



# Trabajo Fin de Grado

DOBLE TITULACIÓN: GRADO EN MATEMÁTICAS  
Y GRADO EN INGENIERÍA TELEMÁTICA

Segmentación de fondo y objetos en vídeo utilizando  
movimiento entre imágenes

MARCO SÁNCHEZ BEECKMAN

**Tutores**

Antoni Buades Capó      Bartomeu Coll Vicens

Escola Politécnica Superior  
Universitat de les Illes Balears  
Palma, 22 de septiembre de 2020



# ÍNDICE GENERAL

<b>Índice general</b>	<b>i</b>
<b>Acrónimos</b>	<b>iii</b>
<b>Resumen</b>	<b>v</b>
<b>1 Introducción</b>	<b>1</b>
<b>2 Movimiento en secuencias de imágenes</b>	<b>5</b>
2.1 Imágenes, vídeos, y cámaras . . . . .	5
2.1.1 Modelo de cámara perspectiva . . . . .	6
2.1.2 Espacios de color . . . . .	8
2.2 El flujo óptico . . . . .	10
2.3 Trayectorias de puntos . . . . .	16
2.3.1 Puntos de interés . . . . .	16
2.3.2 Trayectorias semidensas . . . . .	17
<b>3 Segmentación de trayectorias</b>	<b>23</b>
3.1 Modelado geométrico de fondo . . . . .	24
3.1.1 Homografías entre fotogramas . . . . .	25
3.1.2 Extracción de trayectorias del fondo . . . . .	31
3.2 Grafos de trayectorias . . . . .	36
3.2.1 Afinidad de trayectorias . . . . .	37
3.2.2 Cortes mínimos normalizados . . . . .	39
3.2.3 Paseos aleatorios . . . . .	43
3.2.4 Segmentación interactiva . . . . .	46
<b>4 Segmentación de regiones de píxeles</b>	<b>53</b>
4.1 Mapas de prominencia . . . . .	54
4.2 Extracción de regiones . . . . .	56
4.3 Descripción de regiones . . . . .	59
4.4 Propagación de votos . . . . .	60
4.5 Clasificación de regiones . . . . .	62
<b>5 Densificación de trayectorias</b>	<b>65</b>
5.1 Segmentación en el espacio bilateral . . . . .	66
5.1.1 Construcción de la rejilla bilateral . . . . .	67
5.1.2 Partición de la rejilla . . . . .	68

5.1.3	Recuperación de las imágenes segmentadas . . . . .	70
5.2	Sesgo por trayectorias en el espacio bilateral . . . . .	71
<b>6</b>	<b>Evaluación de los algoritmos</b>	<b>77</b>
6.1	Medidas . . . . .	78
6.1.1	Similitud de objetos . . . . .	78
6.1.2	Exactitud de contornos . . . . .	79
6.1.3	Estabilidad temporal . . . . .	79
6.2	Resultados . . . . .	80
<b>7</b>	<b>Conclusiones y trabajo futuro</b>	<b>89</b>
<b>Bibliografía</b>		<b>91</b>

## ACRÓNIMOS

**CIE** Commission Internationale de l'Éclairage

**DAVIS** Densely Annotated Vldeo Segmentation

**DLT** Direct Linear Transform

**DTW** Dynamic Time Warping

**FPS** Fotogramas por Segundo

**HOG** Histogram of Oriented Gradients

**IRLS** Iteratively Reweighted Least Squares

**KLT** Kanade–Lucas–Tomasi

**LDOF** Large Displacement Optical Flow

**MRF** Markov Random Field

**NLCV** Non-Local Consensus Voting

**RANSAC** RANdom SAmple Consensus

**RGB** Red–Green–Blue

**SCD** Shape Context Descriptor

**SLIC** Simple Linear Iterative Clustering

**TV-L<sup>1</sup>** Total Variation- $L^1$

**VOL** Video Object Layer

## RESUMEN

La información de carácter visual constituye una gran parte de los datos que se intercambian en la actualidad. Debido a ello, surge la necesidad de entender a alto nivel el contenido de imágenes y vídeos. Preguntas como «¿Qué hay?», «¿Dónde se encuentra?», y «¿Cuándo aparece?» son de especial interés, siendo estudiadas por el campo de la visión artificial. Dentro de esta, la segmentación de objetos pretende contestar a la segunda pregunta dividiendo las imágenes en las entidades individuales perceptiblemente significativas que las conforman; asimismo, la respuesta a la tercera se obtiene extendiendo naturalmente el proceso sobre vídeos. En este trabajo se presentan y comparan estrategias de segmentación en vídeos con la finalidad de identificar, delimitar, y seguir con precisión objetos característicos a lo largo del tiempo, enfatizando su separación de las regiones menos importantes que componen su fondo.

Apoyado por el color y la textura, el movimiento entre imágenes es el pilar fundamental en el que se respaldan los métodos de segmentación exhibidos. Tanto la información a corto plazo dada en campos de flujo óptico como las trayectorias de larga duración de puntos clave se utilizan para diferenciar objetos que se desplazan distintivamente. Específicamente para describir estas trayectorias se desarrollan algoritmos de rastreo de muestras de puntos, cuyos recorridos asisten en su clasificación: por un lado, empleando modelos geométricos se logra aislar los puntos de fondos planares del resto de contenido; por otro, se construyen grafos que conectan trayectorias similares, sobre los que se realizan cortes y paseos aleatorios para agrupar puntos que se comportan igual. La clasificación de las muestras luego se extiende a toda la superficie de las imágenes. Segmentaciones densas se obtienen también estableciendo conexiones probabilísticas entre regiones parecidas prominentes en los vídeos, que se separan por un sistema de decisión por consenso.

Para cada uno de los métodos planteados se ofrecen resultados cualitativos y cuantitativos sobre una serie de vídeos, que permiten valorar visual y numéricamente su fidelidad a la realidad. Dados estos últimos, se concluye que todos ellos tienen cabida en aplicaciones con diferentes propósitos, desde la segmentación automática de objetos hasta la edición interactiva de vídeos.





## INTRODUCCIÓN

El rápido desarrollo de terminales móviles inteligentes y la creciente accesibilidad a Internet en grandes partes del mundo ha conducido a un incremento exponencial en el intercambio de datos en forma de vídeos. Tecnologías emergentes son cada vez más dependientes de información que procede de cámaras, el tratamiento manual de la cual es inviable a gran escala. De este modo, el progreso en el campo de la visión artificial se ha vuelto necesario para que máquinas sean capaces de analizar y manipular eficientemente estos datos. En específico, el problema de localizar, seguir, y aislar objetos de interés en un contexto determinado es especialmente importante para llegar a un nivel de comprensión de su contenido equiparable al de la percepción humana. La división de la superficie de las imágenes que constituyen un vídeo para crear agrupaciones correspondientes visualmente a las entidades u objetos presentes en él es el proceso por el que se busca una solución a este último problema. Bajo el nombre de *segmentación de objetos en vídeo*, este es precisamente el foco de estudio de este trabajo.

El objetivo de la segmentación de objetos es extraer instancias de elementos que destacan visualmente dentro de un vídeo para separarlas entre sí, sin atribuirles necesariamente una categoría semántica. Un caso particular de esta es la segmentación binaria o de fondo, en la que los puntos del vídeo únicamente se agrupan en dos conjuntos disjuntos  $\mathcal{F}$  y  $\mathcal{O}$ , donde uno de ellos engloba todas las diferentes instancias de objetos que aparecen. Estrechamente relacionada a ella se encuentra la tarea del seguimiento de objetos, que pretende determinar el movimiento de estos a lo largo del tiempo. Ciertamente, desde la solución del problema de segmentación es inmediato saber cómo y a dónde se desplazan los distintos objetos sobre el plano de imagen. De manera inversa, los indicios que proporciona un rastreo preciso de estos pueden servir de guía para identificar la posición de los puntos que los conforman. A pesar de que la estimación del movimiento no es indispensable para llegar a segmentaciones fieles a la realidad, como se ha evidenciado en trabajos recientes como [34], sí es una herramienta valiosa que ha fundado la base de la gran mayoría de estrategias concebidas hasta la fecha [12, 24, 36, 37, 38, 48]. Justificado en este hecho, el uso del movimiento tanto a

## 1. INTRODUCCIÓN

---

corto como a largo plazo es prevalente en todos los algoritmos de segmentación que se exhiben en esta memoria.

Es habitual en la literatura organizar las estrategias de segmentación según su grado de automaticidad [49]. Por un lado, métodos *no supervisados* [12, 24, 38, 48] se caracterizan por producir segmentaciones basándose exclusivamente en la información contenida en los vídeos a partir de hipótesis pre establecidas sobre el comportamiento de los objetos, como su movimiento relativo y estructura. En el otro extremo, métodos *interactivos* [36, 37] emplean datos adicionales suministrados por un usuario para que este tenga un mayor control sobre los resultados. La categoría de métodos *semisupervisados* también es utilizada para denotar concretamente aquellos que propagan en el tiempo una segmentación manual inicial [39]. Tanto el grado de interacción como la forma en la que esta se acepta depende del diseño de cada algoritmo; mientras que la opción más habitual para involucrar al usuario es a través de anotaciones, en los últimos años se han estudiado alternativas como el uso de expresiones del lenguaje natural [25] para cumplir su misma función. Así, la necesidad de un tipo de estrategia u otro para un propósito dado depende del entorno en el que ha de aplicarse.

La utilidad de la segmentación y seguimiento de objetos ha ido aumentando desde principios de siglo. La compresión de vídeos y la selección automática de fotogramas que los resumen son ejemplos de procesos que se sirven de sus resultados. Específicamente en el primero, el estándar MPEG-4 de compresión de datos audiovisuales define el concepto de Video Object Layer (VOL) para codificar secuencias en función de su contenido segmentado, y flexibilizar su reconstrucción y manipulación en el decodificador [22, 26]. Métodos no supervisados dan también soporte a tareas de videovigilancia [9], de reconocimiento con drones [32], se usan en *software* de videoconferencias para extraer el fondo sin pantalla verde [22], e incluso pueden ayudar en la toma de decisiones en vehículos autónomos [27]. Asimismo, la edición de vídeo se beneficia de segmentaciones interactivas, que agilizan el trabajo de delimitar manualmente objetos de interés [37]. Todas estas aplicaciones, y las que pueden existir en un futuro, son un buen motivo para desarrollar variedad de algoritmos de segmentación adecuados a distintas finalidades.

La intención de este trabajo es presentar e implementar diferentes algoritmos de segmentación de objetos, aplicables a vídeos con principio y final conocidos, siguiendo estrategias variadas que han sido propuestas en la bibliografía de la última década. Desde métodos no supervisados hasta interactivos, el estudio se centra en aquellos que usan el movimiento entre imágenes como aspecto principal para discernir los objetos entre sí y separarlos del fondo, reflejando los procedimientos predominantes en el ámbito. Concretamente, se hace énfasis en el seguimiento de trayectorias de muestras de puntos clave como herramienta auxiliar para llegar a segmentaciones coherentes en el tiempo, que ha probado ser eficaz para distinguir objetos por su comportamiento a largo plazo [24, 37, 38, 48]. De este modo, en la memoria se ofrecen múltiples algoritmos de rastreo de puntos y de agrupación de sus recorridos, junto con una técnica de *densificación* para extender sus resultados parciales a segmentaciones exhaustivas y detalladas. Además, otro algoritmo basado en la localización y vinculación probabilística de regiones con desplazamientos prominentes [12] se expone en contrapartida a los anteriores para comparar sus características. Implementados todos estos algoritmos, se pretende identificar sus virtudes y sus flaquezas cualitativa y cuantitativamente, y con ello finalmente inferir los tipos de vídeos y entornos más aptos para su aplicación.

