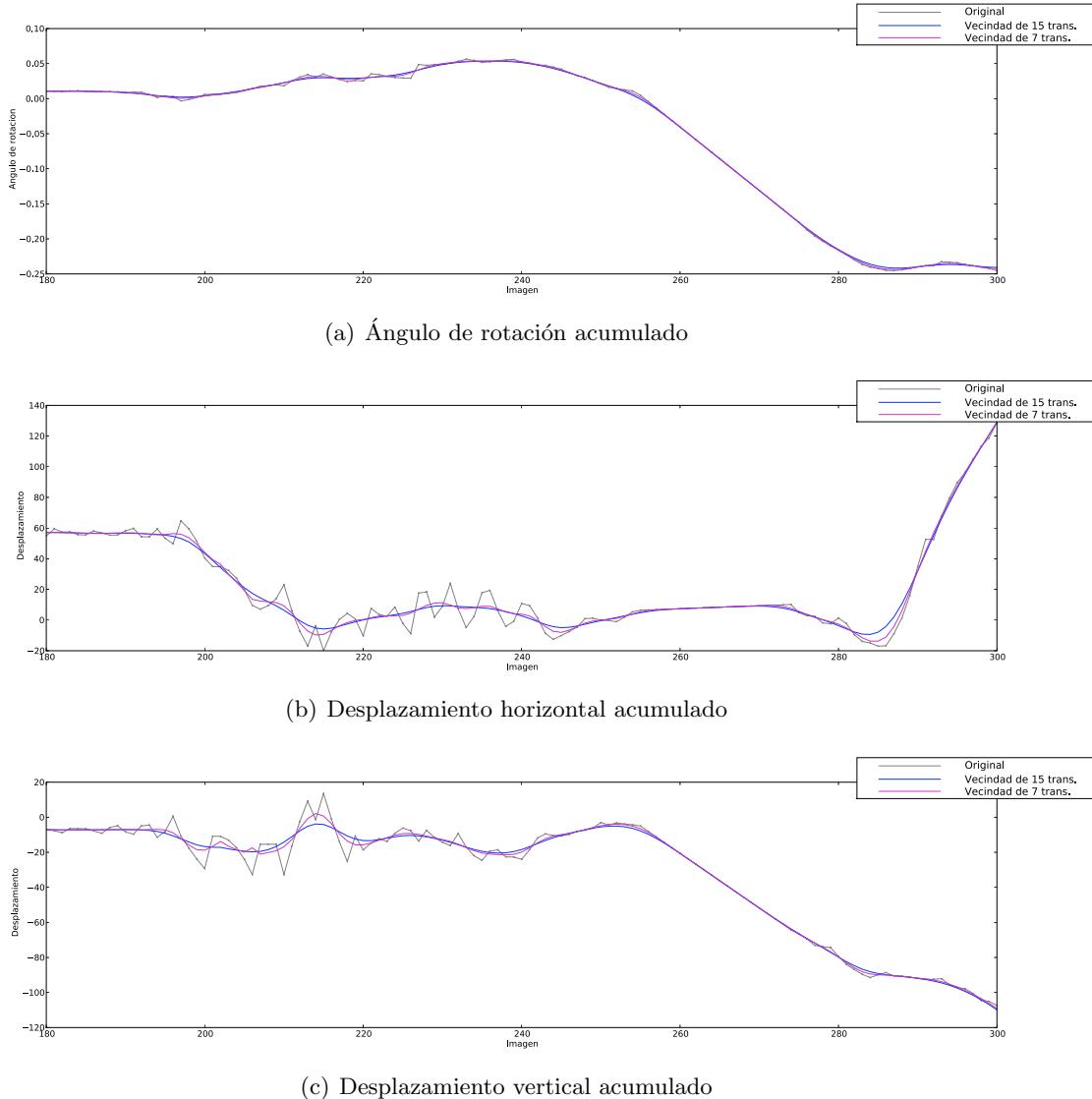


Figura 25: Estimación del movimiento acumulado y su componente intencional bajo un modelo afín por mínimos cuadrados, considerando las imágenes que van desde la 180 hasta la 300 de la secuencia de *YouTube.com* procesada en el Experimento VI. La línea morada representa la estimación de la componente intencional con una vecindad de 7 transformaciones, mientras que la azul muestra el resultado utilizando 15. la línea gris describe el movimiento global acumulado.

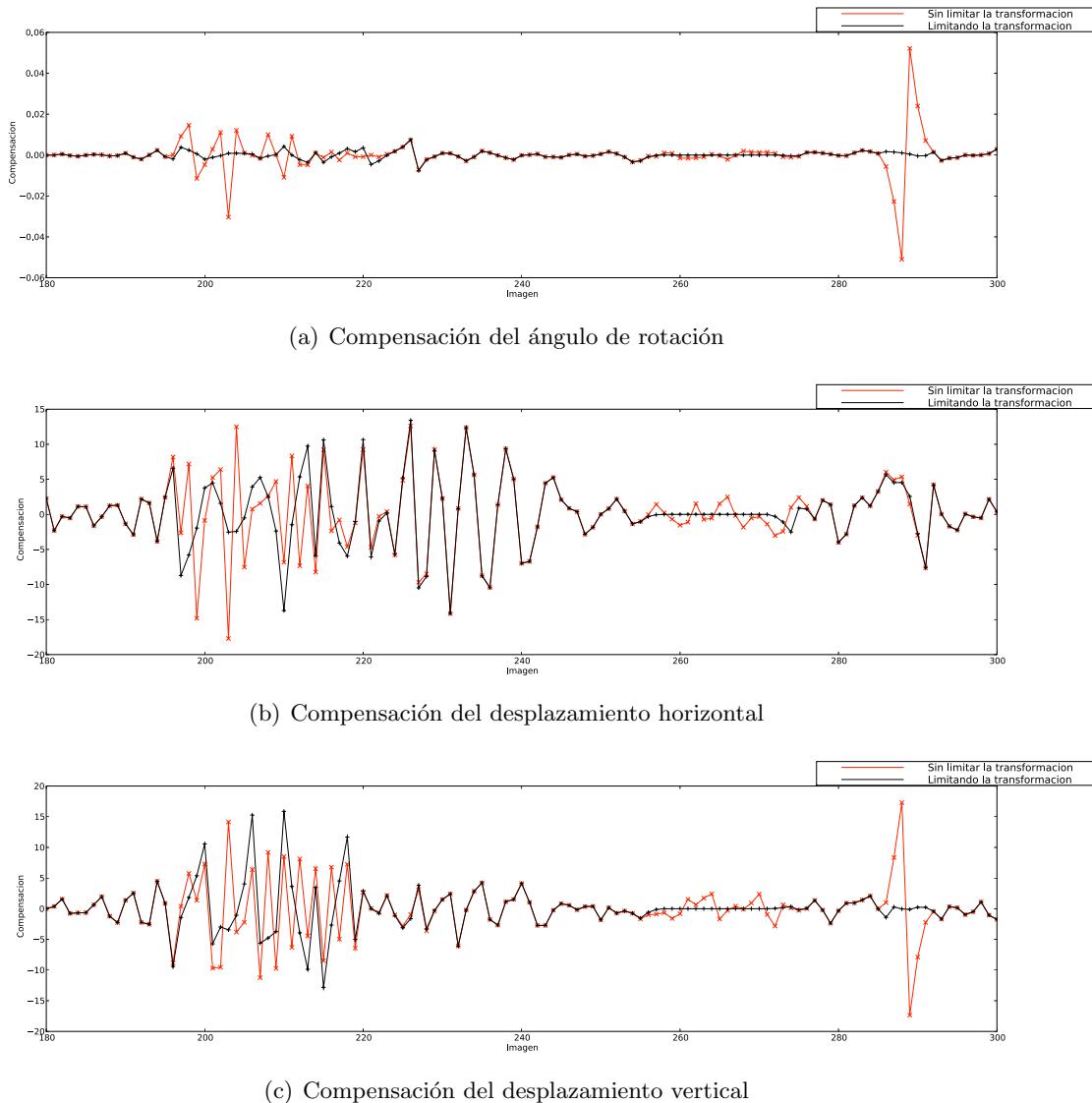


Figura 26: Magnitud de la compensación según el estimado de movimiento global bajo un modelo afín por mínimos cuadrados, considerando las imágenes que van desde la 180 hasta la 300 de la secuencia de *YouTube.com* procesada en el Experimento VI. Se considera tanto el caso en que se permite libremente cualquier tipo de transformación para ser compensada (línea roja), así como en el que se descartan aquellas que posiblemente no representan el movimiento global (línea negra). Los resultados son obtenidos para una vecindad de 7 transformaciones, con un máximo de 100 rasgos “buenos” que pueden seleccionarse y W de 7×7 píxeles.

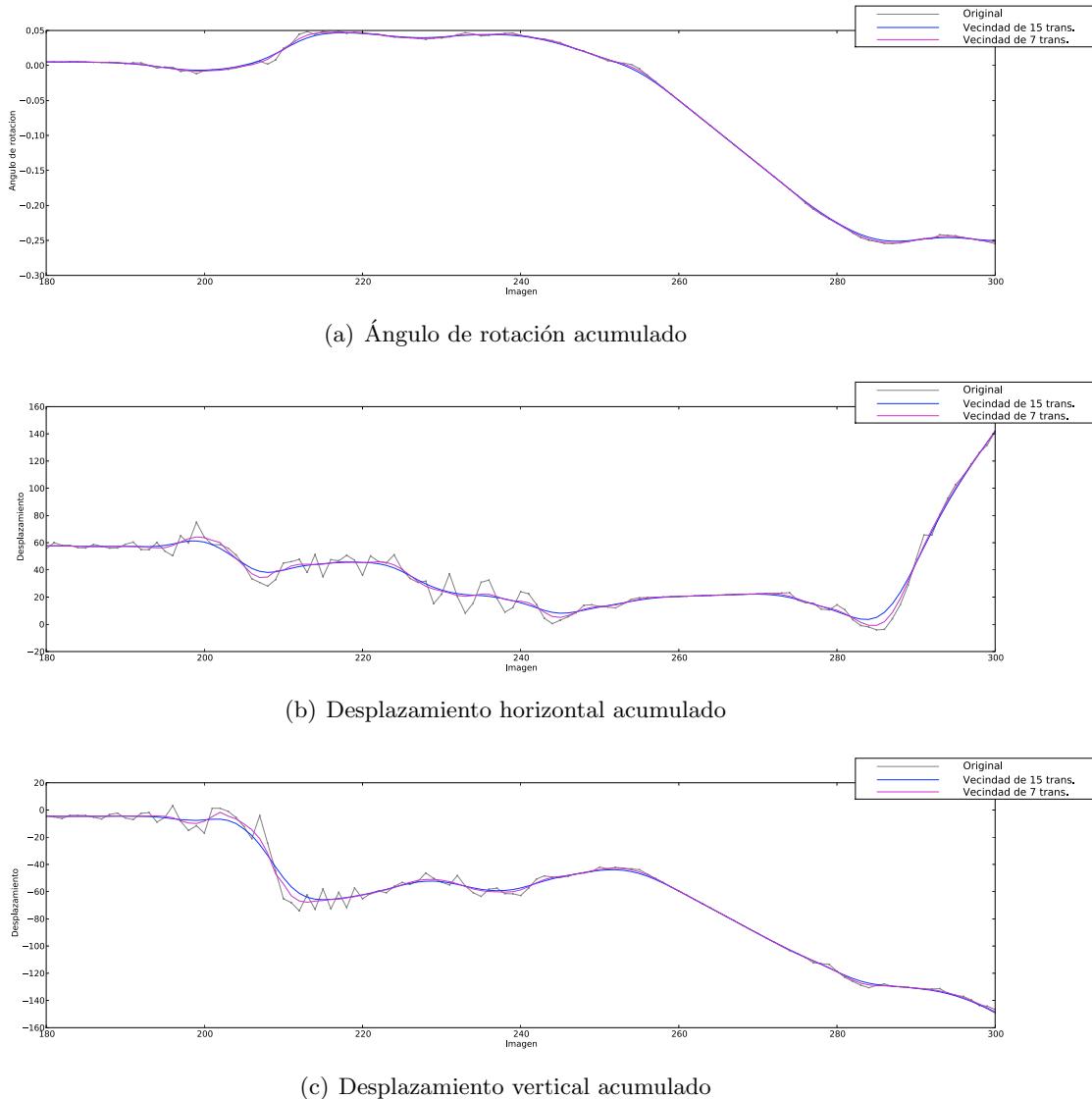


Figura 27: Estimación del movimiento acumulado y su componente intencional bajo un modelo afín por mínimos cuadrados totales, considerando las imágenes que van desde la 180 hasta la 300 de la secuencia de *YouTube.com* procesada en el Experimento VI. La línea morada representa la estimación de la componente intencional con una vecindad de 7 transformaciones, mientras que la azul muestra el resultado utilizando 15. la línea gris describe el movimiento global acumulado.

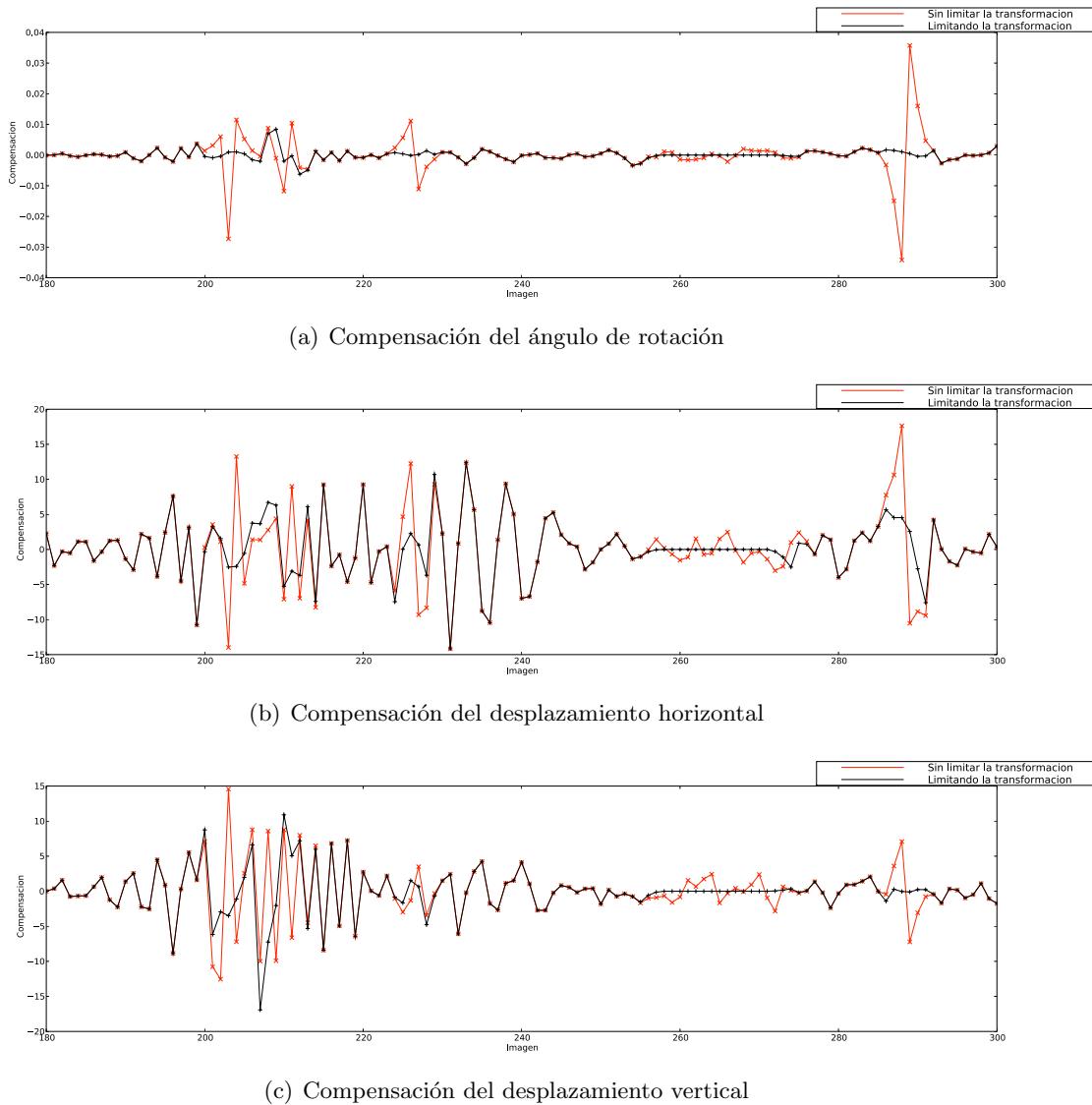


Figura 28: Magnitud de la compensación según el estimado de movimiento global bajo un modelo afín por mínimos cuadrados totales, considerando las imágenes que van desde la 180 hasta la 300 de la secuencia de *YouTube.com* procesada en el Experimento VI. Se considera tanto el caso en que se permite libremente cualquier tipo de transformación para ser compensada (línea roja), así como en el que se descartan aquellas que posiblemente no representan el movimiento global (línea negra). Los resultados son obtenidos para una vecindad de 7 transformaciones, con un máximo de 100 rasgos “buenos” que pueden seleccionarse y W de 7×7 píxeles.

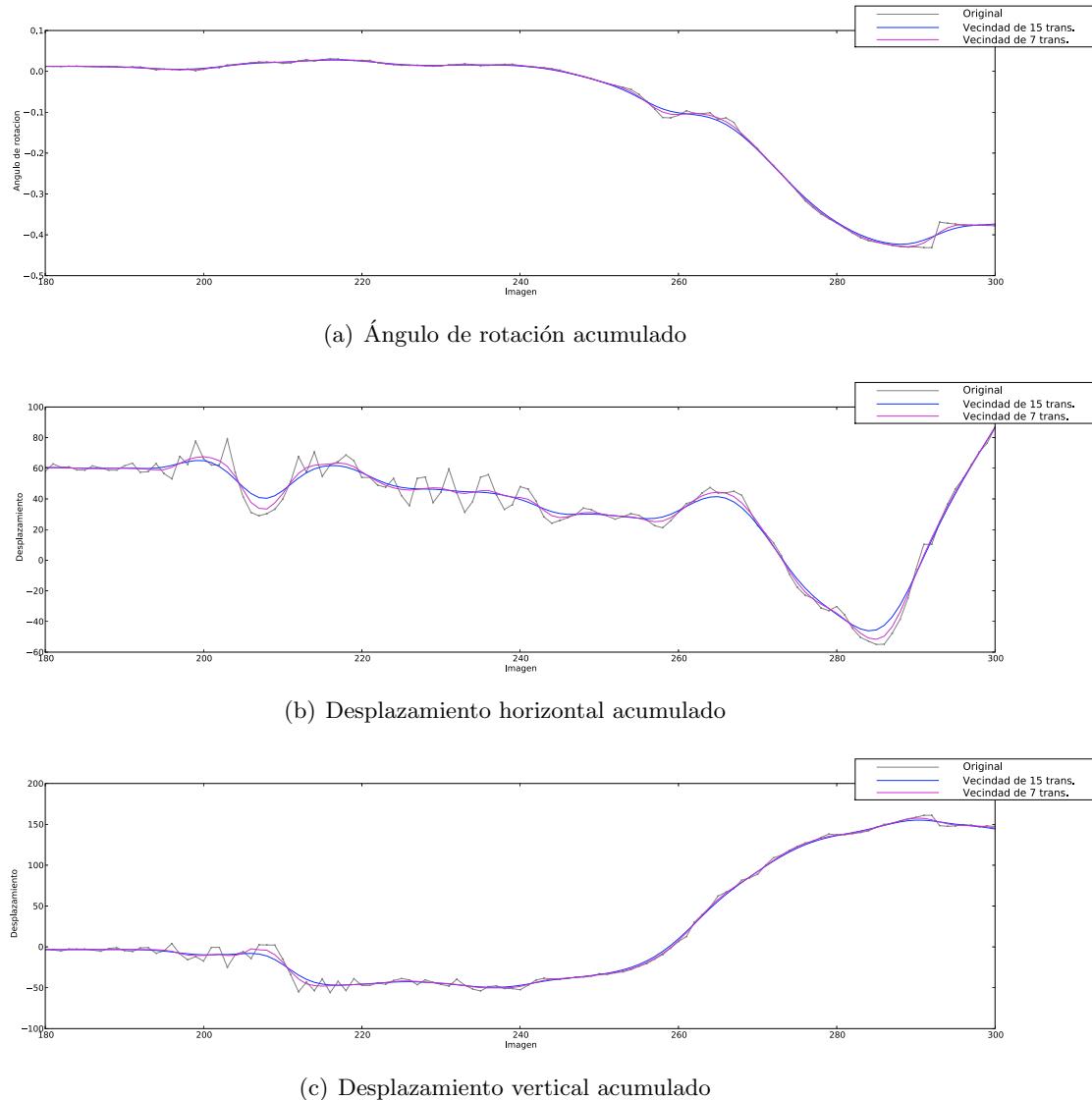


Figura 29: Estimación del movimiento acumulado y su componente intencional bajo un modelo similar por mínimos cuadrados, considerando las imágenes que van desde la 180 hasta la 300 de la secuencia de *YouTube.com* procesada en el Experimento VI. La línea morada representa la estimación de la componente intencional con una vecindad de 7 transformaciones, mientras que la azul muestra el resultado utilizando 15. la línea gris describe el movimiento global acumulado.