---

La estructura del documento es la que sigue a continuación. En el Capítulo 2 se introducen formalmente los conceptos básicos de imagen y vídeo que fundamentan los algoritmos de los capítulos siguientes; se explica el modelo de proyección perspectiva usado en cámaras, y se describen maneras de estimar el flujo óptico entre imágenes y de rastrear trayectorias de puntos. En el Capítulo 3 se exponen algoritmos para clasificar las trayectorias obtenidas en el Capítulo 2 según el objeto (o fondo) al que siguen. Entre ellos se incluyen un método de modelado geométrico del fondo para su extracción automática, y una estrategia interactiva basada en la propagación de anotaciones en el espacio y el tiempo. En el Capítulo 4 se presenta una estrategia no supervisada que prescinde de las trayectorias y logra segmentaciones densas formulando un sistema de decisión por consenso entre regiones de apariencia similar, a las que se atribuyen votos según la prominencia de su movimiento. En el Capítulo 5 se recuperan los resultados del Capítulo 3 y se les aplica un proceso de densificación para que se acomoden a la superficie completa de las imágenes. Exhibidos ya todos los algoritmos, en el Capítulo 6 se dan métricas para su evaluación, y se discuten los resultados visuales y cuantitativos de su aplicación sobre vídeos de los que se dispone de una segmentación manual exacta. Para acabar, en el Capítulo 7 se sintetizan los puntos más importantes presentados en la memoria y se exploran diferentes direcciones en las que se puede continuar con la línea de investigación.



## MOVIMIENTO EN SECUENCIAS DE IMÁGENES

Este capítulo tiene por objetivos formalizar los conceptos de imagen y vídeo, y exponer la manera en la que estos reproducen la información de una escena del mundo real. En concreto, su foco se encuentra en el estudio del movimiento entre imágenes como herramienta para la segmentación de objetos en vídeo que se lleva a cabo en los capítulos que le siguen.

En la Sección 2.1 se explica la noción de imagen como la representación bidimensional de los puntos de una región del espacio capturados por una cámara, extendiéndola en el tiempo para presentar la idea de vídeo. Se describen también el modelo general de proyección que utiliza una cámara, y distintos espacios usados para interpretar el color. A continuación, en la Sección 2.2, se define el flujo óptico como el movimiento aparente entre dos imágenes, y se formula su obtención como un problema de optimización. Se proporcionan distintos métodos para su cálculo, sus limitaciones, y se comparan entre ellos. Finalmente, en la Sección 2.3, se introducen las trayectorias de puntos como forma de conocer el movimiento a largo plazo dentro de un vídeo. Se define lo que es un punto de interés para rastrear, y se presentan dos algoritmos de estimación de trayectorias, que se diferencian en la cantidad de ellas que son capaces de seguir.

### 2.1 Imágenes, vídeos, y cámaras

Matemáticamente, una *imagen* es una aplicación  $I: \Omega \rightarrow \mathbb{R}^d$  que asocia a  $\mathbf{x} \in \Omega \subseteq \mathbb{R}^2$  un valor  $I(\mathbf{x}) \in \mathbb{R}^d$ , presente en un *espacio de color*. El conjunto  $\Omega$  es el dominio espacial de la imagen, que puede ser cualquier subconjunto cerrado de  $\mathbb{R}^2$  de área finita, normalmente un rectángulo. A la dimensionalidad  $d$  del espacio de llegada se le llama el número de *canales* de la imagen: en el caso específico  $d = 1$ , se dice que la imagen está en escala de grises y a  $I(\mathbf{x})$  se le nombra *luminosidad* o *nivel de gris*; con  $d = 3$ ,  $I(\mathbf{x})$  es el *color* de  $\mathbf{x}$ ; y cuando  $d > 3$ , a  $I$  se le denomina una *imagen hiperespectral*.

Una imagen digital es aquella cuyo dominio espacial ha sido muestreado, y sus valores cuantizados para cada uno de sus canales. El muestreo espacial da lugar a

## 2. MOVIMIENTO EN SECUENCIAS DE IMÁGENES

---

que  $\Omega = \{1, \dots, M\} \times \{1, \dots, N\}$ , posibilitando la representación de la imagen como una matriz multidimensional de tamaño  $M \times N \times d$ , mientras que la cuantización establece la existencia de un número finito de posibles valores de  $\text{Im } I$ , que dependen únicamente del espacio de color. Los puntos de  $\Omega$  en una imagen digital, correspondientes a cada muestra espacial, son llamados *píxeles*. En la práctica, todas las imágenes sobre las que se aplican los algoritmos descritos en este trabajo son digitales; no obstante, salvo especificarse lo contrario, la notación utilizada en las expresiones matemáticas no hace suposiciones sobre el tipo de imágenes tratadas.

La definición de imagen puede extenderse en el tiempo, constituyendo el concepto de *vídeo*, una aplicación  $\mathcal{V} : \Omega \times [1, T] \rightarrow \mathbb{R}^d$  cuyo dominio distingue sus puntos según el momento en el que aparecen. El intervalo  $[1, T] \subset \mathbb{R}$  (con  $T \in \mathbb{N}$ ) usualmente se discretiza en  $T$  instantes, lo que convierte un vídeo en una secuencia de  $T$  imágenes coherentes temporalmente (con *movimiento*) entre sí, nombradas *fotogramas* (o *frames*). Cuando no se tiene en cuenta el tiempo como variable, se define la notación  $I_t = \mathcal{V}(\cdot, t)$ , que fijado un fotograma  $t$ , indica la  $t$ -ésima imagen de la secuencia.

### 2.1.1 Modelo de cámara perspectiva

La gran mayoría de imágenes y vídeos son capturas de una escena tridimensional representadas en dos dimensiones (e. g. fotografías), o pueden modelarse como tales (e. g. animaciones). Es por ello que, con la finalidad de interpretar el contexto original a partir de su representación, existe especial interés en entender la forma en la que este proceso se lleva a cabo.

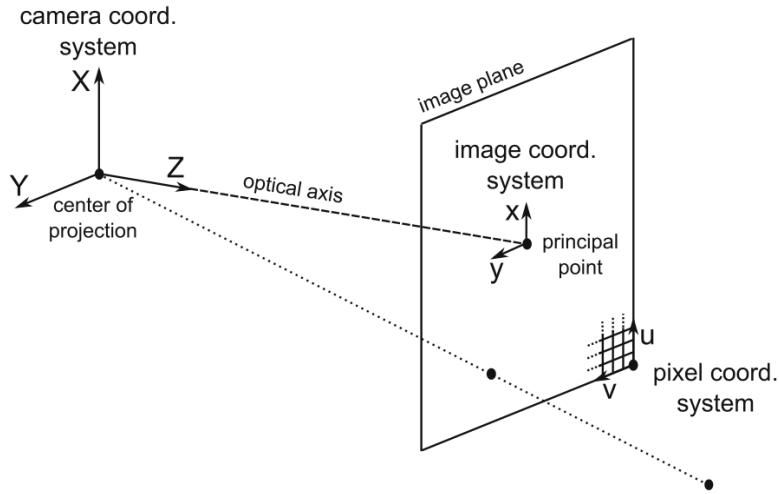
El observador que capta una escena es habitualmente una cámara, que generalmente consiste en una caja oscura cerrada con una abertura en un extremo que deja pasar la luz, un objetivo formado por lentes posicionado delante de la abertura, y una superficie plana en el otro extremo donde se forman las imágenes [44]. La manera más sencilla de modelar una de ellas consiste en suponer que la abertura es un único punto en el espacio, y que no existe ningún tipo de distorsión provocada por las lentes, conociéndose como el modelo de cámara perspectiva.

La Figura 2.1 [44, Fig. 2] muestra la proyección en perspectiva de un punto en el espacio sobre una imagen. Las coordenadas de la cámara se toman sobre un sistema de referencia ortogonal de tal manera que el centro de proyección (o *foco*) se encuentra en su origen, y el plano de imagen (o *plano retinal*) se sitúa perpendicular a uno de sus ejes (el *eje óptico*) a una distancia  $f > 0$ , denominada *distancia focal*. El punto de intersección entre el plano retinal y el eje óptico se llama *punto principal*, y sirve de origen para el sistema de coordenadas de la imagen, cuyos ejes son paralelos a los otros dos de la escena. Definidas estas referencias, un punto  $(X, Y, Z)$  en el espacio y su proyección  $(x, y)$  están relacionados por las igualdades

$$\frac{x}{X} = \frac{y}{Y} = \frac{f}{Z}, \quad (2.1)$$

que utilizando coordenadas homogéneas pueden escribirse linealmente como

$$\begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (2.2)$$



**Figura 2.1:** Diagrama del modelo de cámara perspectiva en el que figuran los sistemas de coordenadas de la cámara, de la imagen, y de los píxeles. Un punto del espacio se proyecta sobre la imagen intersecando el segmento que traza hacia el centro de proyección con el plano perpendicular al eje óptico a una distancia focal  $f$ .

pudiendo recuperar  $x$  e  $y$  si  $s = Z \neq 0$ . En un caso real esto siempre es posible, porque las cámaras captan únicamente la luz que les incide desde delante del plano focal (con  $Z > 0$ ). Es importante destacar que en la proyección (2.2) todos los puntos del espacio de la forma  $(\lambda X, \lambda Y, \lambda Z)$  con  $\lambda > 0$  se corresponden con el mismo punto  $(x, y)$  de la imagen, ya que trazan el mismo rayo hacia el foco. De esta forma, se pierde información de la escena real al reproducirla en una imagen, lo que se hace evidente al no poder capturar puntos situados detrás de un objeto opaco. Asimismo, las rectas paralelas en el espacio cuya dirección tiene una componente no nula en el eje óptico no se preservan al proyectarse en una imagen, sino que se juntan en un punto de fuga.

El modelo de cámara perspectiva puede adaptarse a cámaras digitales definiendo un nuevo sistema de referencia para píxeles. La configuración usual de estos últimos consiste en una rejilla rectangular donde se ordenan en filas y columnas, cuyo origen se sitúa en una de las esquinas de la imagen, con ejes paralelos a los del sistema de coordenadas de esta. Sean  $k_u$  y  $k_v$  las respectivas densidades (o *tasas de muestreo espaciales*) de píxeles por columnas y filas, y  $(x_0, y_0)$  la posición de la esquina referida en las coordenadas de la imagen, un punto  $(x, y)$  representado en estas últimas y su píxel  $(u, v)$  están relacionados por las igualdades

$$\begin{aligned} u &= k_u(x - x_0), \\ v &= k_v(y - y_0), \end{aligned} \tag{2.3}$$

que junto a la ecuación (2.2) conducen a

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} k_u f & 0 & -k_u x_0 & 0 \\ 0 & k_v f & -k_v y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{2.4}$$

## 2. MOVIMIENTO EN SECUENCIAS DE IMÁGENES

---

Las entradas de la matriz del sistema (2.4) se conocen como los *parámetros intrínsecos* del modelo de la cámara; y la submatriz cuadrada  $\mathbf{K}$  de dimensión  $3 \times 3$  que resulta de excluir su última columna, como su *matriz de calibración*. Es posible considerar un quinto parámetro intrínseco en lugar del cero de la primera fila para modelar ejes de muestreo distorsionados, aunque en cámaras modernas acostumbra a ser negligible.

En el escenario de una cámara en movimiento o múltiples cámaras capturando una misma escena desde distintos ángulos, es necesario describir la posición y orientación del observador respecto de una referencia común fija. Esta puede escogerse arbitrariamente, siendo lo más conveniente utilizar el sistema de coordenadas de cualquiera de las cámaras (en el caso de múltiples de ellas), o de la cámara en el primer fotograma (en el caso de un vídeo). Sean  $\mathbf{c}$  las coordenadas del centro de proyección respecto de la anterior referencia, y  $\mathbf{R}$  la matriz de rotación que representa la orientación de la cámara, un punto  $(X_w, Y_w, Z_w)$  en el espacio se escribe en el sistema de referencia específico de esta última como

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & -\mathbf{R}\mathbf{c} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}. \quad (2.5)$$

Combinando (2.4) y (2.5), resulta la expresión completa del modelo de cámara digital perspectiva:

$$\begin{aligned} \begin{bmatrix} su \\ sv \\ s \end{bmatrix} &= [\mathbf{K} \quad \mathbf{0}] \begin{bmatrix} \mathbf{R} & -\mathbf{R}\mathbf{c} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \\ &\iff \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \underbrace{[\mathbf{K}\mathbf{R} \quad -\mathbf{K}\mathbf{R}\mathbf{c}]}_{\mathbf{P}} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}, \end{aligned} \quad (2.6)$$

en la cual  $\mathbf{P}$  es la *matriz de proyección* o *matriz de cámara*.

### 2.1.2 Espacios de color

El color es una sensación creada en respuesta a la excitación del sistema visual humano por la luz, correspondiente a la región visible del espectro electromagnético (con longitudes de onda de entre 400 nm y 700 nm), que incide sobre la retina del ojo humano [40, p. 1]. Esta última tiene tres tipos de células fotoreceptoras, llamadas *conos*, sensibles respectivamente a longitudes de onda cortas (430 nm, luz azul), medianas (530 nm, luz verde), y largas (700 nm, luz roja). Un cuarto tipo de células fotoreceptoras, los *bastones*, se encuentran también en la retina, siendo efectivas solo en bajas condiciones de luminosidad e incapaces de percibir colores.

Como existen exactamente tres tipos de células fotoreceptoras dedicadas a la percepción del color, tres componentes numéricas son necesarias y suficientes para describirlo, siempre y cuando se ponderen adecuadamente en función de las respuestas espetrales de dichas células. Consecuentemente, una imagen requiere de tres canales para especificar el color de sus píxeles.

La forma de definir cómo ha de representarse un color a partir de una terna de valores depende del espacio de color de una imagen. La colorimetría estudia la medida del color, existiendo numerosos métodos para cuantificar cómo el ser humano lo percibe. El modelo más sencillo consiste en la adición de los tres colores primarios a los que son sensibles los conos para formar el resto; otra opción es separar los canales de manera que uno de ellos contenga información sobre la luminosidad (dependiente únicamente de la cantidad de luz), y los otros dos su *cromaticidad*. El espacio de color más adecuado para representar una imagen depende del tipo de manipulaciones a las que se le van a someter, siendo su elección en ocasiones crucial para que el resultado de su procesamiento sea perceptiblemente correcto.

A pesar de que existe una gran cantidad de espacios de color, en esta sección solamente se explican los que resultan de interés para los propósitos de este trabajo.

### Espacio RGB

El espacio Red–Green–Blue (RGB) sobrepone diferentes intensidades de rojo, verde, y azul de forma lineal para generar toda la gama de colores. Los valores de cada color están acotados, pudiéndose normalizar para que se encuentren en  $[0, 1]$ , de modo que el espacio tridimensional que conforman es un cubo sólido cuyos vértices  $(0, 0, 0)$  y  $(1, 1, 1)$  corresponden a los colores negro y blanco, respectivamente. En imágenes digitales es usual que, en su lugar, tomen valores discretos entre 0 y  $2^b - 1$  después del proceso de cuantización en  $b$  bits por canal, donde  $b = 8$  es el número de bits más habitual.

La representación RGB de una imagen digital es la más utilizada para su almacenamiento, pero depende de la cromaticidad de los colores primarios y del blanco usado como referencia, que pueden diferir entre estándares [40, pp. 9–13]. Por este motivo, estas dos características deben conocerse previamente para poder manipular y comparar imágenes en este espacio de color. La conversión a escala de grises, por ejemplo, viene dada por

$$\pi_{\text{RGB}} : [0, 1]^3 \rightarrow [0, 1] \\ (r, g, b) \mapsto [w_R \quad w_G \quad w_B] \cdot \begin{bmatrix} r \\ g \\ b \end{bmatrix}, \quad (2.7)$$

donde  $w_R$ ,  $w_G$ , y  $w_B$  son pesos que dependen del estándar RGB utilizado.

El principal inconveniente de este espacio de color es que no es perceptiblemente uniforme, lo que quiere decir que la distancia entre la representación de dos colores arbitrarios no se ajusta a su diferencia percibida por el ojo humano. Por este motivo, no está recomendado su uso para comparar colores, existiendo espacios de color uniformes dedicados para esta tarea.

### Espacios CIELab y CIELuv

La sensibilidad visual a pequeñas diferencias de color es de particular importancia en problemas de comparación y clasificación de regiones de imágenes, ya que la exactitud de sus resultados depende de la percepción humana. Es por ello que es de especial interés trabajar sobre un espacio en el que una perturbación en un punto sea igualmente perceptible independientemente de su color. La Commission Internationale

## 2. MOVIMIENTO EN SECUENCIAS DE IMÁGENES

---

de l'Éclairage (CIE) formula con esa pretensión los espacios de color L\*a\*b\* y L\*u\*v\*, ambos aproximadamente uniformes e independientes de la máquina en la que se visualizan [40, pp. 32–37].

Tanto L\*a\*b\* como L\*u\*v\* aislan la luminosidad del color en una componente L\*, y concentran su cromaticidad en dos componentes cuyo rango de valores se mueve entre colores opuestos (aproximadamente rojo–verde y amarillo–azul). Consecuentemente, su restricción a escala de grises consiste en únicamente el descarte de estas dos últimas y la retención de L\*. Además, dada una imagen  $I: \Omega \rightarrow \mathbb{R}^3$  en uno de estos espacios de colores, la diferencia perceptual  $\Delta E^*$  del color de dos puntos  $\mathbf{x}$  e  $\mathbf{y}$  es la distancia euclídea

$$\Delta E^* = \|I(\mathbf{x}) - I(\mathbf{y})\|_2, \quad (2.8)$$

permitiendo así comparar fácilmente puntos por su color.

No hay diferencias sustanciales entre L\*a\*b\* y L\*u\*v\* en cuanto a su precisión para distinguir colores, lo que hace que ambos coexistan y usualmente se dé preferencia a uno u a otro de manera indistinta [18, p. 591]. Su uso, no obstante, suele darse únicamente en caso de necesitar estrictamente sus propiedades, ya que es computacionalmente intensivo transformar un espacio RGB en cualquiera de ellos.

## 2.2 El flujo óptico

Se llama *flujo óptico* al desplazamiento de cada punto de una secuencia de imágenes en el tiempo [29], que se corresponde con el movimiento aparente en dos dimensiones de la información capturada por un observador. Este observador comúnmente es una cámara, a través de la cual se representan escenas tridimensionales en fotografías planas, por lo que en este caso se puede entender el flujo óptico como la proyección del movimiento real de la escena sobre el plano bidimensional de imagen [14].

Formalmente, el flujo óptico se define como un campo de vectores  $\mathbf{u}: \Omega \rightarrow \mathbb{R}^2$ , donde a cada píxel  $\mathbf{x} = (x, y)$  de una imagen  $I_t$  de una secuencia se le atribuye un único vector  $\mathbf{u}(x, y) = (u(x, y), v(x, y))$  de modo que

$$I_{t+1}(\mathbf{x} + \mathbf{u}(\mathbf{x})) = I_t(\mathbf{x}). \quad (2.9)$$

Es notable que, aunque este campo de vectores está *a priori* definido para todos los píxeles de una imagen, tiene la limitación de poder únicamente describir el movimiento aparente de estos. Por ejemplo, la variación de la iluminación entre imágenes consecutivas puede hacer que no exista una correspondencia exacta para ciertos píxeles, e incluso crear la ilusión de movimiento cuando no lo hay, como en la Figura 2.2. Además, el movimiento de objetos que entran y salen del encuadre no puede ser determinado.

La proyección de los puntos de una escena real al plano de imagen, efectuada al realizar grabaciones con una cámara, no permite capturar la luz que se encuentra tras cuerpos opacos. Por este motivo, es común que partes de objetos de la escena queden ocultas detrás de otras al moverse, a lo que se denomina *occlusiones*. Del mismo modo, a la aparición en una imagen de un objeto que estaba previamente cubierto por otro se le llama *desocclusión*. Ambos efectos conllevan la inexistencia de un vector de flujo para los puntos afectados.

La luminosidad de un punto de un vídeo tiende a variar con el tiempo, así como un mismo color puede darse en varios píxeles de una misma imagen. El cálculo del



**Figura 2.2:** Ilusión de movimiento originada por un cambio de iluminación. La esfera se mantiene estática en toda la secuencia mientras la fuente de luz verde orbita a su alrededor, dando la impresión de que la esfera está rotando. Los vectores de flujo óptico apuntan en el sentido del cambio de color, siguiendo el movimiento aparente y no el real. Texturas creadas con Paint3D.

flujo óptico, pues, no puede realizarse simplemente comparando de manera directa los valores de los píxeles de los fotogramas, y en la práctica es imposible encontrar un campo de vectores que cumpla exactamente la ecuación (2.9) en todos los puntos del dominio de imagen. Como consecuencia, la estimación del flujo generalmente se plantea como un problema de minimización en el que se penaliza la suma de las diferencias de color entre pares de correspondencias.

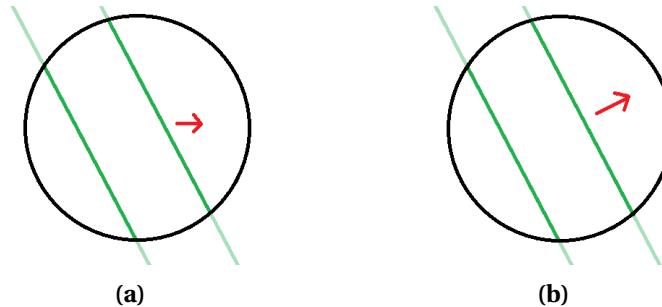
Sean  $I_t, I_{t+1}: \Omega \rightarrow \mathbb{R}^d$ , con  $d \in \mathbb{N}$ , dos imágenes consecutivas de una secuencia de vídeo y  $\phi: \mathbb{R}^d \rightarrow \mathbb{R}$  una función de penalización (generalmente convexa) tal que  $\phi(\mathbf{0}) = 0$  y  $\phi(\mathbf{s}) \geq 0$  para todo  $\mathbf{s} \in \mathbb{R}^d$ . La aplicación (o *energía*)

$$E_{OF}(\mathbf{u}) = \int_{\Omega} \phi(I_{t+1}(\mathbf{x} + \mathbf{u}(\mathbf{x})) - I_t(\mathbf{x})) d\Omega \quad (2.10)$$

es el objetivo de la minimización, cuyo minimizador es la aproximación del flujo óptico real entre las dos imágenes [46, pp. 409–413]. No obstante, el problema (2.10) es subestringido, puesto que el número de mediciones sobre las imágenes (tantas como puntos) es insuficiente para determinar a la vez las componentes horizontal y vertical en las que el flujo se divide (que duplican ese número). Para resolver esta limitación y obtener una solución única para el problema, los métodos de actuación más habituales son calcular las sumas localmente sobre regiones superpuestas (llamadas a veces *parches* o *ventanas*) asumiendo que el movimiento en ellas puede describirse por una función paramétrica, o añadir restricciones adicionales al problema con el fin de regularizar las variables, que son referidas como *terminos de suavizado*.