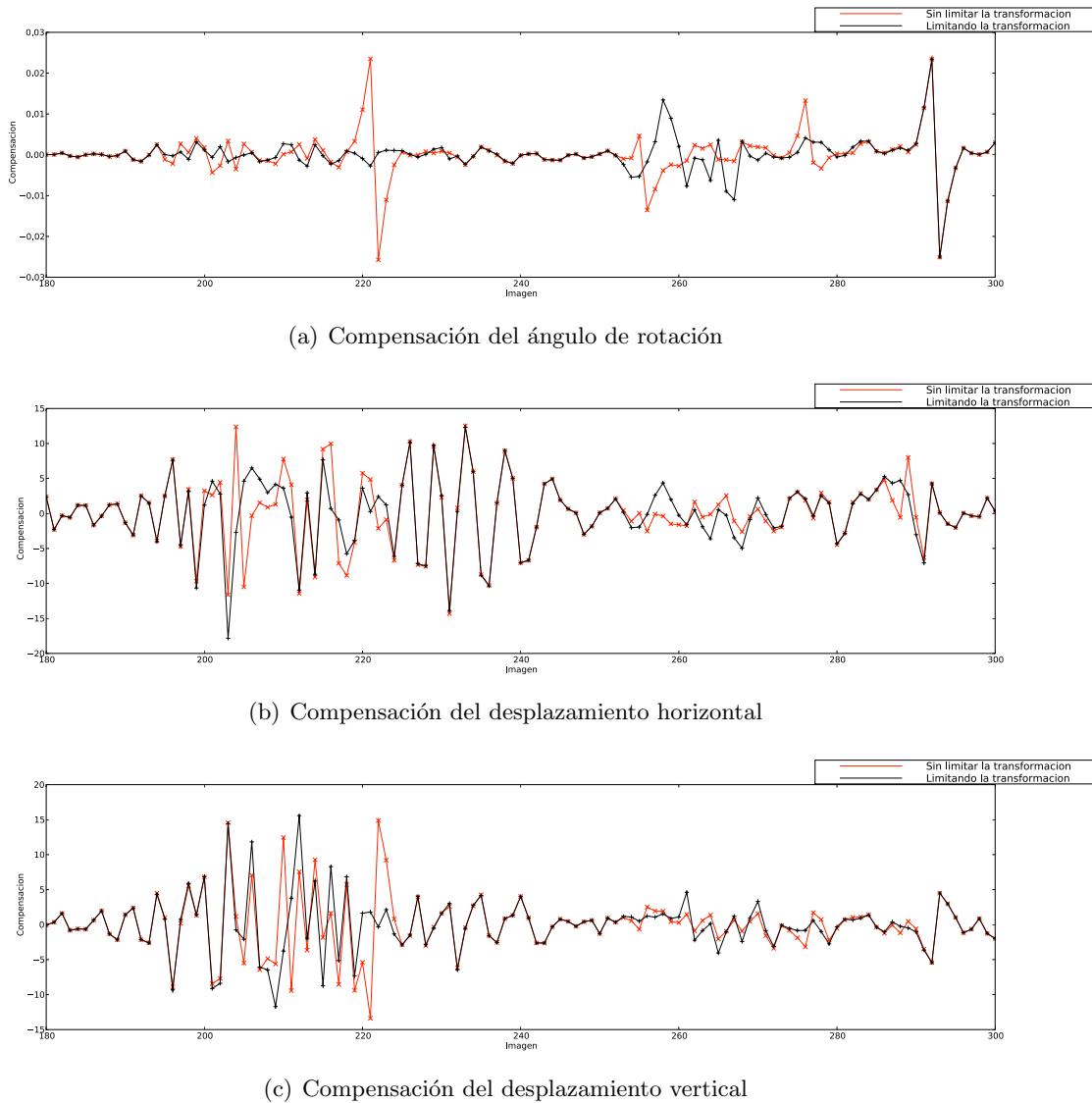


Figura 30: Magnitud de la compensación según el estimado de movimiento global bajo un modelo similar por mínimos cuadrados, considerando las imágenes que van desde la 180 hasta la 300 de la secuencia de *YouTube.com* procesada en el Experimento VI. Se considera tanto el caso en que se permite libremente cualquier tipo de transformación para ser compensada (línea roja), así como en el que se descartan aquellas que posiblemente no representan el movimiento global (línea negra). Los resultados son obtenidos para una vecindad de 7 transformaciones, con un máximo de 100 rasgos “buenos” que pueden seleccionarse y W de 7×7 píxeles.

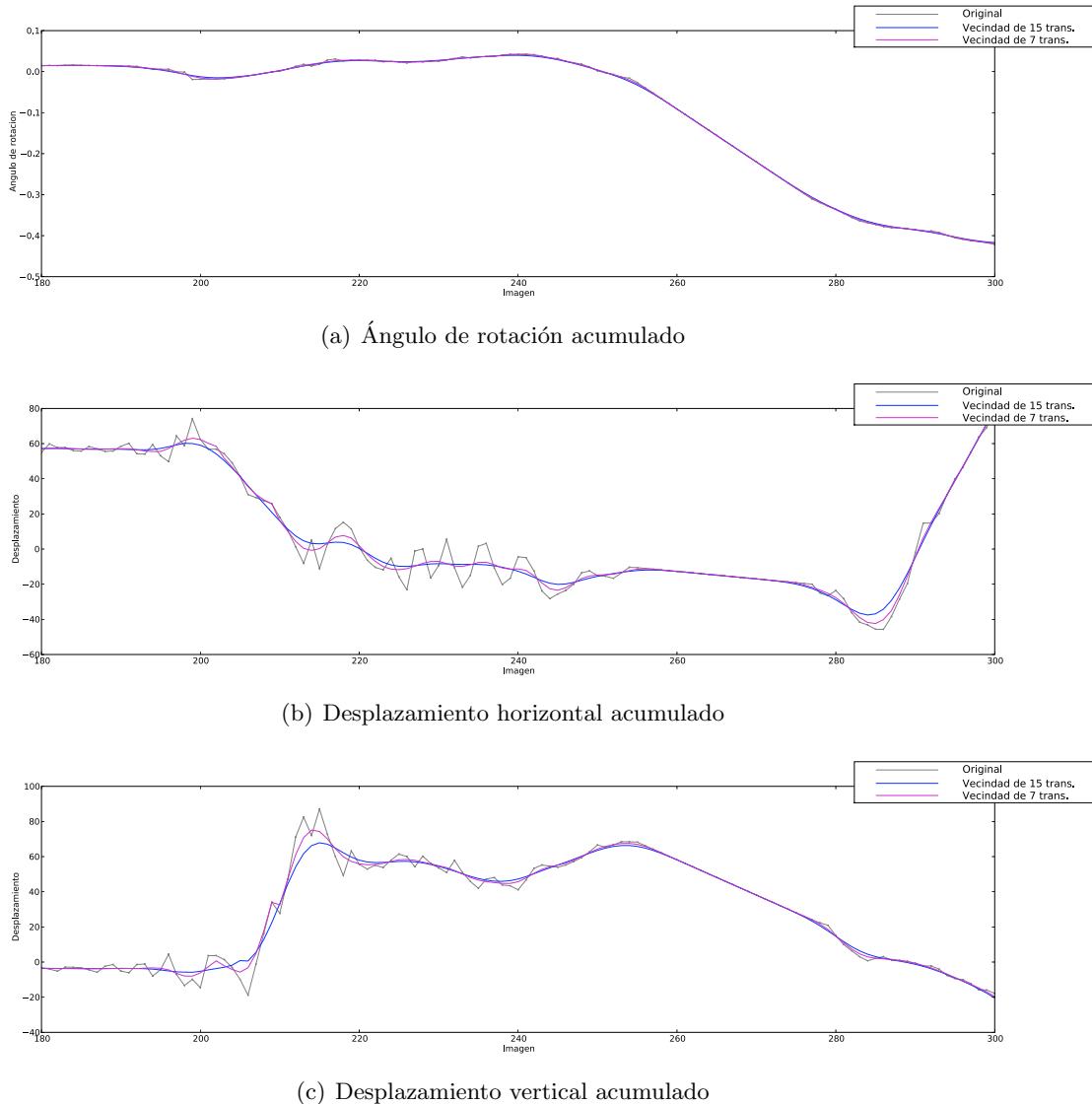


Figura 31: Estimación del movimiento acumulado y su componente intencional bajo un modelo bilineal por mínimos cuadrados, considerando las imágenes que van desde la 180 hasta la 300 de la secuencia de *YouTube.com* procesada en el Experimento VI. La línea morada representa la estimación de la componente intencional con una vecindad de 7 transformaciones, mientras que la azul muestra el resultado utilizando 15. la línea gris describe el movimiento global acumulado.

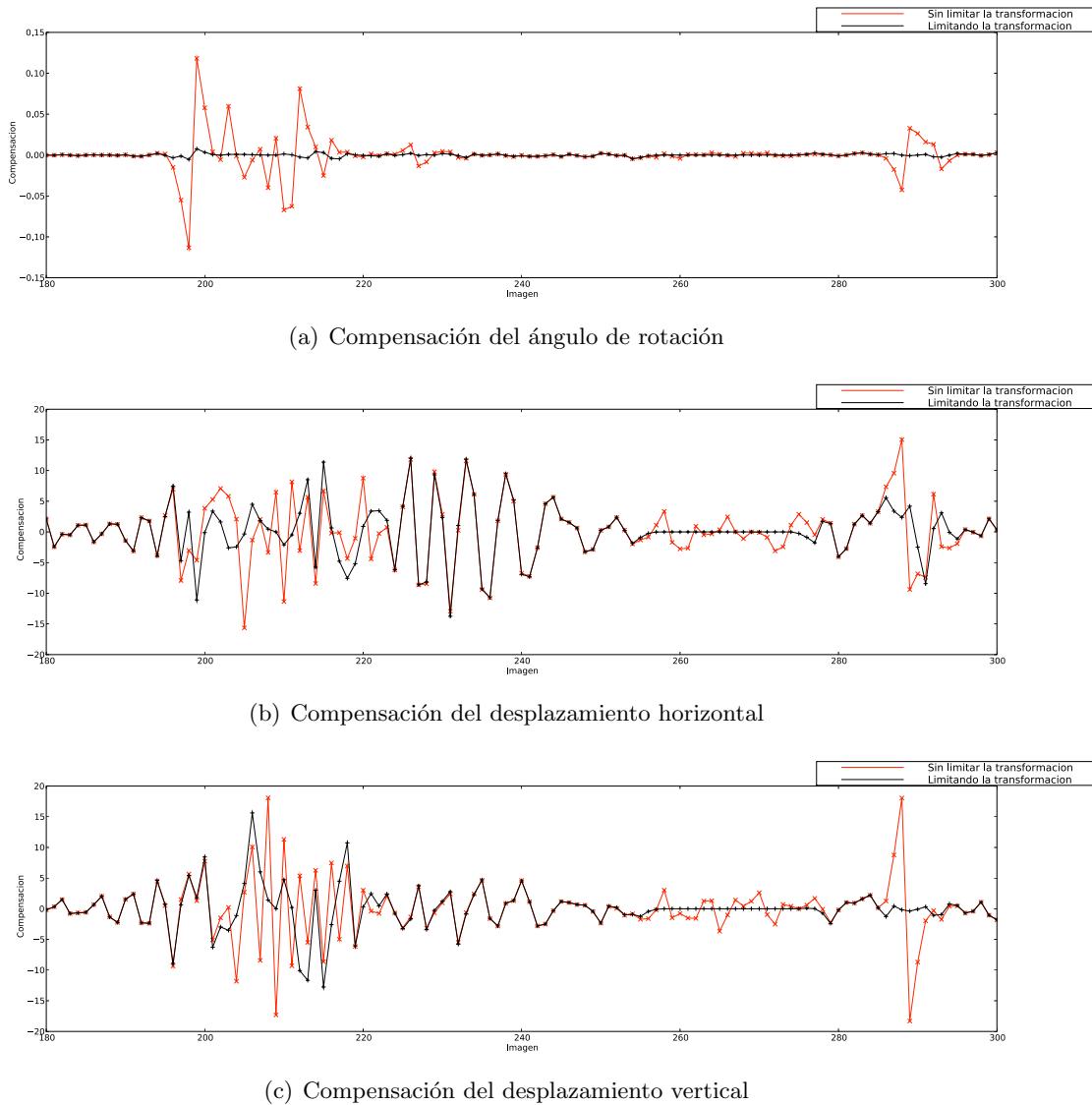


Figura 32: Magnitud de la compensación según el estimado de movimiento global bajo un modelo bilineal por mínimos cuadrados, considerando las imágenes que van desde la 180 hasta la 300 de la secuencia de *YouTube.com* procesada en el Experimento VI. Se considera tanto el caso en que se permite libremente cualquier tipo de transformación para ser compensada (línea roja), así como en el que se descartan aquellas que posiblemente no representan el movimiento global (línea negra). Los resultados son obtenidos para una vecindad de 7 transformaciones, con un máximo de 100 rasgos “buenos” que pueden seleccionarse y W de 7×7 píxeles.

M.3.2. Detalles sobre secuencia capturada desde ChocoLate

En esta sección se muestra un grupo de 15 imágenes del video capturado desde ChocoLate que se estudia en el Experimento VI, sin filtrar la transformación que modela el movimiento global, así como suprimiendo aquellas que se suponen erróneas.

En este caso se consideran una transformación afín, una similar y una bilineal por mínimos cuadrados, así como una afín por mínimos cuadrados totales para el conjunto de imágenes presentadas. En todos los casos se puede observar que la presencia de ruido producto de la transmisión analógica de las imágenes degrada severamente su contenido. En vista de esto, se estima una transformación que poco explica el movimiento global de las imágenes y al compensar la secuencia para estabilizarla, más bien se degrada su calidad.

De estas imágenes en particular, al considerar todos los modelos propuestos para la implementación de la aplicación de estabilización, se puede apreciar que el bilineal es el más afectado por el ruido.

Al comparar la estimación de un modelo afín por medio de mínimos cuadrados y la obtenida utilizando mínimos cuadrados totales, la segunda parece ser más estable para este conjunto de imágenes.

Al igual que antes, también se presenta el acumulado del ángulo de rotación y los desplazamientos estimados a lo largo de 120 imágenes que forman parte de este mismo video, así como la compensación de éstas utilizando vecindades de 7 y 13 transformaciones, respectivamente.