Una suposición típica para facilitar la minimización del problema (2.10) es considerar que los vectores de flujo son pequeños. Esta suposición, no apropiada para pares arbitrarios de imágenes, es válida en secuencias de vídeo, ya que estos suelen estar grabados con una tasa de más de veinte Fotogramas por Segundo (FPS) (siendo incluso 60 FPS el estándar *de facto* para algunos contenidos multimedia). Una frecuencia alta de fotogramas implica un movimiento reducido entre dos imágenes consecutivas, lo que posibilita linealizar la ecuación de flujo (2.9) con escaso error siempre y cuando la escena no se mueva en exceso. Tomando  $\mathcal{V}(\mathbf{x}, t) = I_t(\mathbf{x})$  como una función de tres variables y un canal, la serie de Taylor del lado izquierdo de la ecuación (2.9) alrededor del punto  $(\mathbf{x}, t)$  lleva a

$$\mathcal{V}(\mathbf{x} + \mathbf{u}(\mathbf{x}), t + 1) = \mathcal{V}(\mathbf{x}, t) + \nabla \mathcal{V}(\mathbf{x}, t) \cdot [\mathbf{u}(\mathbf{x})]_1^T + R_1(\mathbf{u}(\mathbf{x})), \quad (2.11)$$



**Figura 2.3:** Problema de la apertura. Los segmentos paralelos de color verde se desplazan hacia la derecha (a) de tal modo que sus topes quedan ocultos detrás de un objeto opaco con una apertura (indicado con un verde transparente). Como los segmentos son perpendiculares al gradiente de la imagen en todos sus puntos, no se puede determinar la componente del flujo óptico paralela a ellos, sino únicamente la perpendicular (b).

donde  $R_1(\mathbf{u}(\mathbf{x}))$  es el residuo resultante de truncar la serie en el primer término. Como  $R_1(\mathbf{u}(\mathbf{x})) \rightarrow 0$  cuando  $\|\mathbf{u}(\mathbf{x})\| \rightarrow 0$  (de hecho,  $R_1(\mathbf{u}(\mathbf{x})) = o(\|\mathbf{u}(\mathbf{x})\|)$  siguiendo la notación de Landau), la suposición de que hay poco movimiento permite aproximar la ecuación (2.9) como

$$\nabla \mathcal{V}(\mathbf{x}, t) \cdot [\mathbf{u}(\mathbf{x}) \ 1]^T = 0, \quad (2.12)$$

que escrito expandiendo los vectores  $\mathbf{x} = (x, y)$  y  $\mathbf{u}(\mathbf{x}) = (u(x, y), v(x, y))$  resulta en

$$\frac{\partial \mathcal{V}(x, y, t)}{\partial x} \cdot u(x, y) + \frac{\partial \mathcal{V}(x, y, t)}{\partial y} \cdot v(x, y) + \frac{\partial \mathcal{V}(x, y, t)}{\partial t} = 0. \quad (2.13)$$

La ecuación (2.13) tiene a  $u$  y  $v$  por incógnitas, mientras que las derivadas parciales que aparecen en ella son calculables. Ahora bien, esta ecuación es indeterminada y es imposible medir la componente del flujo perpendicular al gradiente de la imagen. En efecto, suponiendo que el vector  $\mathbf{u} = (u, v) \in \mathbb{R}^2$  satisface la ecuación de flujo (2.12) de  $\mathcal{V}(\mathbf{x}, t)$  a  $\mathcal{V}(\mathbf{x}, t+1)$ , si  $\mathbf{u}' = (u', v') \in \mathbb{R}^2$  es otro vector ortogonal al gradiente de la imagen en el punto (es decir,  $\nabla \mathcal{V}(\mathbf{x}, t) \cdot [\mathbf{u}' \ 0]^T = 0$ ), entonces

$$\begin{aligned} \nabla \mathcal{V}(\mathbf{x}, t) \cdot \begin{bmatrix} \mathbf{u} + \mathbf{u}' \\ 1 \end{bmatrix} &= \nabla \mathcal{V}(\mathbf{x}, t) \cdot \begin{bmatrix} \mathbf{u} \\ 1 \end{bmatrix} + \nabla \mathcal{V}(\mathbf{x}, t) \cdot \begin{bmatrix} \mathbf{u}' \\ 0 \end{bmatrix} \\ &= 0, \end{aligned} \quad (2.14)$$

de donde resulta que  $\mathbf{u} + \mathbf{u}'$  también lo satisface. Esta propiedad conlleva que haya ambigüedad al representar movimiento a lo largo de bordes, sobre los cuales no se puede calcular la magnitud del flujo paralelo a ellos. Un ejemplo de esto último es el llamado *problema de la apertura*, que queda ilustrado en la Figura 2.3.

En lugar de tratar cada vector de flujo de forma independiente, una estrategia para estimarlo consiste en suponer que existe cierta coherencia espacial en las imágenes y que por tanto píxeles cercanos en un fotograma  $f$  se mueven de manera idéntica. Lucas y Kanade [33] proponen utilizar ventanas alrededor de determinados puntos de la imagen, asumiendo que el flujo óptico es constante en un entorno de los puntos

considerados. Una ventana de  $n$  píxeles permite llevar la ecuación (2.13) a un sistema de ecuaciones lineal sobre determinado

$$\underbrace{\begin{bmatrix} \frac{\partial \mathcal{V}(x_1, y_1, f)}{\partial x} & \frac{\partial \mathcal{V}(x_1, y_1, f)}{\partial y} \\ \frac{\partial \mathcal{V}(x_2, y_2, f)}{\partial x} & \frac{\partial \mathcal{V}(x_2, y_2, f)}{\partial y} \\ \vdots & \vdots \\ \frac{\partial \mathcal{V}(x_n, y_n, f)}{\partial x} & \frac{\partial \mathcal{V}(x_n, y_n, f)}{\partial y} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} \frac{\partial \mathcal{V}(x_1, y_1, f)}{\partial t} \\ \frac{\partial \mathcal{V}(x_2, y_2, f)}{\partial t} \\ \vdots \\ \frac{\partial \mathcal{V}(x_n, y_n, f)}{\partial t} \end{bmatrix}}_{\mathbf{b}}, \quad (2.15)$$

para el cual, pese a que generalmente no tiene una solución exacta, se puede encontrar un vector  $\mathbf{u}$  que minimice la diferencia  $\|\mathbf{Au} - \mathbf{b}\|_2^2$ . Este vector debe cumplir que

$$\begin{aligned} & \frac{d}{d\mathbf{u}} \|\mathbf{Au} - \mathbf{b}\|_2^2 = \mathbf{0} \\ \iff & \frac{d}{d\mathbf{u}} [(\mathbf{Au} - \mathbf{b})^T (\mathbf{Au} - \mathbf{b})] = \mathbf{0} \\ \iff & \frac{d}{d\mathbf{u}} [\mathbf{u}^T \mathbf{A}^T \mathbf{A} \mathbf{u} - \mathbf{u}^T \mathbf{A}^T \mathbf{b} - \mathbf{b}^T \mathbf{A} \mathbf{u} + \mathbf{b}^T \mathbf{b}] = \mathbf{0} \\ \iff & 2(\mathbf{A}^T \mathbf{A} \mathbf{u} - \mathbf{A}^T \mathbf{b}) = \mathbf{0} \\ \iff & \mathbf{A}^T \mathbf{A} \mathbf{u} = \mathbf{A}^T \mathbf{b}, \end{aligned} \quad (2.16)$$

donde tanto  $\mathbf{A}^T \mathbf{A}$  como  $\mathbf{A}^T \mathbf{b}$  son matrices de dimensión  $2 \times 2$ . Utilizando la notación  $\mathcal{V}_i = \mathcal{V}(x_i, y_i, f)$ , el sistema (2.16) queda escrito como

$$\underbrace{\begin{bmatrix} \sum_{i=1}^n \left( \frac{\partial \mathcal{V}_i}{\partial x} \right)^2 & \sum_{i=1}^n \frac{\partial \mathcal{V}_i}{\partial x} \frac{\partial \mathcal{V}_i}{\partial y} \\ \sum_{i=1}^n \frac{\partial \mathcal{V}_i}{\partial x} \frac{\partial \mathcal{V}_i}{\partial y} & \sum_{i=1}^n \left( \frac{\partial \mathcal{V}_i}{\partial y} \right)^2 \end{bmatrix}}_{\mathbf{A}^T \mathbf{A}} \begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} \sum_{i=1}^n \frac{\partial \mathcal{V}_i}{\partial x} \frac{\partial \mathcal{V}_i}{\partial t} \\ \sum_{i=1}^n \frac{\partial \mathcal{V}_i}{\partial y} \frac{\partial \mathcal{V}_i}{\partial t} \end{bmatrix}}_{\mathbf{A}^T \mathbf{b}}. \quad (2.17)$$

Aunque el sistema (2.17) tiene solución cuando  $\mathbf{A}^T \mathbf{A}$  es invertible, en la práctica esta condición no es suficiente para alcanzar un resultado porque los puntos de la ventana pueden estar mal condicionados. Un estudio de los valores propios de la matriz ayuda a identificar cuándo el flujo óptico puede calcularse. Sean  $\lambda_1, \lambda_2$  estos valores, reales y no negativos al ser  $\mathbf{A}^T \mathbf{A}$  simétrica y semidefinida positiva, sus vectores propios asociados apuntan en la dirección de máximo cambio de intensidad y a su orthogonal. Suponiendo  $\lambda_1 \geq \lambda_2$ , entonces:

- Si  $\lambda_1$  y  $\lambda_2$  son ambos muy cercanos a 0,  $\mathbf{A}^T \mathbf{A}$  es a efectos prácticos una matriz de rango 0. Al no haber cambios sustanciales de intensidad en el parche, es imposible determinar ninguna componente del flujo óptico. Geométricamente, los puntos se encuentran sobre una superficie *plana* o *sin textura*.
- Si  $\lambda_1 \gg \lambda_2$ ,  $\mathbf{A}^T \mathbf{A}$  se comporta como una matriz de rango 1. Se puede determinar la componente del flujo óptico en la dirección del vector propio asociado a  $\lambda_1$ , pero se produce el problema de la apertura. Geométricamente, la ventana se halla sobre una recta o un borde.

## 2. MOVIMIENTO EN SECUENCIAS DE IMÁGENES

---

- Si  $\lambda_1 \gg 0, \lambda_2 \gg 0$ , y sus ordenes de magnitud son parecidos,  $\mathbf{A}^T \mathbf{A}$  tiene rango completo y se pueden obtener las dos componentes del flujo óptico. Geométricamente, el parche se sitúa sobre una esquina o una zona con *muchas texturas*.

A la matriz  $\mathbf{A}^T \mathbf{A}$ , que aporta esta información sobre la configuración de los píxeles en un entorno, se le denomina *tensor de estructura* o *matriz de segundos momentos*.

La principal desventaja de los métodos locales como el propuesto por Lucas y Kanade radica en que sus resultados son pobres sobre zonas uniformes. Asimismo, el uso de agrupaciones de píxeles y la hipótesis de coherencia espacial pueden ser inadecuados cerca de discontinuidades de movimiento, como en la frontera entre un objeto y el fondo de la imagen. Estos dos motivos causan que estos métodos no produzcan campos densos de vectores como solución al problema de flujo (2.10). En su lugar, Horn y Schunck [19] expresan el problema como la minimización global de

$$E_{OF-HS}(\mathbf{u}) = \int_{\Omega} \left[ (\nabla \mathcal{V}(\mathbf{x}, f) \cdot [\mathbf{u}(\mathbf{x})]^T)^2 + \alpha^2 \|\nabla \mathbf{u}(\mathbf{x})\|_2^2 \right] d\Omega, \quad (2.18)$$

usando  $\alpha \in \mathbb{R}$  como un parámetro de regularización con el cometido de balancear la restricción de flujo con el término de suavizado. Esta formulación incentiva que se minimice el cambio en el flujo óptico entre puntos cercanos, lo que en zonas sin textura significa asignar un vector constante; y en bordes, el vector más corto en la dirección del gradiente. El campo de vectores resultante, por tanto, especifica un vector de flujo para cada píxel de la imagen.

La energía (2.18) es convexa porque su integrando es la suma de un término lineal y una norma, ambos convexos. Además, su mínimo global se alcanza en la solución de su ecuación de Euler–Lagrange asociada: empleando  $\mathcal{V}_x, \mathcal{V}_y$ , y  $\mathcal{V}_t$  para denotar las derivadas parciales de  $\mathcal{V}$  en un punto  $\mathbf{x}$  y fotograma  $f$ , y siendo  $\Delta = \nabla^2$  el operador de Laplace, su minimización consiste en la resolución del sistema de ecuaciones en derivadas parciales elípticas de segundo orden

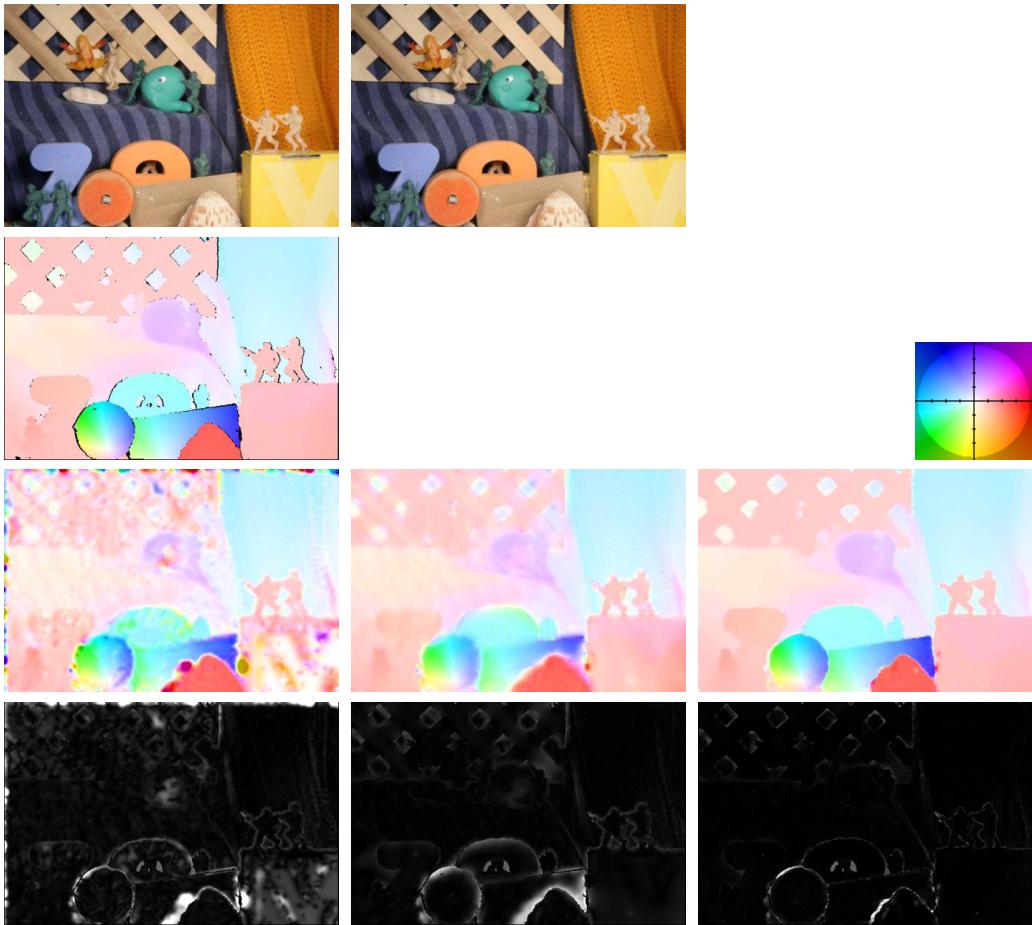
$$\begin{bmatrix} \mathcal{V}_x^2 & \mathcal{V}_x \mathcal{V}_y \\ \mathcal{V}_x \mathcal{V}_y & \mathcal{V}_y^2 \end{bmatrix} \mathbf{u}(\mathbf{x}) + \begin{bmatrix} \mathcal{V}_x \mathcal{V}_t \\ \mathcal{V}_y \mathcal{V}_t \end{bmatrix} = \alpha^2 \Delta \mathbf{u}(\mathbf{x}) \quad (2.19)$$

con determinadas condiciones de contorno, que tiene solución única [43] y puede obtenerse de manera numérica. Dichas restricciones en la frontera usualmente se escogen de manera que el flujo (condición de Dirichlet) o su gradiente (condición de Neumann) se anulen en ella, aunque pueden variar con el método numérico.

Los modelos globales con restricciones de suavizado pueden generalizarse añadiendo a la energía (2.10) un término de penalización en función de las derivadas del flujo óptico, dando lugar a

$$E_{OF-Smooth}(\mathbf{u}) = \int_{\Omega} [\phi(I_{f+1}(\mathbf{x} + \mathbf{u}(\mathbf{x})) - I_f(\mathbf{x})) + \alpha^2 \psi(\mathbf{u}, \nabla \mathbf{u}, \dots)] d\Omega. \quad (2.20)$$

En particular, la energía (2.18) de Horn–Schunck corresponde a  $\phi(s) = s^2$  y  $\psi(\nabla \mathbf{u}) = \|\nabla \mathbf{u}\|_2^2$ , previamente habiendo linealizado la restricción de flujo, y usando las imágenes  $I_f, I_{f+1}$  en escala de grises. A pesar de que esta última tiene la ventaja de ser fácil de optimizar debido a su convexidad y diferenciabilidad, en la práctica no está libre de inconvenientes: el error de linealización crece considerablemente en desplazamientos largos, y el término de regularización cuadrático suaviza en exceso las estimaciones de



**Figura 2.4:** Visualización de las estimaciones del flujo óptico en la secuencia ARMY de la base de datos de Middlebury. Primera fila: imágenes de la secuencia. Segunda fila: valores reales del flujo óptico (izquierda), codificación de colores de los vectores (derecha). Tercera fila: estimación de Lucas–Kanade (izquierda), Horn–Schunck (centro), y TV– $L^1$  (derecha). Cuarta fila: error angular de Lucas–Kanade (izquierda), Horn–Schunck (centro), y TV– $L^1$  (derecha).

flujo, esencialmente provocando que no existan discontinuidades en los contornos de objetos en movimiento. De igual forma, se ha observado que es más sensible al ruido que los métodos locales, ya que este provoca gradientes elevados que se penalizan fuertemente [8]. Es por estos motivos que los métodos globales han evolucionado hacia el uso de restricciones más robustas, como el suavizado con norma  $L^1$ . Un ejemplo de uno de ellos es el Total Variation– $L^1$  (TV– $L^1$ ) [50], que utiliza  $\phi(s) = |s|$  y  $\psi(\nabla \mathbf{u}) = \|\nabla \mathbf{u}\|_1$  junto con un esquema numérico para minimizar la energía sin necesidad de que esta sea diferenciable. En la Figura 2.4 se muestra una comparación de las estimaciones de flujos ópticos obtenidas con los algoritmos de Lucas–Kanade, Horn–Schunck, y TV– $L^1$ , donde se percibe que el primero falla en zonas uniformes, el segundo no delimita bien los bordes de objetos, y el tercero mejora los resultados de los otros dos.

Desde que se presentaron los primeros algoritmos de estimación del flujo óptico, estos han ido perfeccionándose tanto en precisión como en complejidad computacio-

nal [30]. Se ha dado paso a un uso más prominente del color en contrapartida al predominio inicial de métodos centrados en escala de grises, así como también se ha tomado una dirección hacia el desarrollo de modelos capaces de hacer frente a largos desplazamientos [7]. Finalmente, el departamento de visión de la Universidad de Middlebury ha definido un estándar de evaluación de estos algoritmos [3], proporcionando una base de datos y métricas para determinar la fidelidad de sus estimaciones en relación a los valores reales, además de una clasificación que permite comparar los resultados de varios de ellos en diferentes secuencias de vídeo.

## 2.3 Trayectorias de puntos

El objetivo fundamental de la estimación del flujo óptico es conocer el movimiento que sucede entre imágenes. Su cálculo, no obstante, suele estar limitado a pares de fotogramas sirviendo de origen y destino, de modo que para definir el movimiento en una secuencia es necesario un campo de vectores de flujo por cada dos imágenes sucesivas. Este conjunto de campos no proporciona una información global, sino una lista de desplazamientos en períodos de tiempo muy cortos, que individualmente no distinguen sus puntos por su procedencia. Asimismo, la presencia de ruido, occlusiones y desocclusiones, cambios de iluminación, y el propio error de medición provocan que algunos vectores de flujo no sean fiables. Es por esta ambigüedad que el movimiento real que se produce en una escena a lo largo del tiempo no puede ser definido únicamente por sus campos de flujo óptico. La propiedad que les falta es la coherencia temporal, alcanzable por medio de la definición de trayectorias de puntos.

Dado un vídeo  $\mathcal{V} : \Omega \times [1, T] \rightarrow \mathbb{R}^d$ , una trayectoria  $c : [t_1, t_2] \rightarrow \Omega$  se define como una curva sobre la superficie  $\Omega$  que describe el desplazamiento de un determinado punto a lo largo de un período de tiempo  $[t_1, t_2] \subseteq [1, T]$ . El conjunto  $\mathcal{C}$  de todas las trayectorias del vídeo registra el movimiento de cada uno de los puntos de la secuencia desde que aparece por primera vez (o se desocluye) hasta que se ocluye o no quedan más fotogramas. Según esta definición, un punto que se ocluye y desocluye más adelante requiere de más de una trayectoria para explicar su movimiento.

En la práctica, determinar las trayectorias que siguen todos los puntos de un vídeo es inefficiente y poco conveniente. Ciertos puntos pueden ser difíciles de seguir, así como no todas las zonas de una imagen aportan información útil cuando se las rastrea a lo largo del tiempo. Shi y Tomasi argumentan en [20] que el seguimiento de puntos inadecuados puede resultar inservible en el mejor de los casos y perjudicial en el peor, dependiendo del propósito de este. Por ese motivo, la estimación de trayectorias suele efectuarse sobre un conjunto disperso de píxeles bien condicionados.

### 2.3.1 Puntos de interés

Sea  $x \in \Omega$  un punto sobre el fotograma  $t_1$  de un vídeo  $\mathcal{V}$  y  $H$  el tensor de estructura alrededor de  $x$ . En la Sección 2.2 se ha mostrado que si los valores propios  $\lambda_1, \lambda_2$  de este último son grandes y de un orden de magnitud parecido,  $x$  presenta una estructura con mucha textura en su vecindad. Son precisamente estos puntos los más apropiados para ser rastreados. Generalmente, la variación de intensidad de una imagen está acotada por un valor máximo de píxel, por lo que el valor propio más grande de  $H$  no puede ser

arbitrariamente elevado; como consecuencia, es suficiente que

$$\min(\lambda_1, \lambda_2) > \tau \quad (2.21)$$

para que  $\mathbf{x}$  sea un punto de interés, donde  $\tau > 0$  es un umbral suficientemente grande en el contexto del rango de valores que puede tomar la imagen.

Un punto con mucha textura en su fotograma de origen puede demostrar no ser un punto de interés con el paso del tiempo. Por ejemplo, el cruce de dos barras situadas a distintas profundidades tiene inicialmente una estructura de esquina, pero dicha intersección no existe en la escena real. Un cambio de perspectiva debido a un movimiento de la cámara modificaría su posición, enviándolo a un lugar equivocado. Detectar este tipo de sucesos obliga a terminar la trayectoria para evitar la acumulación de errores, como también debe hacerse en caso de oclusión.

El *tracker* Kanade–Lucas–Tomasi (KLT) [47] consiste en un esquema que combina la búsqueda de puntos bien condicionados junto con el uso del flujo óptico local de Lucas–Kanade para rastrearlos individualmente. Su funcionamiento se basa en la extensión de trayectorias hasta su desaparición:

1. KLT realiza una selección de  $N$  puntos  $\{\mathbf{x}_i\}_{i=1}^N \subset \Omega$  en un primer fotograma  $t_1$  aplicando el criterio (2.21) a una ventana deslizante de pocos píxeles. Esta selección inicializa las trayectorias  $\{c_i\}_{i=1}^N$  de manera que  $c_i(t_1) = \mathbf{x}_i$  para  $i \in \{1, \dots, N\}$ .
2. El sistema (2.17) es resuelto para dichos puntos, obteniéndose sus vectores de flujo óptico  $\{\mathbf{u}_i \mid \mathbf{u}_i = \mathbf{u}(\mathbf{x}_i)\}_{i=1}^N$  y los nuevos puntos  $\{\mathbf{y}_i \mid \mathbf{y}_i = \mathbf{x}_i + \mathbf{u}_i\}_{i=1}^N$  pertenecientes a la siguiente imagen.
3. El paso anterior se repite para las subsiguientes imágenes  $t_1 + 1, t_1 + 2, \dots$  de la secuencia con los últimos puntos obtenidos. A medida que este paso se lleva a cabo, se compara el nivel de gris de los píxeles con sus valores al inicio de la trayectoria. Si la disimilitud no ha crecido pasado un umbral, se enlaza el nuevo punto a la trayectoria; de lo contrario, esta termina sin añadirle el nuevo punto.
4. Para compensar el abandono de trayectorias, se inicializan nuevas cada cierto número de fotogramas. El mismo proceso se aplica sobre ellas, siempre manteniendo presente su intensidad inicial.