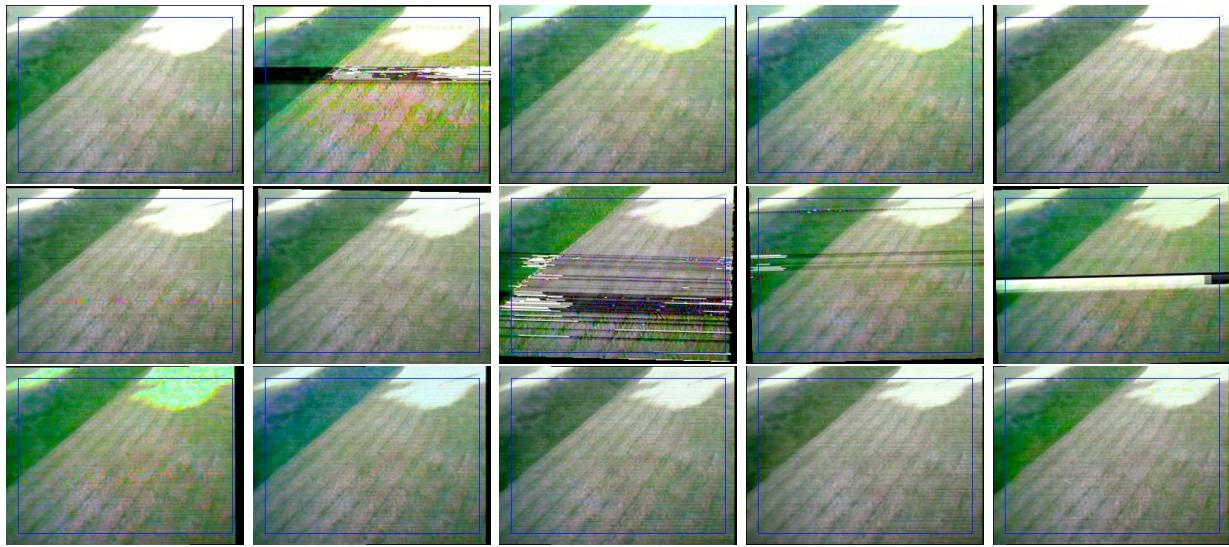


Figura 33: Secuencia de imágenes sin filtrar las transformaciones en video capturado desde Chocolate. Se presentan ordenadas de izquierda a derecha y de arriba a abajo. Se utiliza un modelo afín bajo mínimos cuadrados para estimar las transformaciones que modelan el movimiento global.

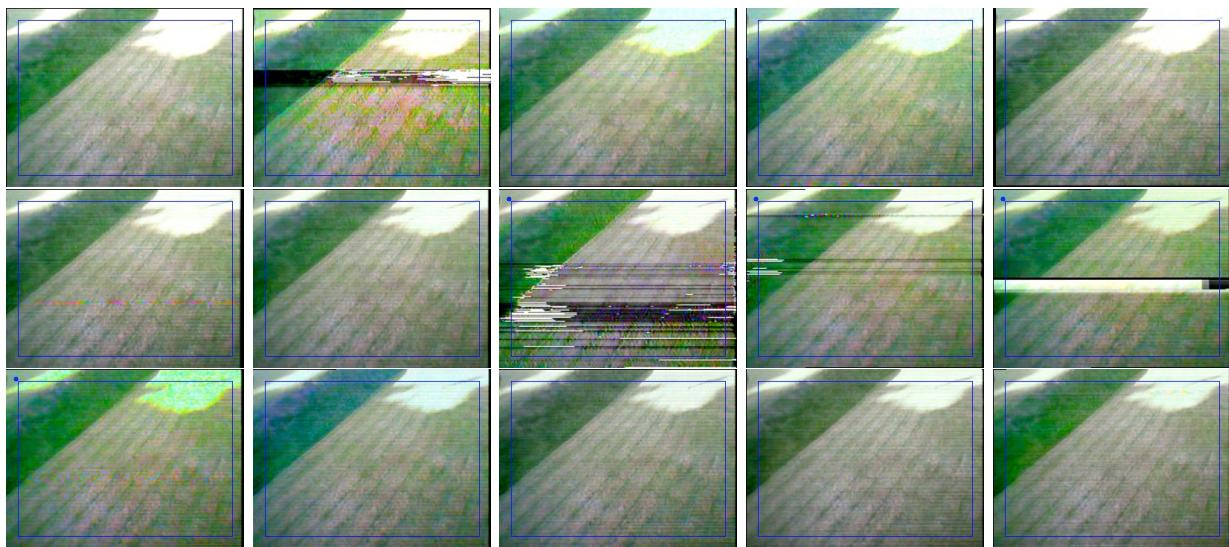


Figura 34: Secuencia de imágenes filtrando las transformaciones en video capturado desde Chocolate. Se presentan ordenadas de izquierda a derecha y de arriba a abajo. Las imágenes que poseen un punto azul en la esquina superior izquierda son aquellas para las cuales se considera la transformación anterior al estimar la componente intencional del movimiento percibido entre pares. Se utiliza un modelo afín bajo mínimos cuadrados para estimar las transformaciones que modelan el movimiento global.

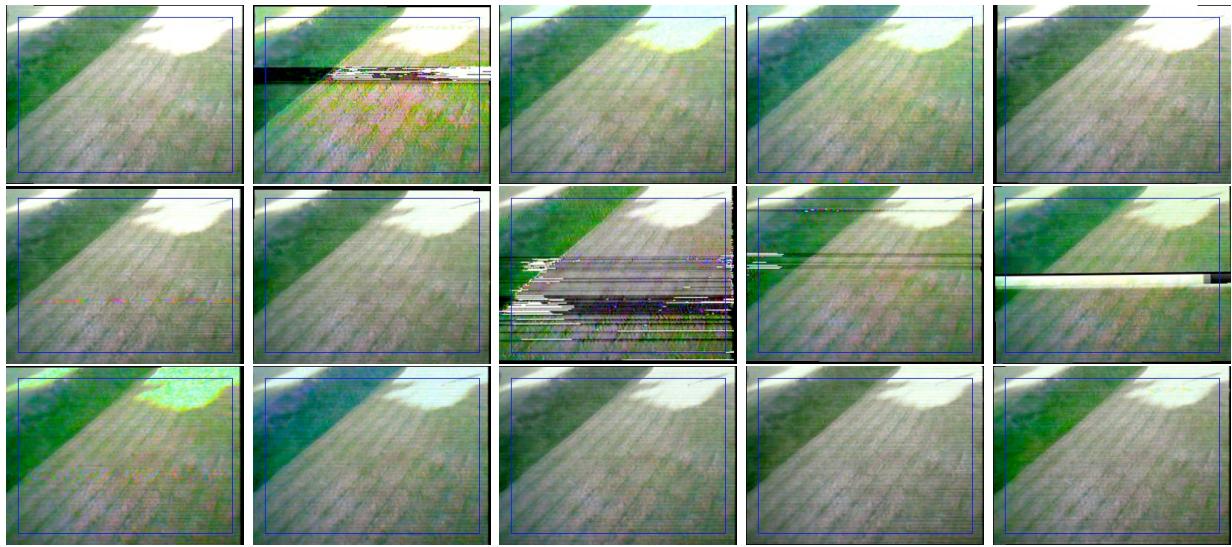


Figura 35: Secuencia de imágenes sin filtrar las transformaciones en video capturado desde ChocoLate. Se presentan ordenadas de izquierda a derecha y de arriba a abajo. Se utiliza un modelo afín bajo mínimos cuadrados totales para estimar las transformaciones que modelan el movimiento global.

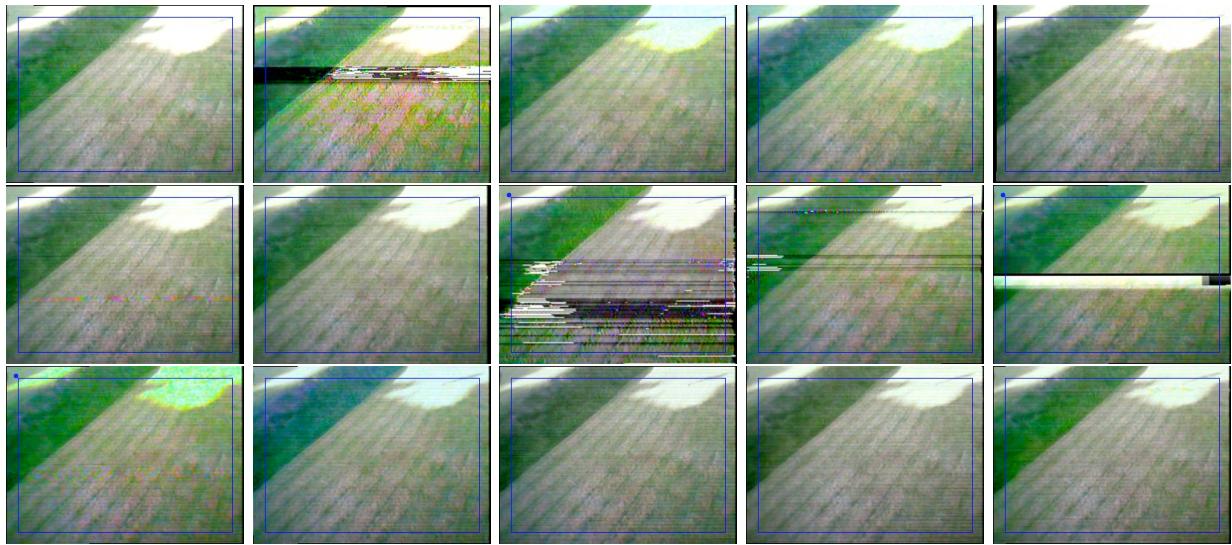


Figura 36: Secuencia de imágenes filtrando las transformaciones en video capturado desde ChocoLate. Se presentan ordenadas de izquierda a derecha y de arriba a abajo. Las imágenes que poseen un punto azul en la esquina superior izquierda son aquellas para las cuales se considera la transformación anterior al estimar la componente intencional del movimiento percibido entre pares. Se utiliza un modelo afín bajo mínimos cuadrados totales para estimar las transformaciones que modelan el movimiento global.

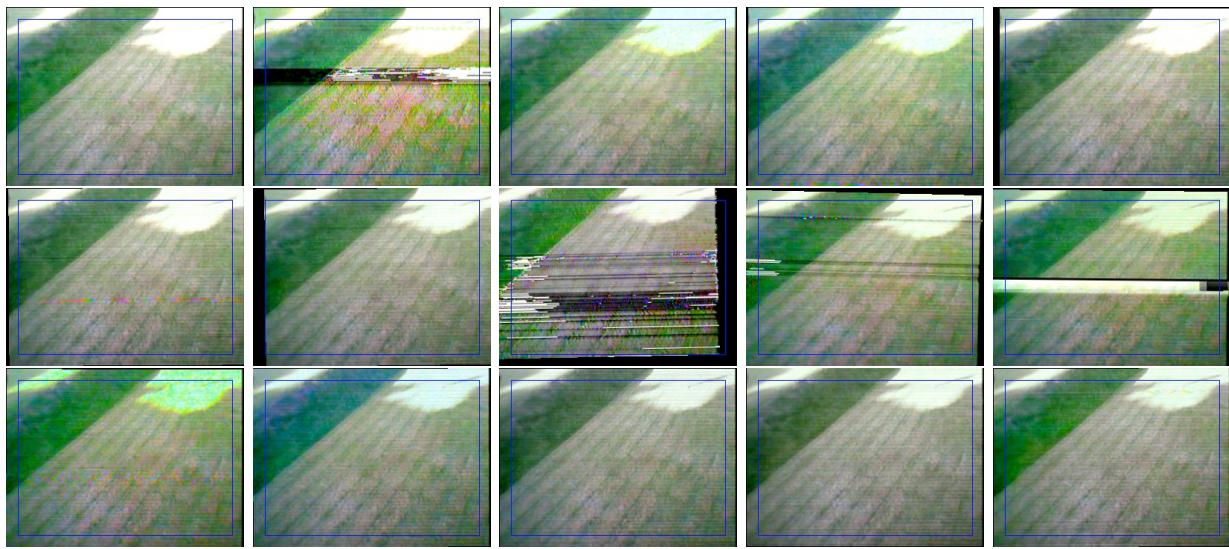


Figura 37: Secuencia de imágenes sin filtrar las transformaciones en video capturado desde ChocoLate. Se presentan ordenadas de izquierda a derecha y de arriba a abajo. Se utiliza un modelo similar bajo mínimos cuadrados para estimar las transformaciones que modelan el movimiento global.