A los puntos de la selección que toma KLT se les llama comúnmente *rasgos (features)*, ya que son las localizaciones más distintivas de una secuencia. Cabe notar que el número de estos rasgos suele ser órdenes de magnitud más pequeño el tamaño de una imagen.

### 2.3.2 Trayectorias semidensas

KLT es preciso y computacionalmente rápido; sin embargo, en determinadas tareas como la segmentación en vídeo, el número de trayectorias que traza es insuficiente. El seguimiento de un conjunto disperso de rasgos debe ser sustituido por el de una cantidad mucho más elevada de puntos, que no puede desempeñarse por un método de flujo óptico local como el de Lucas–Kanade. Como remedio se encuentra el uso de métodos variacionales, cuyo suavizado global facilita el rastreo en zonas sin estructura de esquina a cambio de una mayor complejidad computacional. Un *tracker* semidenso se caracteriza por el uso de estos métodos para determinar las trayectorias de una

## 2. MOVIMIENTO EN SECUENCIAS DE IMÁGENES

---



**Figura 2.5:** Estimación de trayectorias de puntos a lo largo de una secuencia de cinco imágenes. Primera fila: primer fotograma de la secuencia (izquierda), rasgos que rastrea KLT (centro), y puntos que sigue LDOF (derecha). Segunda fila: quinto fotograma de la secuencia (izquierda), rasgos de KLT (centro), puntos de LDOF (derecha). Únicamente están marcadas las trayectorias que se hallan sobre la jugadora.

muestra de puntos distribuida uniformemente en el espacio, que cubre un porcentaje importante de los píxeles que este último contiene.

Sundaram, Brox, y Keutzer plantean en [45] un modelo de *tracker* semidenso capaz de capturar desplazamientos largos de forma precisa. Inicialmente pensado para funcionar en base a Large Displacement Optical Flow (LDOF) [7], su uso junto con otros algoritmos de flujo modernos ha manifestado ser igualmente apto. La Figura 2.5, recuperada de [45, Fig. 4], expone los resultados de la aplicación de este modelo (ini-ciando el rastreo en todos los píxeles) sobre una secuencia de cinco fotogramas, y los compara con los de KLT, haciéndose evidente la diferencia en el número de trayectorias que abarca.

Dado un vídeo  $\mathcal{V}: \Omega \times [1, T] \rightarrow \mathbb{R}^d$ , el cálculo de trayectorias de [45] comienza con la inicialización de puntos en el primer fotograma  $t_1$ . Al contrario que KLT, no es necesario buscar puntos de interés: es habitual tomar una muestra uniforme de píxeles, aunque teóricamente es posible inicializarlos todos debido a la densidad del flujo óptico. Aun así, entre ellos puede haber puntos problemáticos que se encuentren en áreas faltas de estructura, por lo que estos se descartan y no se rastrean. Sea  $I: \Omega \rightarrow \mathbb{R}^d$  la imagen en cuestión, de modo que  $I(\mathbf{x}) = (I_1(\mathbf{x}), \dots, I_d(\mathbf{x})) = \mathcal{V}(\mathbf{x}, t_1)$  para todo  $\mathbf{x} \in \Omega$ , el *tracker* suprime los puntos para los que el valor propio más pequeño de

$$\mathbf{H}_\rho(\mathbf{x}) = \left( \mathfrak{G}_\rho * \sum_{k=1}^d \nabla I_k \nabla I_k^T \right)(\mathbf{x}) \quad (2.22)$$

es menor que una porción de la media de este en toda la imagen. La expresión (2.22) es una generalización del tensor de estructura que tiene en consideración que la imagen puede ser de varios canales, y que difumina los valores de los segundos momentos por

medio de la convolución con un filtro Gaussiano bidimensional

$$\mathfrak{G}_\rho(\mathbf{x}) = \frac{1}{2\pi\rho^2} e^{-\frac{\|\mathbf{x}\|_2^2}{2\rho^2}}, \quad (2.23)$$

que dota de robustez al ruido. El valor de la desviación típica del filtro sugerido en [45] es de  $\rho = 1$ , mientras que la proporción aceptable del valor propio puede variar según el número de trayectorias deseadas y su proximidad a los bordes de la imagen.

Sea  $\mathbf{u}_t: \Omega \rightarrow \mathbb{R}^2$  el campo de vectores de flujo para un fotograma  $t \in [1, T]$  hacia  $t + 1$ , cada una de las trayectorias  $\{c_i\}$  puede ser continuada rastreando sus puntos por medio de la recurrencia

$$c_i(t+1) = c_i(t) + \mathbf{u}_t(c_i(t)). \quad (2.24)$$

En el caso de imágenes digitales, donde  $c_i(t)$  puede tomar un valor situado entre varios píxeles, se utiliza interpolación bilineal para inferir su flujo, lo que ofrece una mayor precisión en comparación con aproximar al píxel más cercano.

En la igualdad (2.24), si el punto  $c_i(t+1)$  no cae sobre la superficie  $\Omega$ , se considera que el flujo es erróneo y se detiene su seguimiento. De la misma manera, es importante suspender una trayectoria en el momento en que un punto se oculta, para que no comparta el movimiento de dos objetos distintos. En lugar de comparar la apariencia de un entorno de los puntos a lo largo de la trayectoria que forman, en [45] se opta por verificar la consistencia del flujo óptico respecto de su estimación en el sentido opuesto. Sea  $\hat{\mathbf{u}}_t: \Omega \rightarrow \mathbb{R}^2$  el flujo que va desde la imagen  $t+1$  a la  $t$ , los vectores de flujo hacia adelante y hacia atrás de un punto  $\mathbf{x}$  que no se oculta deben ser inversos:

$$\mathbf{u}_t(\mathbf{x}) = -\hat{\mathbf{u}}_t(\mathbf{x} + \mathbf{u}_t(\mathbf{x})). \quad (2.25)$$

Si la ecuación (2.25) no se cumple para determinado punto, entonces o bien está siendo ocluido en el fotograma  $t+1$ , o bien la estimación de su flujo óptico presenta errores; ambos casos justifican que deje de ser rastreado en el tiempo  $t$ . Como en la práctica siempre se producen pequeños errores en la estimación del flujo óptico, el *tracker* mantiene una trayectoria  $c$  cuando

$$\|\mathbf{u}_t(c(t)) + \hat{\mathbf{u}}_t(c(t+1))\|_2^2 < 0.1 \cdot (\|\mathbf{u}_t(c(t))\|_2^2 + \|\hat{\mathbf{u}}_t(c(t+1))\|_2^2) + 0.5, \quad (2.26)$$

dando así un margen de error que crece con la magnitud del desplazamiento.

Además de las occlusiones, un último problema que el *tracker* mitiga son los errores causados por discontinuidades de movimiento. La localización exacta de estas puede fluctuar debido al suavizado global de la estimación del flujo óptico, creando la posibilidad de que un punto cercano a una de ellas se desvíe al aplicar la fórmula (2.24) y cruce la frontera que esta delimita. Esto ocasiona que su trayectoria englobe incorrectamente el movimiento de varios objetos. La solución, nuevamente, es interrumpir aquellas que muestran indicios de encontrarse en esa situación. Por este motivo, el *tracker* establece la condición

$$\|\nabla \mathbf{u}_t(c(t))\|_2^2 < 0.01 \cdot \|\mathbf{u}_t(c(t))\|_2^2 + 0.002 \quad (2.27)$$

para toda trayectoria  $c$  en tiempo  $t$ , que impone que la variación de su flujo no supere un valor de tolerancia relativo a su magnitud, pasado el cual pueda considerarse que se encuentra sobre una discontinuidad.

## 2. MOVIMIENTO EN SECUENCIAS DE IMÁGENES

---

Debido a la cantidad de trayectorias que el *tracker* rastrea, es usual que las restricciones (2.26) y (2.27) conlleven la finalización de una parte significativa de ellas en cada cambio de imagen de la secuencia. Las occlusiones van habitualmente acompañadas de desocclusiones, así como una trayectoria que se pierde ha de reponerse. Con la finalidad de que las zonas del vídeo que se ven sometidas a dichos acontecimientos sigan sondándose, pues, en cada fotograma se incorporan nuevas trayectorias sobre puntos ubicados a un radio de más de  $R$  píxeles de su trayectoria más cercana, siempre y cuando posean textura suficiente acorde a los valores propios de la matriz (2.22).

Como material complementario a este trabajo se proporciona una implementación en C++ del algoritmo presentado en esta sección, que puede ejecutarse en conjunto con cualquier método de flujo óptico global. En la Figura 2.6 se exhibe un ejemplo de su aplicación sobre la secuencia BMX-BUMPS de la base de datos Densely Annotated Video Segmentation (DAVIS) [39], en el que las trayectorias se inicializan cada  $R = 8$  píxeles en horizontal y vertical, y se continúan mediante un flujo óptico TV- $L^1$ . Se puede observar en ella que los puntos sobre gran parte de la arena son descartados por no disponer de suficiente textura, y que las occlusiones y desocclusiones suponen la eliminación y reposición de trayectorias, respectivamente. Por último, es destacable que dos de las trayectorias que comienzan sobre la espalda del ciclista no consiguen corregirse con la restricción (2.27) y sufren un desvío hacia el fondo; el resto de ellas identifica el movimiento del vídeo a largo plazo con precisión.



**Figura 2.6:** Trayectorias identificadas por la implementación del *tracker* semidenso de este trabajo. Fila 1: fotogramas 1, 6, y 11 de la secuencia BMX-BUMPS de la base de datos DAVIS. Filas 2–4: totalidad de las trayectorias visibles en los fotogramas (izquierda), trayectorias que en el fotograma 1 están situadas sobre el ciclista (derecha). Las flechas que salen de los puntos indican el sentido y la magnitud de su flujo óptico. Imágenes recortadas a relación de aspecto 4:3.



CAPÍTULO



## SEGMENTACIÓN DE TRAYECTORIAS

La segmentación de imágenes es un problema de clasificación que busca separar sus píxeles en agrupaciones disjuntas correspondientes a distintas entidades. La forma de asociar dichos puntos es subjetiva, dependiendo tanto de la visión humana como del propósito de las capturas. Cuando las imágenes forman una secuencia, el movimiento que se manifiesta entre ellas asiste en distinguir las zonas de más interés; es por ello que el seguimiento de trayectorias de puntos se presenta como un recurso valioso que se puede utilizar para segmentar vídeos. En este capítulo se estudian diferentes formas de clasificar las entidades que aparecen en un vídeo a partir de la segmentación de su conjunto de trayectorias.

El *fondo* (*background*) de un vídeo es el conjunto de todos los puntos que no pertenecen al primer plano (*foreground*) de sus fotogramas, y que por tanto tienen un menor interés visual respecto al resto. Por otro lado, a toda entidad sobre la que ha de recaer la atención de un observador se le llama *objeto*. En una grabación ordinaria, que algo sea considerado fondo u objeto depende de la intención de quien la realiza, que no se puede conocer disponiendo únicamente de una secuencia de imágenes. Por este motivo, es necesario establecer ciertas suposiciones sobre el comportamiento del fondo en relación a los objetos. La más importante de ellas, y el pilar fundamental de los algoritmos presentados en este trabajo, es que el fondo tiene un movimiento coherente en todos sus puntos, diferente del de los objetos del vídeo, y mayoritario en la escena.