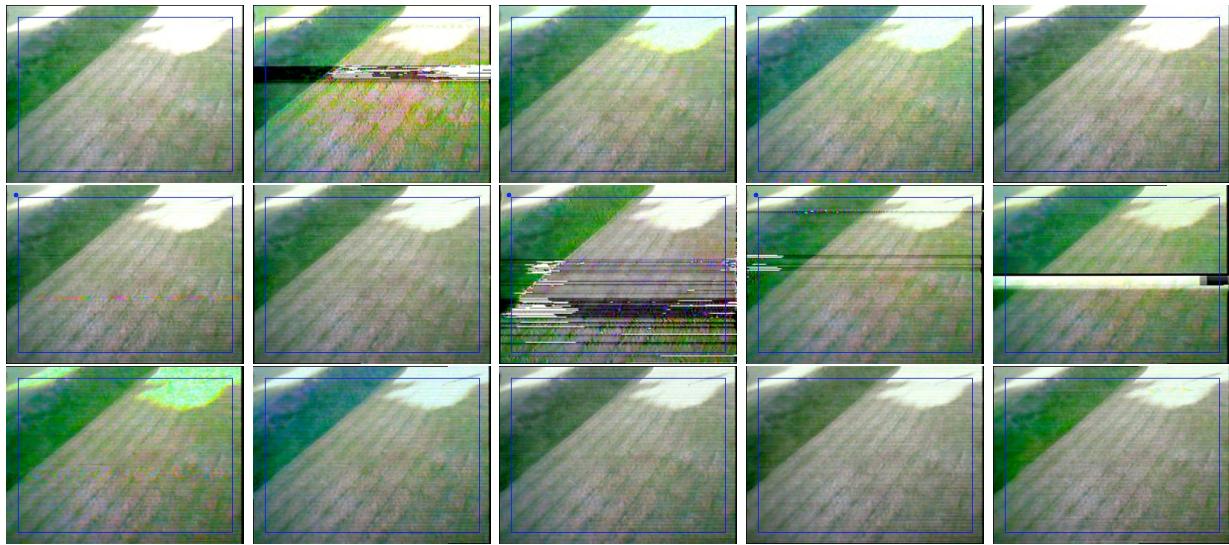


Figura 38: Secuencia de imágenes filtrando las transformaciones en video capturado desde ChocoLate. Se presentan ordenadas de izquierda a derecha y de arriba a abajo. Las imágenes que poseen un punto azul en la esquina superior izquierda son aquellas para las cuales se considera la transformación anterior al estimar la componente intencional del movimiento percibido entre pares. Se utiliza un modelo similar bajo mínimos cuadrados para estimar las transformaciones que modelan el movimiento global.

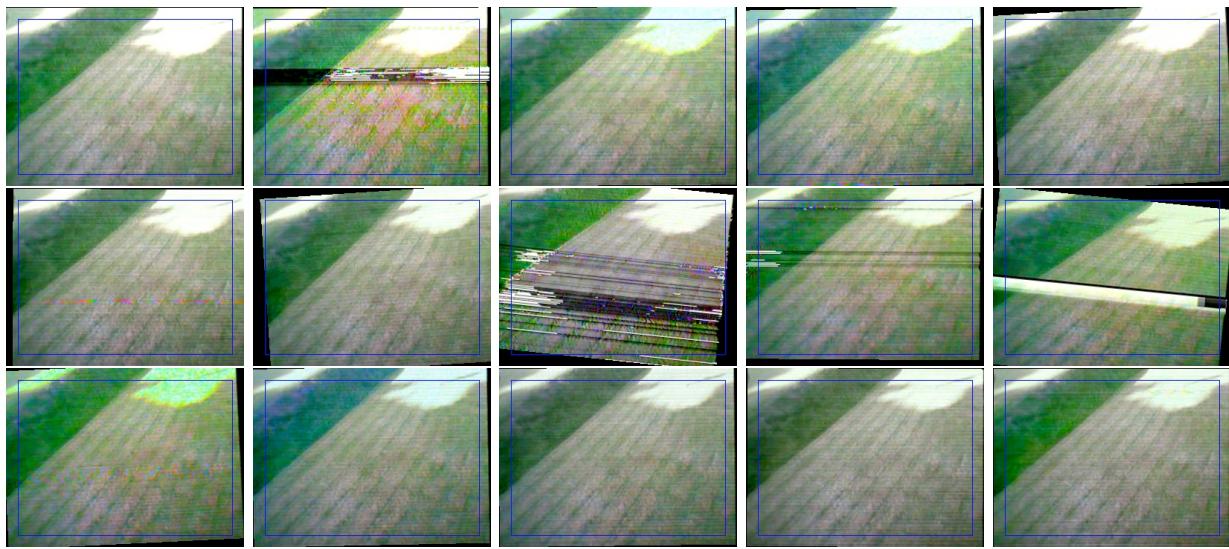


Figura 39: Secuencia de imágenes sin filtrar las transformaciones en video capturado desde ChocoLate. Se presentan ordenadas de izquierda a derecha y de arriba a abajo. Se utiliza un modelo bilineal bajo mínimos cuadrados para estimar las transformaciones que modelan el movimiento global.

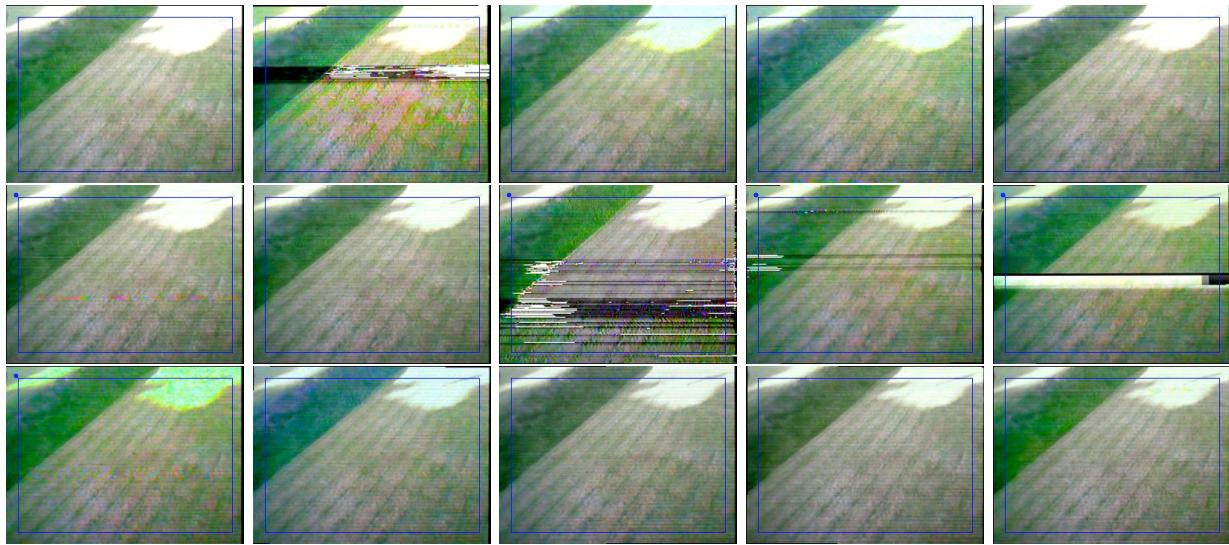


Figura 40: Secuencia de imágenes filtrando las transformaciones en video capturado desde ChocoLate. Se presentan ordenadas de izquierda a derecha y de arriba a abajo. Las imágenes que poseen un punto azul en la esquina superior izquierda son aquellas para las cuales se considera la transformación anterior al estimar la componente intencional del movimiento percibido entre pares. Se utiliza un modelo bilineal bajo mínimos cuadrados para estimar las transformaciones que modelan el movimiento global.

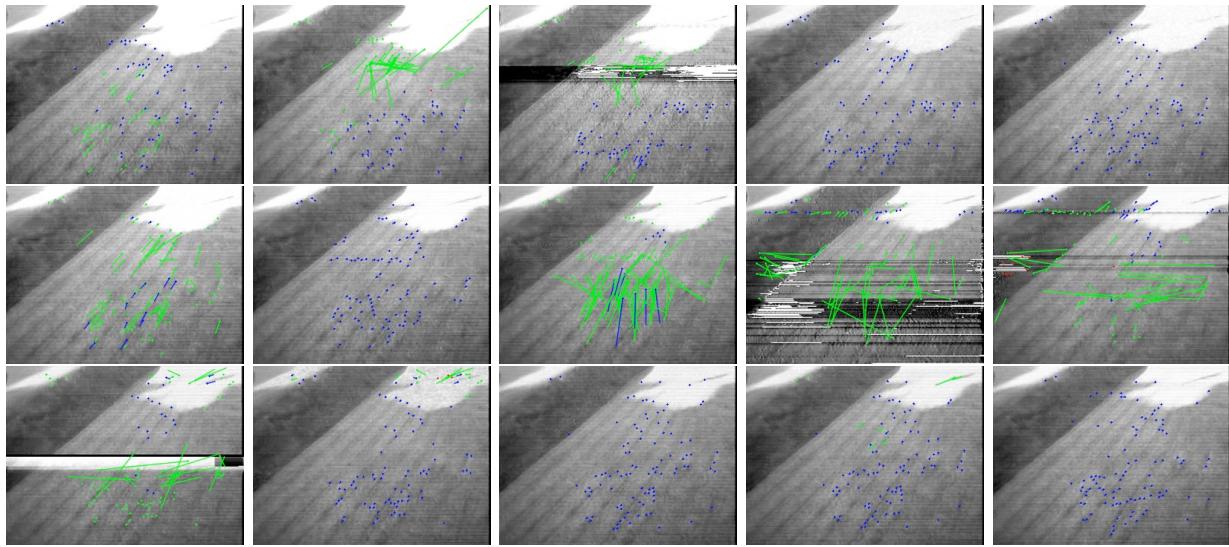


Figura 41: Estimación del campo de movimiento en secuencia de imágenes de video capturado desde ChocoLate. Las imágenes se presentan ordenadas de izquierda a derecha y de arriba a abajo. Se utiliza un modelo bilineal bajo mínimos cuadrados para estimar la transformación que modela el movimiento global. Las líneas verdes representan vectores de movimiento identificados como atípicos.