Un algoritmo de segmentación de vídeo que utiliza solamente la información que puede extraer de una secuencia y clasifica las entidades según nociones preconcebidas de su aspecto, textura, y comportamiento se denomina *no supervisado*. Mientras que estos métodos son capaces de agrupar puntos fielmente a la realidad bajo las correctas circunstancias, esta información no siempre es suficiente para llegar a resultados significativos. Por ejemplo, en una grabación de un coche estacionado donde la cámara rota a su alrededor, es evidente que este es el objeto principal a segmentar, pero su movimiento no es distintivo del del fondo; asimismo, en un vídeo en el que aparecen varios animales, es posible que solamente uno de ellos sea de interés y el resto haya de considerarse como fondo. Una solución para corregir ambigüedades en estos ca-

### 3. SEGMENTACIÓN DE TRAYECTORIAS

---

sos es aceptar información adicional proporcionada por un humano, en cuyo caso el algoritmo es *interactivo*.

No existe una única estrategia para lograr una segmentación correcta, por lo que en este capítulo se explican dos maneras conceptualmente distintas de realizar el proceso. En la Sección 3.1 se presenta un método de segmentación no supervisado que modela geométricamente el fondo de un vídeo para identificar los puntos que no pertenecen a él, y llegar a una segmentación binaria. Por otra parte, en la Sección 3.2 se interpreta el conjunto de trayectorias como un grafo y se plantea un método de segmentación interactivo basado en la partición de sus nodos, que es capaz de diferenciar objetos entre sí. Los resultados que ambos algoritmos producen se limitan a clasificar los puntos por los que pasa una trayectoria, por lo que no cubren la totalidad de los píxeles de las imágenes. En el caso de requerir una segmentación densa, en el Capítulo 5 se expone cómo obtenerla a partir de ellos.

## 3.1 Modelado geométrico de fondo

Una de las maneras más sencillas de segmentar objetos en una secuencia de imágenes es modelar geométricamente el movimiento del fondo, y considerar como objeto todo punto que no se mueve como él. Habitualmente, el desplazamiento de los puntos del fondo de un vídeo depende únicamente de los cambios de posición y orientación de la cámara, manteniéndose estáticos o con un movimiento rígido en la escena tridimensional. Con esta suposición, dado un vídeo  $\mathcal{V}: \Omega \times [1, T] \rightarrow \mathbb{R}^d$  y los sistemas de referencia explicados en la Sección 2.1.1, un punto  $\mathbf{X} = (X, Y, Z)$  del fondo de la escena se proyecta sobre los fotogramas  $t$  y  $t + 1$  en los respectivos píxeles  $\mathbf{x} = (x, y)$  y  $\mathbf{x}' = (x', y')$  a través de las expresiones

$$\begin{bmatrix} sx \\ sy \\ s \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad \begin{bmatrix} s'x' \\ s'y' \\ s' \\ 1 \end{bmatrix} = \mathbf{P}' \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (3.1)$$

donde  $\mathbf{P}$  y  $\mathbf{P}'$  son las matrices de proyección de la cámara en los tiempos  $t$  y  $t + 1$ , cuya forma se puede ver en la ecuación (2.6). Ambas matrices son de dimensión  $3 \times 4$ , por lo que únicamente a partir de las expresiones de (3.1) no es posible establecer una relación biúnica entre  $\mathbf{x}$  y  $\mathbf{x}'$ . Bajo determinadas circunstancias, sin embargo, el fondo de una escena se comporta como un plano en el espacio: si se encuentra lejos de la cámara, o si está formado por superficies lisas (como muros), se considera un fondo *planar* y no se requiere de una de sus componentes espaciales para describirlo [48].

Introduciendo la hipótesis de planaridad del fondo, y suponiendo que este viene dado por la ecuación  $Z = aX + bY + c$  con  $a, b, c \in \mathbb{R}$ , se puede escribir

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a & b & c \\ 0 & 0 & 1 \end{bmatrix}}_C \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (3.2)$$

siempre y cuando dicho plano no sea paralelo al eje de  $Z$  (en cuyo caso se utiliza  $X = aY + bZ + c$  ó  $Y = aX + bZ + c$ ), expresión que combinada con (3.1) conduce a

$$\begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \mathbf{P}\mathbf{C} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}, \quad \begin{bmatrix} s'x' \\ s'y' \\ s' \end{bmatrix} = \mathbf{P}'\mathbf{C} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}. \quad (3.3)$$

Las ecuaciones de (3.3) corresponden a la proyección de un plano sobre otro (el fondo sobre la imagen), que es una aplicación biyectiva. Consecuentemente, las matrices  $\mathbf{P}\mathbf{C}$  y  $\mathbf{P}'\mathbf{C}$  son no singulares. En concreto, son las matrices asociadas a proyectividades de un espacio en sí mismo, llamadas *homografías*. Sabiendo que existen sus inversas, tomando  $\bar{s} = s'/s$ , de (3.3) es inmediato que

$$\begin{bmatrix} \bar{s}x' \\ \bar{s}y' \\ \bar{s} \end{bmatrix} = \underbrace{(\mathbf{P}'\mathbf{C})(\mathbf{P}\mathbf{C})^{-1}}_{\mathbf{H}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (3.4)$$

donde  $\mathbf{H}$  es también la matriz de una homografía, porque el conjunto de todas ellas forma un grupo con la composición [23, pp. 99–100].

El conjunto de las  $T - 1$  homografías que hacen cumplir la ecuación (3.4) para los puntos del fondo de cada par de fotogramas consecutivos de  $\mathcal{V}$  es su *modelo de movimiento*. Es importante notar que en la práctica es difícil encontrar imágenes cuyo fondo sea rígido y completamente planar, lo que supone que este modelo sea únicamente una aproximación de su movimiento global en gran parte de los casos, e incluso pueda ser inválido en videos que no satisfagan las hipótesis planteadas. Aun así, en [48] se argumenta que la mayoría de videos de interés para segmentar son modelables de manera precisa a través de él, siendo su obtención el primer paso propuesto para la segmentación de objetos por modelado de fondo.

### 3.1.1 Homografías entre fotogramas

La transformación geométrica que sufre un fondo planar entre dos fotogramas, dada por la matriz  $\mathbf{H}$  vista en la ecuación (3.4), solamente es calculable si se disponen de suficientes pares de correspondencias de puntos entre ambas imágenes para determinar todas sus entradas. Sean  $\mathbf{x} = [x \ y]^T$  y  $\mathbf{x}' = [x' \ y']^T$  dos puntos correspondientes, y  $\tilde{\mathbf{x}} = [\mathbf{x}^T \ 1]^T$ ,  $\tilde{\mathbf{x}}' = [\mathbf{x}'^T \ 1]^T$  sus respectivas coordenadas homogéneas, estos están relacionados a través de  $\mathbf{H}$  por

$$\tilde{\mathbf{x}}' \propto \mathbf{H}\tilde{\mathbf{x}}, \quad (3.5)$$

donde  $\propto$  indica proporcionalidad. Conocidos los puntos, dehomogeneizar  $\mathbf{H}\tilde{\mathbf{x}}$  e igualar los valores resultantes a los de  $\mathbf{x}'$  lleva a las igualdades

$$\begin{cases} x' = \frac{h_1x + h_2y + h_3}{h_7x + h_8y + h_9} \\ y' = \frac{h_4x + h_5y + h_6}{h_7x + h_8y + h_9} \end{cases}, \quad \text{con} \quad \mathbf{H} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}, \quad (3.6)$$

que pueden escribirse linealmente en función de las entradas de  $\mathbf{H}$  como

$$\begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -x'x & -x'y & -x' \\ 0 & 0 & 0 & x & y & 1 & -y'x & -y'y & -y' \end{bmatrix} \mathbf{h} = \mathbf{0}, \quad (3.7)$$

### 3. SEGMENTACIÓN DE TRAYECTORIAS

---

siendo  $\mathbf{h} = [h_1 \ h_2 \ h_3 \ h_4 \ h_5 \ h_6 \ h_7 \ h_8 \ h_9]^T$  la vectorización por filas de  $\mathbf{H}$ . El problema (3.5) no admite la solución trivial  $\mathbf{h} = \mathbf{0}$  que sí cumple la identidad (3.7); para descartarla, puesto que todas las soluciones son proporcionales entre sí, es suficiente establecer una restricción de regularidad adicional sobre sus componentes que haga que no sean nulas. Utilizando dicho regularizador, de (3.7) se observa que son necesarios cuatro pares de correspondencias conocidas para determinar las nueve entradas de  $\mathbf{H}$ .

Es esperable que las correspondencias de puntos no sean exactas por haberse obtenido por medio del flujo óptico, y que sus fotogramas presenten ruido. Además, si el fondo no es completamente planar, cuatro pares de puntos distintos pueden resultar en diferentes homografías dependiendo de la posición en la que se encuentren. Es por ello que utilizar un número mucho mayor de ellos y ajustar  $\mathbf{H}$  para que minimice el error respecto de los valores esperados en (3.7) es una estrategia más efectiva para determinar la matriz. Sea  $\{(\mathbf{x}_i, \mathbf{x}'_i)\}_{i=1}^N$ , con  $N \geq 4$ , un conjunto de correspondencias de puntos entre dos fotogramas de un vídeo, el método Direct Linear Transform (DLT) [51] consiste en resolver el problema de minimización

$$\begin{aligned} & \min_{\mathbf{h}} \quad \|\mathbf{A}\mathbf{h}\|_2^2 \\ & \text{sujeto a} \quad \|\mathbf{h}\|_2^2 = 1, \end{aligned} \tag{3.8}$$

donde

$$\mathbf{A} = \begin{bmatrix} \tilde{\mathbf{x}}_1^T & \mathbf{0}_{1 \times 3} & -x'_1 \tilde{\mathbf{x}}_1^T \\ \mathbf{0}_{1 \times 3} & \tilde{\mathbf{x}}_1^T & -y'_1 \tilde{\mathbf{x}}_1^T \\ \vdots & \vdots & \vdots \\ \tilde{\mathbf{x}}_N^T & \mathbf{0}_{1 \times 3} & -x'_N \tilde{\mathbf{x}}_N^T \\ \mathbf{0}_{1 \times 3} & \tilde{\mathbf{x}}_N^T & -y'_N \tilde{\mathbf{x}}_N^T \end{bmatrix} \tag{3.9}$$

surge de apilar las matrices de (3.7) para cada par de observaciones.

Tanto la función objetivo como las restricciones del problema (3.8) son convexas e infinitamente diferenciables, cosa que implica la existencia de una solución única global para él, que anula el gradiente de su Lagrangiano  $\mathcal{L}$ . Así pues, definiendo las funciones  $f, g: \mathbb{R}^9 \rightarrow \mathbb{R}$  de tal manera que

$$f(\mathbf{h}) = \|\mathbf{A}\mathbf{h}\|_2^2 = \mathbf{h}^T \mathbf{A}^T \mathbf{A} \mathbf{h}, \tag{3.10}$$

$$g(\mathbf{h}) = 1 - \|\mathbf{h}\|_2^2 = 1 - \mathbf{h}^T \mathbf{h}, \tag{3.11}$$

una condición necesaria para que un vector  $\hat{\mathbf{h}} \in \mathbb{R}^9$  sea óptimo del problema (3.8) es que

$$\underbrace{2\mathbf{A}^T \mathbf{A} \hat{\mathbf{h}} + \lambda \underbrace{(-2\hat{\mathbf{h}})}_{\nabla f(\hat{\mathbf{h}})}}_{\nabla g(\hat{\mathbf{h}})} = \mathbf{0}, \tag{3.12}$$

donde  $\lambda$  es cierto multiplicador real. La igualdad (3.12) equivale a que  $\hat{\mathbf{h}}$  sea un vector propio de  $\mathbf{A}^T \mathbf{A}$  con valor propio asignado  $\lambda$  (no negativo, porque la matriz es semi-definida positiva). Multiplicando  $\hat{\mathbf{h}}^T$  por la izquierda en ambos lados de la igualdad y haciendo uso de la unitariedad de su norma, se llega a

$$f(\hat{\mathbf{h}}) = \hat{\mathbf{h}}^T \mathbf{A}^T \mathbf{A} \hat{\mathbf{h}} = \lambda \hat{\mathbf{h}}^T \hat{\mathbf{h}} = \lambda, \tag{3.13}$$

comportando que  $\hat{\mathbf{h}}$  sea minimizador de  $f$  si y solo si también minimiza  $\lambda$ . Consecuentemente, la solución al problema (3.8) es el vector propio de  $\mathbf{A}^T \mathbf{A}$  pertinente a su valor propio más pequeño, y la homografía que mejor se ajusta a las correspondencias tiene por entradas las componentes de dicho vector ordenadas por filas.