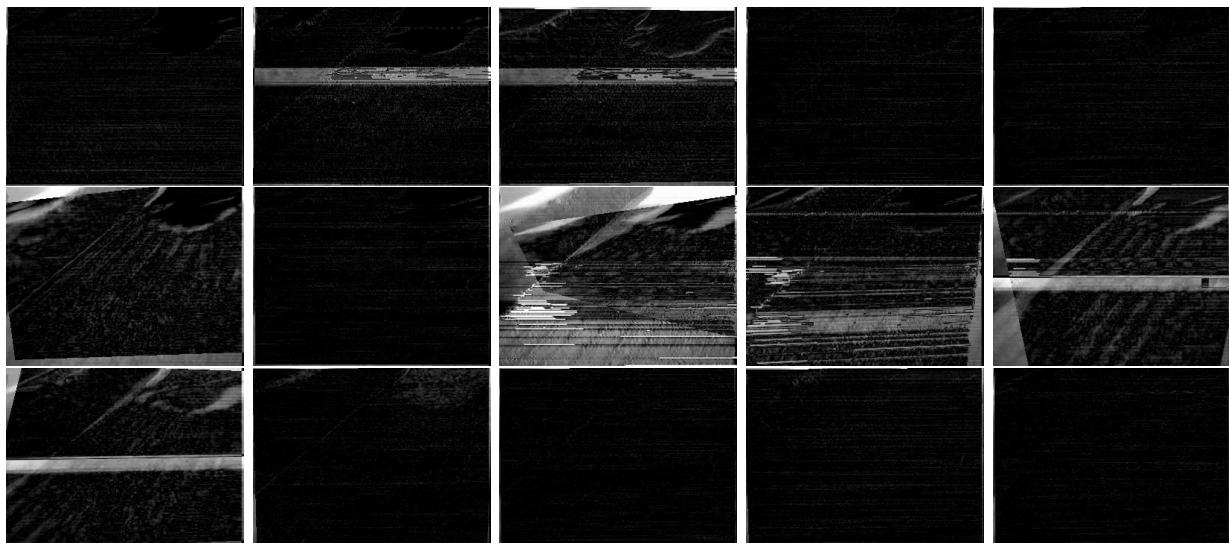


Figura 42: Diferencia entre imágenes consecutivas luego de compensar totalmente el movimiento estimado en secuencia de video capturado desde ChocoLate. Las imágenes se presentan ordenadas de izquierda a derecha y de arriba a abajo. Se utiliza un modelo bilineal bajo mínimos cuadrados para estimar la transformación que modela el movimiento global.

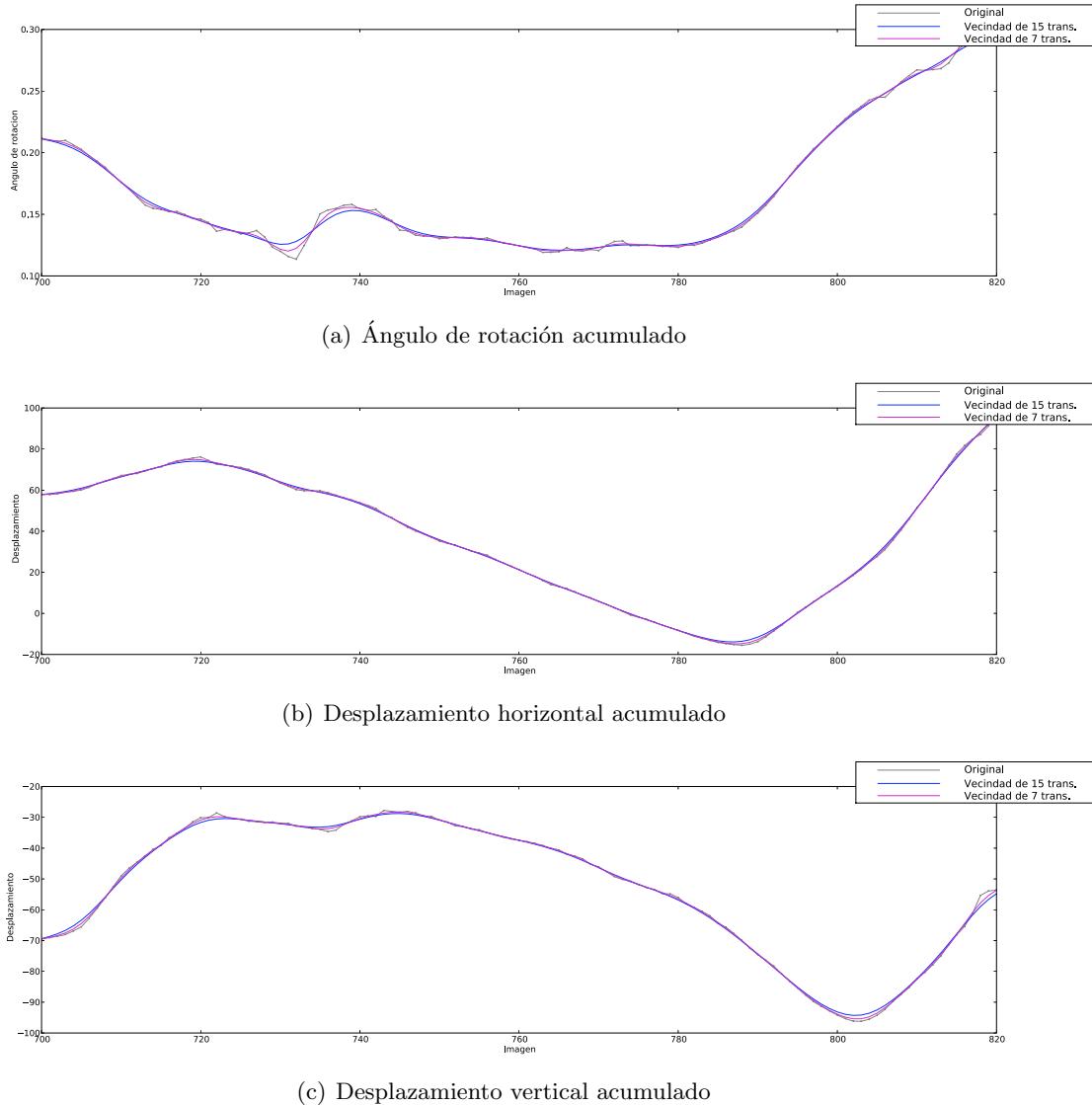


Figura 43: Estimación del movimiento acumulado y su componente intencional bajo un modelo afín por mínimos cuadrados, considerando las imágenes que van desde la 700 hasta la 820 de la secuencia capturada desde ChocoLatte que fue procesada en el Experimento VI. La línea morada representa la estimación de la componente intencional con una vecindad de 7 transformaciones, mientras que la azul muestra el resultado utilizando 15. la línea gris describe el movimiento global acumulado.

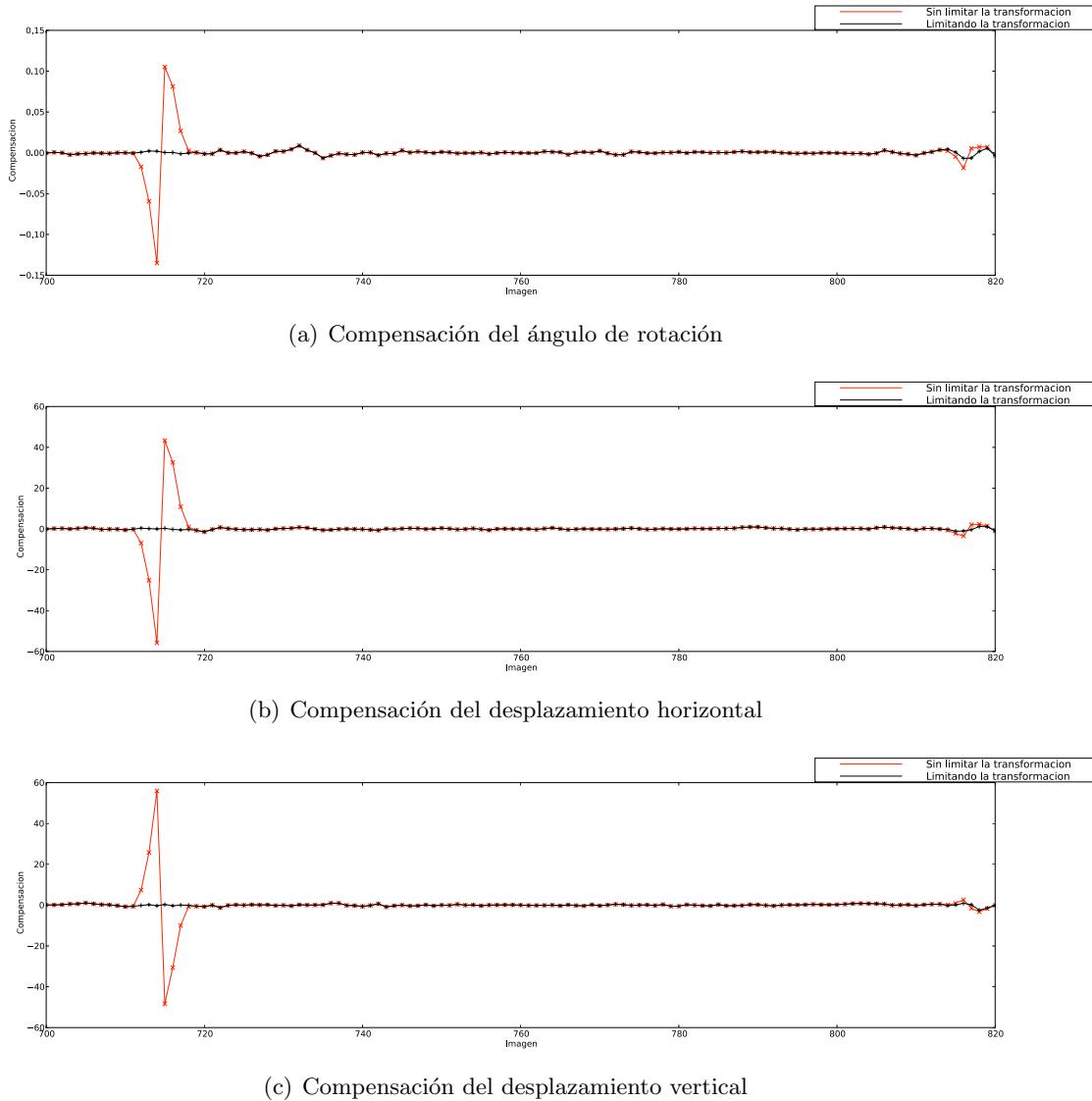


Figura 44: Magnitud de la compensación según el estimado de movimiento global bajo un modelo afín por mínimos cuadrados, considerando las imágenes que van desde la 700 hasta la 820 de la secuencia capturada desde ChocoLate que fue procesada en el Experimento VI. Se considera tanto el caso en que se permite libremente cualquier tipo de transformación para ser compensada (línea roja), así como en el que se descartan aquellas que posiblemente no representan el movimiento global (línea negra). Los resultados son obtenidos para una vecindad de 7 transformaciones, con un máximo de 100 rasgos “buenos” que pueden seleccionarse y W de 7×7 píxeles.

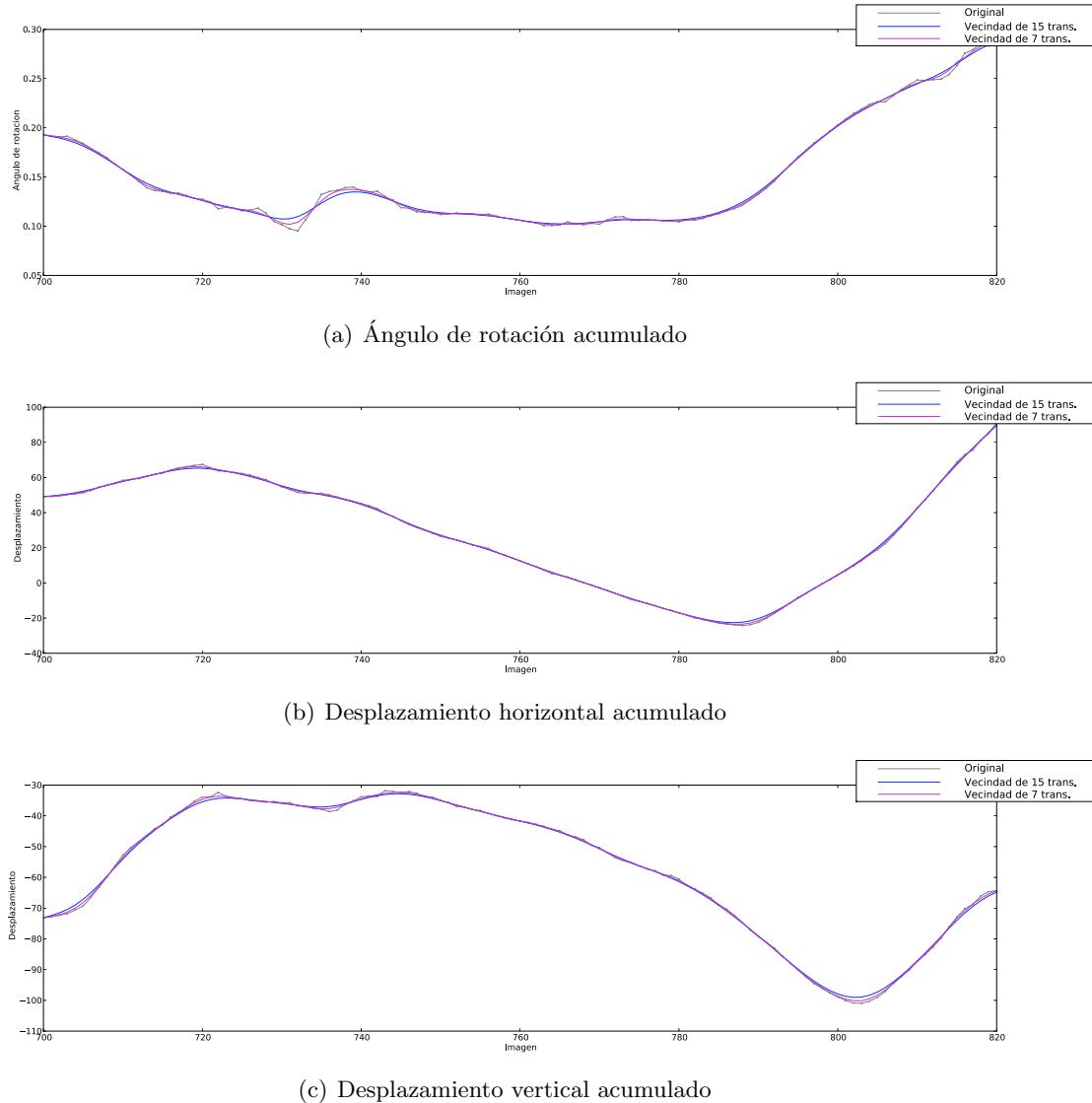


Figura 45: Estimación del movimiento acumulado y su componente intencional bajo un modelo afín por mínimos cuadrados totales, considerando las imágenes que van desde la 700 hasta la 820 de la secuencia capturada desde ChocoLate que fue procesada en el Experimento VI. La línea morada representa la estimación de la componente intencional con una vecindad de 7 transformaciones, mientras que la azul muestra el resultado utilizando 15. la línea gris describe el movimiento global acumulado.

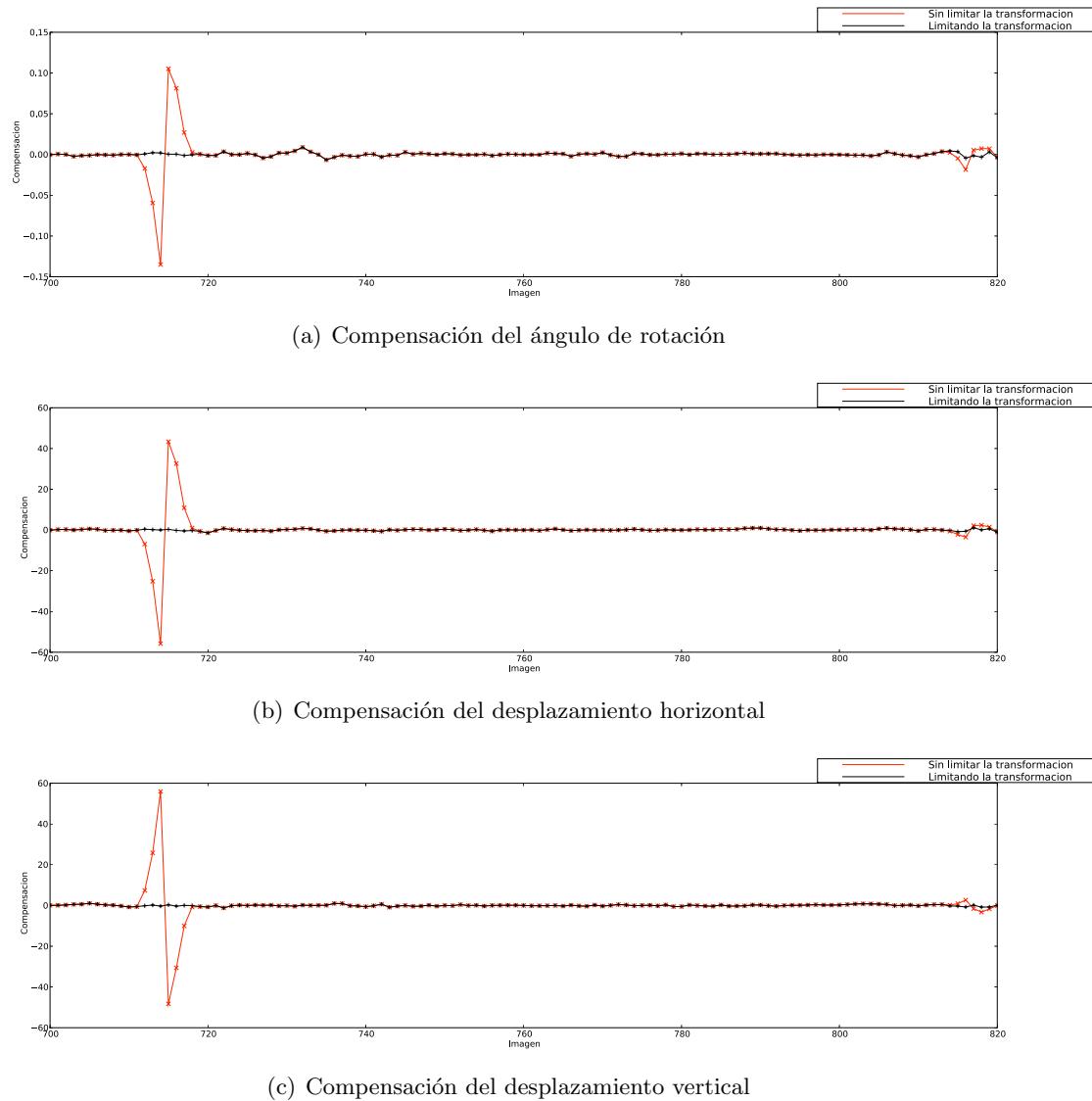


Figura 46: Magnitud de la compensación según el estimado de movimiento global bajo un modelo afín por mínimos cuadrados totales, considerando las imágenes que van desde la 700 hasta la 820 de la secuencia capturada desde ChocoLate que fue procesada en el Experimento VI. Se considera tanto el caso en que se permite libremente cualquier tipo de transformación para ser compensada (línea roja), así como en el que se descartan aquellas que posiblemente no representan el movimiento global (línea negra). Los resultados son obtenidos para una vecindad de 7 transformaciones, con un máximo de 100 rasgos “buenos” que pueden seleccionarse y W de 7×7 píxeles.

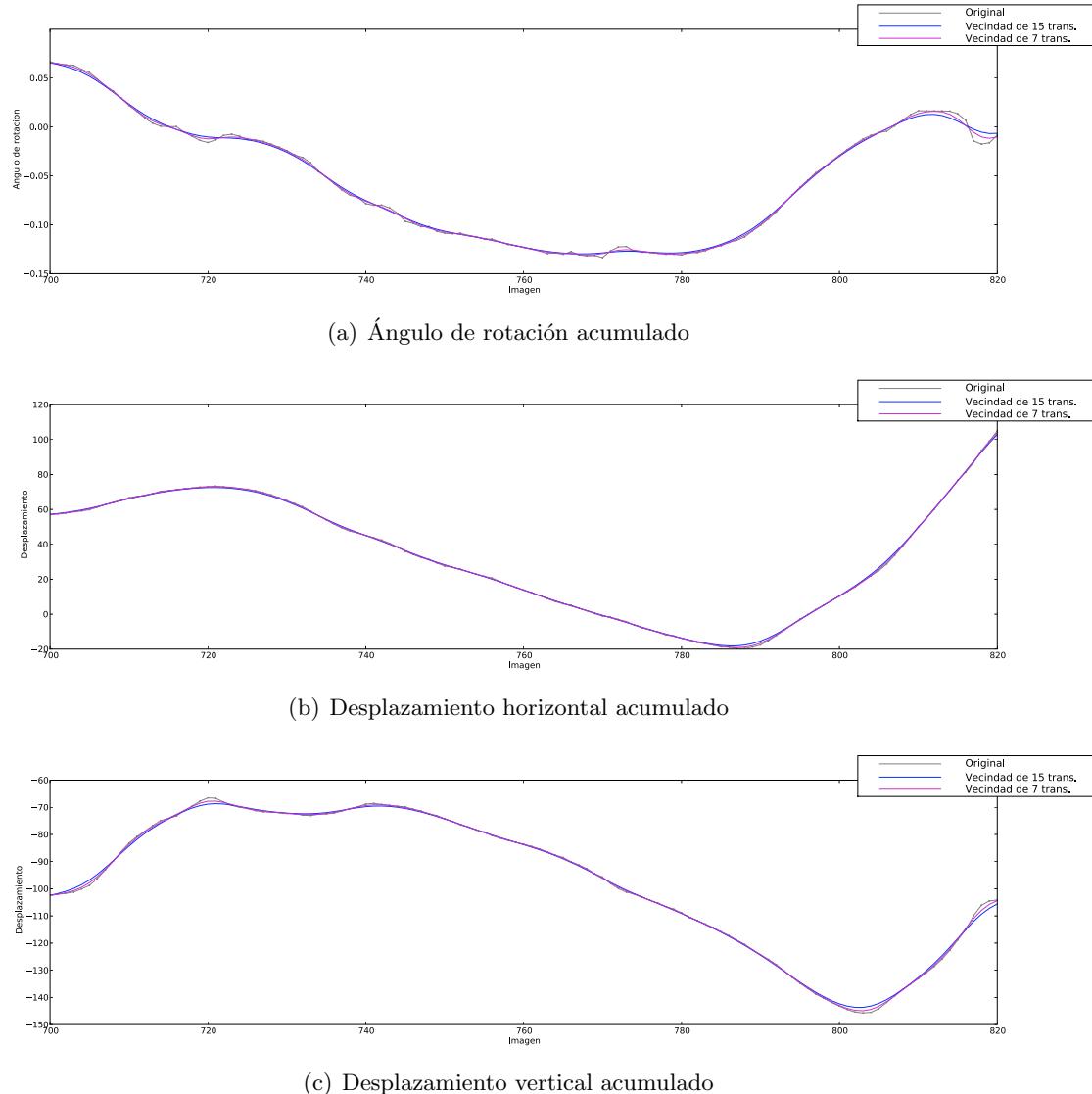


Figura 47: Estimación del movimiento acumulado y su componente intencional bajo un modelo similar por mínimos cuadrados, considerando las imágenes que van desde la 700 hasta la 820 de la secuencia capturada desde ChocoLate que fue procesada en el Experimento VI. La línea morada representa la estimación de la componente intencional con una vecindad de 7 transformaciones, mientras que la azul muestra el resultado utilizando 15. la línea gris describe el movimiento global acumulado.

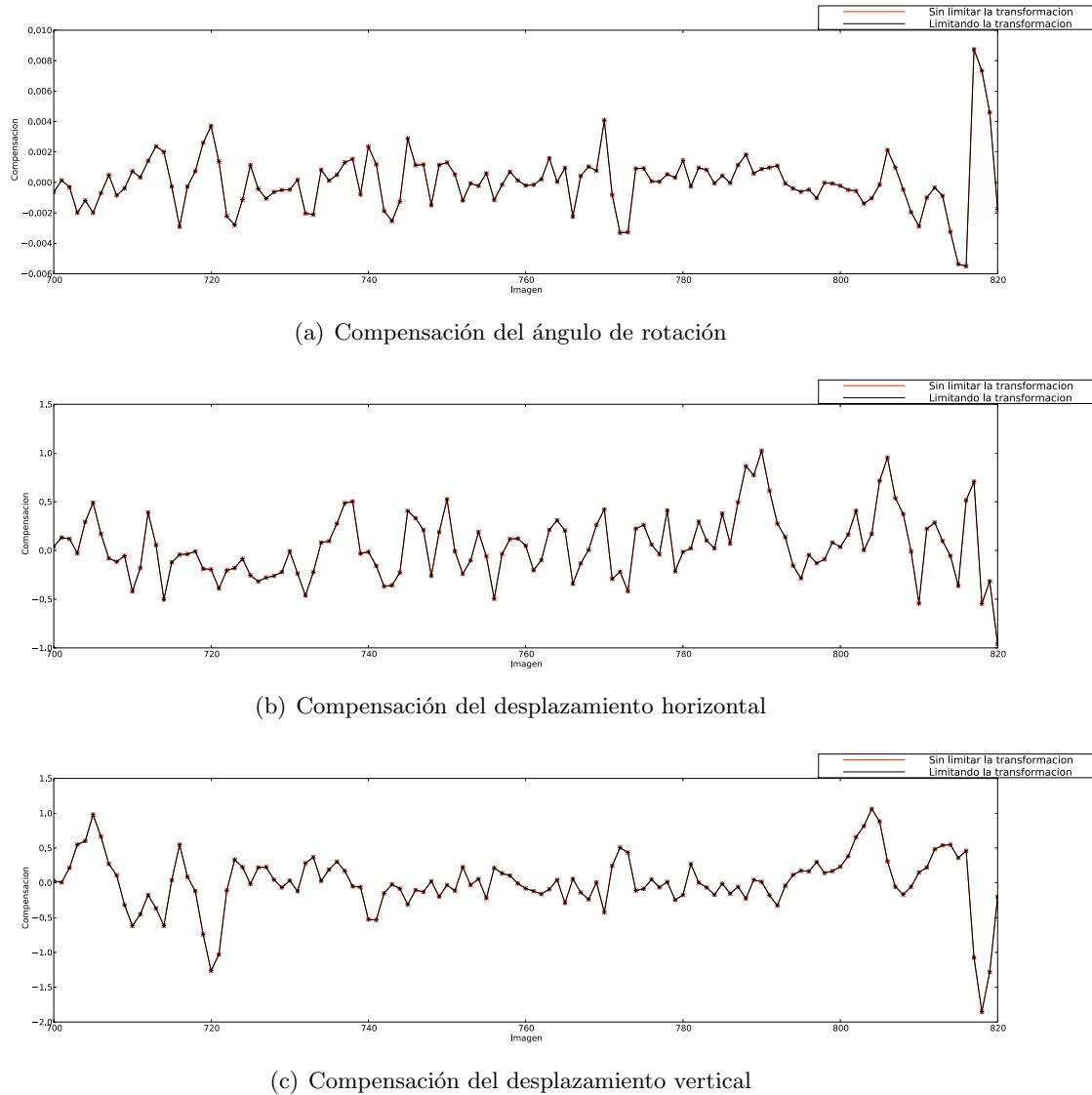


Figura 48: Magnitud de la compensación según el estimado de movimiento global bajo un modelo similar por mínimos cuadrados, considerando las imágenes que van desde la 700 hasta la 820 de la secuencia capturada desde ChocoLake que fue procesada en el Experimento VI. Se considera tanto el caso en que se permite libremente cualquier tipo de transformación para ser compensada (línea roja), así como en el que se descartan aquellas que posiblemente no representan el movimiento global (línea negra). Los resultados son obtenidos para una vecindad de 7 transformaciones, con un máximo de 100 rasgos “buenos” que pueden seleccionarse y W de 7×7 píxeles.

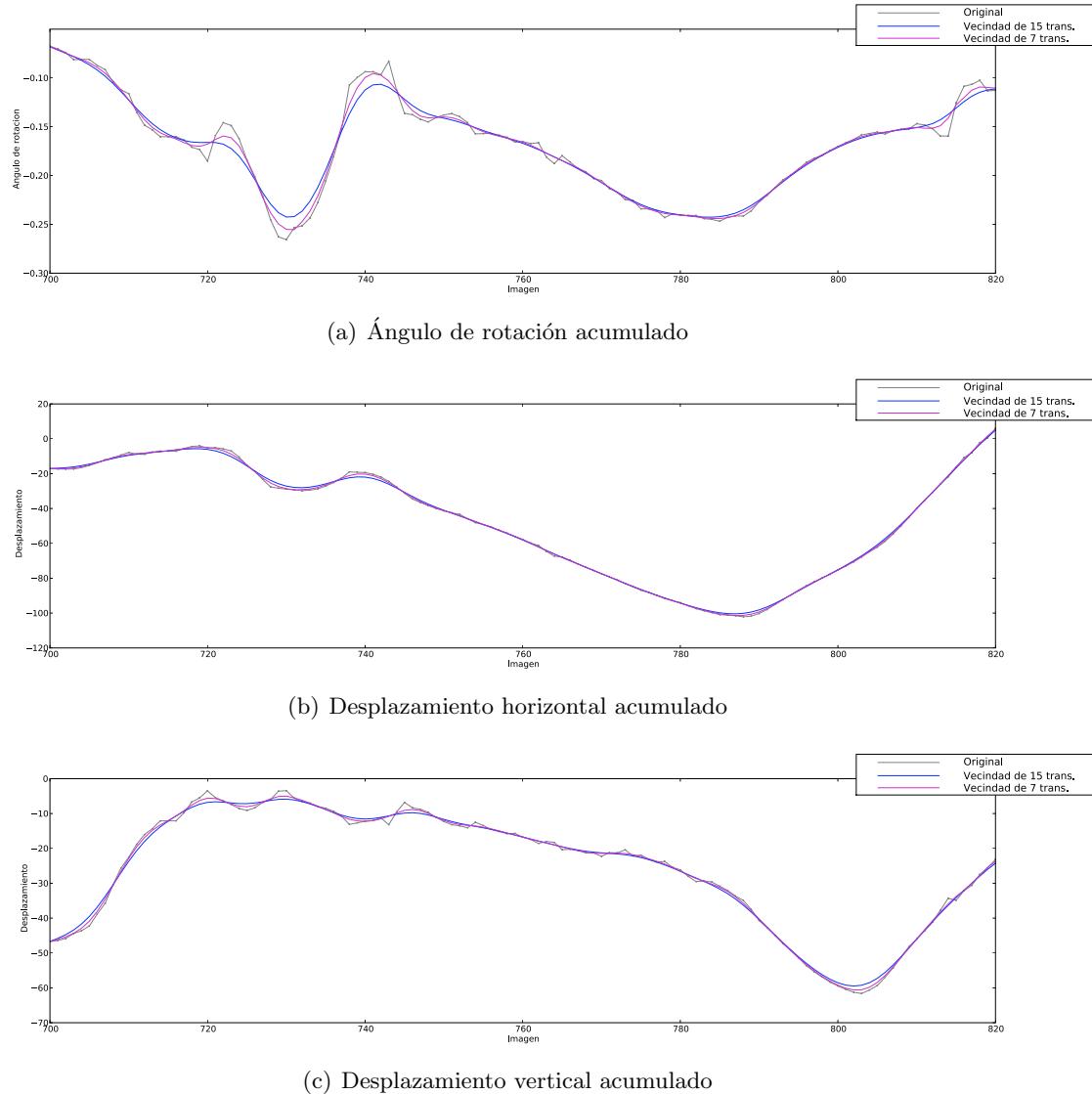


Figura 49: Estimación del movimiento acumulado y su componente intencional bajo un modelo bilineal por mínimos cuadrados, considerando las imágenes que van desde la 700 hasta la 820 de la secuencia capturada desde ChocoLate que fue procesada en el Experimento VI. La línea morada representa la estimación de la componente intencional con una vecindad de 7 transformaciones, mientras que la azul muestra el resultado utilizando 15. la línea gris describe el movimiento global acumulado.

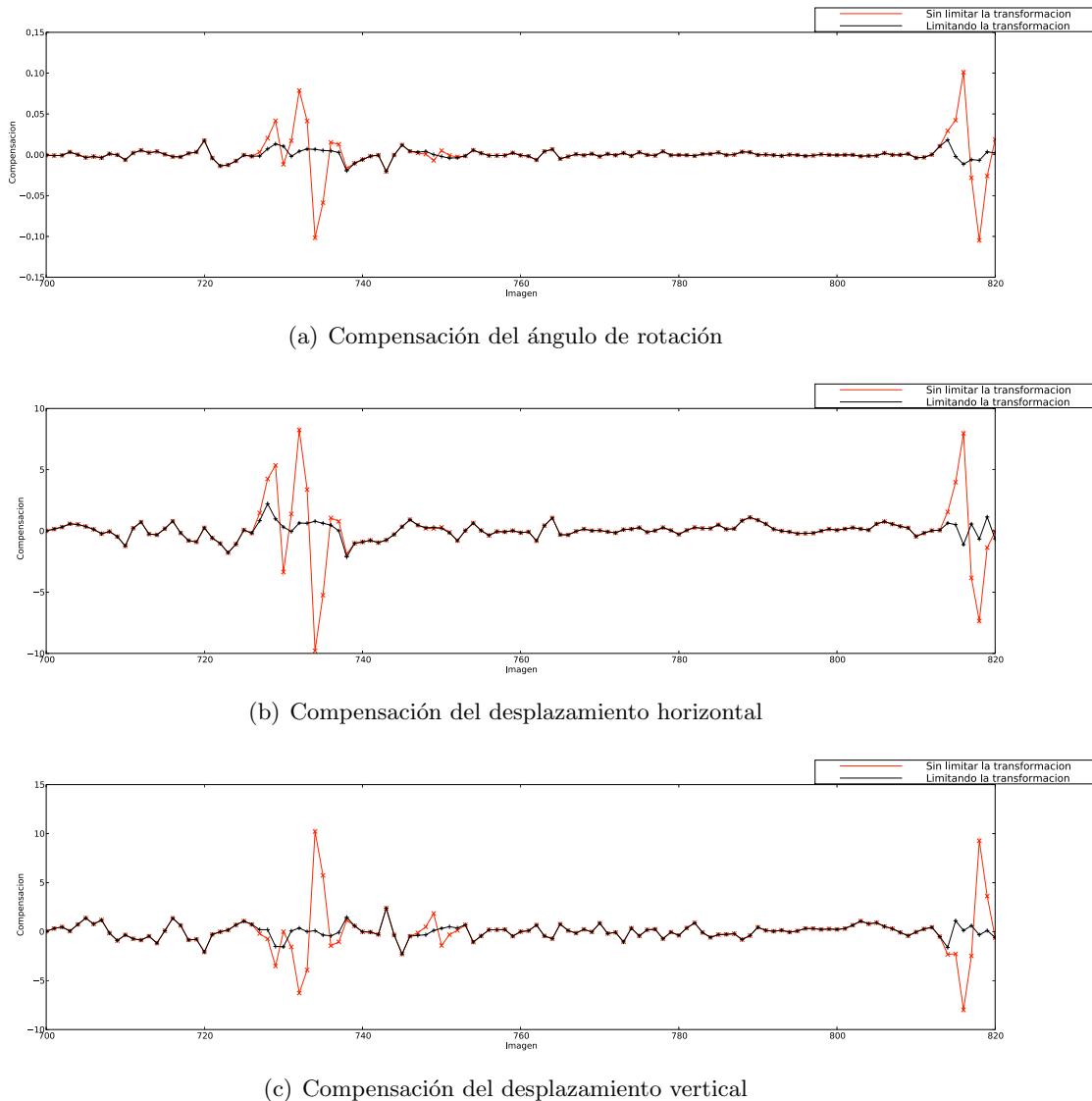


Figura 50: Magnitud de la compensación según el estimado de movimiento global bajo un modelo bilineal por mínimos cuadrados, considerando las imágenes que van desde la 700 hasta la 820 de la secuencia capturada desde ChocoChocolate que fue procesada en el Experimento VI. Se considera tanto el caso en que se permite libremente cualquier tipo de transformación para ser compensada (línea roja), así como en el que se descartan aquellas que posiblemente no representan el movimiento global (línea negra). Los resultados son obtenidos para una vecindad de 7 transformaciones, con un máximo de 100 rasgos “buenos” que pueden seleccionarse y W de 7×7 píxeles.