El método DLT tiene el inconveniente de ser sensible al ruido y de ver sus resultados afectados por el sistema de coordenadas utilizado para las imágenes. En efecto, dadas las transformaciones proyectivas  $\mathbf{T}, \mathbf{T}'$  tales que  $\tilde{\mathbf{y}} = \mathbf{T}\tilde{\mathbf{x}}$  e  $\tilde{\mathbf{y}}' = \mathbf{T}'\tilde{\mathbf{x}}'$ , la relación (3.5) se escribe como

$$\tilde{\mathbf{y}}' \propto \underbrace{\mathbf{T}' \mathbf{H} \mathbf{T}^{-1}}_{\bar{\mathbf{H}}} \tilde{\mathbf{y}}, \quad (3.14)$$

que induce a la minimización de  $\|\bar{\mathbf{A}}\tilde{\mathbf{h}}\|_2^2$  sujeta a  $\|\tilde{\mathbf{h}}\|_2^2 = 1$ . La matriz  $\bar{\mathbf{A}}$  cumple que  $\bar{\mathbf{A}} = \mathbf{AS}$ , donde  $\mathbf{S}$  únicamente depende de las entradas de  $\mathbf{T}$  y  $\mathbf{T}'$  [17], y  $\tilde{\mathbf{h}}$  es la vectrización de  $\bar{\mathbf{H}}$ . Sin embargo, los minimizadores  $\mathbf{h}^*$  y  $\tilde{\mathbf{h}}^*$  de  $\|\mathbf{Ah}\|_2^2$  y  $\|\mathbf{AS}\tilde{\mathbf{h}}\|_2^2$  no están relacionados por  $\tilde{\mathbf{h}}^* = \mathbf{S}^{-1}\mathbf{h}^*$ , porque las restricciones  $\|\mathbf{h}\|_2^2 = 1$  y  $\|\tilde{\mathbf{h}}\|_2^2 = 1$  son fundamentalmente diferentes: si  $\mathbf{h}^*$  es vector propio de  $\mathbf{A}^T \mathbf{A}$  con valor propio  $\lambda$ , entonces

$$\begin{aligned} \bar{\mathbf{A}}^T \bar{\mathbf{A}}(\mathbf{S}^{-1}\mathbf{h}^*) &= \mathbf{S}^T \mathbf{A}^T \mathbf{A} \mathbf{S}(\mathbf{S}^{-1}\mathbf{h}^*) \\ &= \mathbf{S}^T \mathbf{A}^T \mathbf{A} \mathbf{h}^* \\ &= \mathbf{S}^T \lambda \mathbf{h}^* \\ &= \lambda \mathbf{S}^T \mathbf{S}(\mathbf{S}^{-1}\mathbf{h}^*) \\ &\neq \lambda(\mathbf{S}^{-1}\mathbf{h}^*), \end{aligned} \quad (3.15)$$

queriendo decir que  $\mathbf{S}^{-1}\mathbf{h}^*$  no es un vector propio de  $\bar{\mathbf{A}}^T \bar{\mathbf{A}}$ , y por tanto tampoco un minimizador del nuevo problema. De esto se concluye que el sistema de coordenadas utilizado al aplicar el algoritmo DLT influye en la solución que se obtiene de él. En particular, en el sistema de píxeles habitual, donde las dos primeras componentes de las coordenadas homogéneas de un punto pueden ser órdenes de magnitud más grandes que la tercera, los efectos de perturbaciones provocadas por ruido se ven amplificados. Como remedio, en [17] se propone la *normalización isotrópica* de las coordenadas previamente al uso de DLT: trasladar los puntos para que su baricentro se encuentre en  $(0, 0)$ , y escalarlos para que su distancia euclídea media sea de  $\sqrt{2}$ , independientemente en ambas imágenes. Este preacondicionamiento de las correspondencias ajusta sus coordenadas a un sistema común, y con ello mejora la precisión de los resultados. El Algoritmo 3.1 resume el procedimiento completo, incorporando los pasos de normalización y denormalización antes y después de DLT.

Puesto que el objetivo final del cálculo de las homografías es distinguir el fondo de los objetos en vídeo, es natural que se desconozca si las correspondencias utilizadas en DLT pertenecen a un conjunto u otro. Esto supone un problema, ya que el movimiento de los puntos de un objeto no es representativo de cómo se desplaza el fondo. Si este último ocupa una proporción suficiente de las imágenes (mayor que el 50% de los píxeles), entonces los puntos de los objetos son valores atípicos (*outliers*), cuyos efectos pueden paliarse con métodos de estimación robustos. RANdom SAMple Consensus (RANSAC) [13] es uno de ellos, detallado en el Algoritmo 3.2, el cual se caracteriza por aplicar reiteradamente el Algoritmo 3.1 sobre muestras aleatorias de correspondencias y, de sus resultados, escoger el que mejor se ajusta al conjunto global.

### 3. SEGMENTACIÓN DE TRAYECTORIAS

---



---

**Algoritmo 3.1:** DLT con preacondicionamiento isotrópico

---

**Datos:** Correspondencias  $\{(\mathbf{x}_i, \mathbf{x}'_i)\}_{i=1}^N$ ,  $N \geq 4$ , entre fotogramas  
**Resultado:**  $\mathbf{H}$ , matriz de homografía de dimensión  $3 \times 3$

- 1  $(\{\mathbf{y}_i\}, \mathbf{T}) \leftarrow \text{normalizar}(\{\mathbf{x}_i\})$  ▷ Normalización
- 2  $(\{\mathbf{y}'_i\}, \mathbf{T}') \leftarrow \text{normalizar}(\{\mathbf{x}'_i\})$
- 3  $\mathbf{A} \leftarrow \text{Matriz (3.9) usando correspondencias } \{(\mathbf{y}_i, \mathbf{y}'_i)\}_{i=1}^N$  ▷ DLT
- 4  $\bar{\mathbf{h}} \leftarrow \text{Vector propio de valor propio más pequeño de } \mathbf{A}^T \mathbf{A}$
- 5  $\tilde{\mathbf{H}} \leftarrow \text{Ordenar } \bar{\mathbf{h}} \text{ en matriz } 3 \times 3$
- 6  $\mathbf{H} \leftarrow \mathbf{T}'^{-1} \tilde{\mathbf{H}} \mathbf{T}$  ▷ Denormalización
- 7 **devolver**  $\mathbf{H}$
- 8 **Función**  $\text{normalizar}(\{\mathbf{x}_i\})$ :
  - 9     $\bar{\mathbf{x}} \leftarrow \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$
  - 10     $s \leftarrow \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i - \bar{\mathbf{x}}\|_2$
  - 11     $\mathbf{T} \leftarrow \begin{bmatrix} \frac{\sqrt{2}}{s} & 0 & -\frac{\sqrt{2}}{s} \bar{x}_1 \\ 0 & \frac{\sqrt{2}}{s} & -\frac{\sqrt{2}}{s} \bar{x}_2 \\ 0 & 0 & 1 \end{bmatrix}$  ▷ Donde  $\bar{\mathbf{x}} = (\bar{x}_1, \bar{x}_2)$
  - 12    **para**  $i \in \{1, \dots, N\}$  **hacer**
    - 13      $\mathbf{y}_i \leftarrow \frac{\sqrt{2}}{s}(\mathbf{x}_i - \bar{\mathbf{x}})$  ▷ Equivalente a  $[\mathbf{y}_i^T \ 1]^T = \mathbf{T}[\mathbf{x}_i^T \ 1]^T$
  - 14    **fin**
  - 15    **devolver**  $(\{\mathbf{y}_i\}, \mathbf{T})$

---



---

**Algoritmo 3.2:** RANSAC

---

**Datos:** Correspondencias  $C = \{(\mathbf{x}_i, \mathbf{x}'_i)\}_{i=1}^N$ ,  $N \geq 4$ , entre fotogramas  
**Parámetros:** Tolerancia  $\tau$ ; iteraciones  $M$ ; tamaño de muestra  $K$   
**Resultado:**  $\mathbf{H}$ , matriz de homografía de dimensión  $3 \times 3$

- 1 **para**  $j \in \{1, \dots, M\}$  **hacer**
  - 2     $C_j \leftarrow \text{Muestra aleatoria de } K \text{ elementos } \{(\mathbf{x}_{i_k}, \mathbf{x}'_{i_k})\}_{k=1}^K \subset C$
  - 3     $\mathbf{H}_j \leftarrow \text{Aplicar Algoritmo 3.1 sobre } C_j$
  - 4     $S_j \leftarrow \{(\mathbf{x}_i, \mathbf{x}'_i) \in C : \|\mathbf{H}_j \tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}'_i\|_2 < \tau\}$  ▷  $\tilde{\mathbf{x}}_i$  son coord. homogéneas
  - 5    **fin**
  - 6     $J \leftarrow \text{argmax}_j \{|S_j| : j = 1, \dots, M\}$  ▷  $|S_j|$  indica cardinalidad
  - 7     $S \leftarrow S_j$  para cualquier  $j \in J$
  - 8     $\mathbf{H} \leftarrow \text{Aplicar Algoritmo 3.1 sobre } S$
  - 9    **devolver**  $\mathbf{H}$

---

Para cada iteración  $j$ , se define el *conjunto de consenso*  $S_j$  como el subconjunto de las correspondencias para las cuales el error de proyección de la  $j$ -ésima homografía es menor que cierto valor  $\tau$ , y su *soporte* como la cardinalidad de dicho conjunto. Después de  $M$  iteraciones, el conjunto de consenso con mayor soporte contiene mayoritariamente puntos del fondo de la imagen, y se utiliza con el Algoritmo 3.1 para obtener la homografía deseada.

RANSAC es capaz de sobrellevar proporciones elevadas de valores atípicos, ya que

### 3.1. Modelado geométrico de fondo

---

favorece homografías con un soporte alto. Sea  $\epsilon$  la probabilidad de que un par de correspondencias sean *outliers*; si se toman muestras aleatorias de tamaño  $K$  y se quiere una probabilidad  $p$  de que al menos una de las  $M$  muestras no los contenga, debe cumplirse que

$$\begin{aligned} 1 - p &= (1 - (1 - \epsilon)^K)^M \\ \iff M &= \frac{\log(1 - p)}{\log(1 - (1 - \epsilon)^K)}, \end{aligned} \quad (3.16)$$

de manera que aumentando el número de iteraciones siempre es posible situar  $p$  arbitrariamente cerca de 1. En lo que respecta a los demás parámetros, suele escogerse  $K = 4$  (la mínima cantidad de puntos necesaria para poder aplicar DLT), mientras que es habitual que  $\tau$  se elija de forma empírica según los valores atípicos previstos.

Tanto DLT como RANSAC se basan en la resolución del problema (3.8) para obtener el modelo de movimiento del fondo, difiriendo en los puntos que utilizan para llegar a su solución. Una estrategia más versátil consiste en ponderar las correspondencias utilizadas para controlar en qué medida afectan al cálculo de la homografía. Sea  $\mathbf{W} = \text{diag}(w_1, w_1, w_2, w_2, \dots, w_N, w_N)$  una matriz diagonal de dimensión  $2N \times 2N$  con todas sus entradas no negativas, esta última busca resolver

$$\begin{aligned} \min_{\mathbf{h}} \quad & \|\mathbf{W}\mathbf{A}\mathbf{h}\|_2^2 \\ \text{sujeto a} \quad & \|\mathbf{h}\|_2^2 = 1, \end{aligned} \quad (3.17)$$

que tiene por solución el vector propio de valor propio más pequeño de  $\mathbf{A}^T \mathbf{W}^T \mathbf{W} \mathbf{A}$ . Es inmediato ver que DLT equivale a (3.17) cuando  $w_i = 1$  para todo  $i \in \{1, \dots, N\}$ , mientras que la solución que da RANSAC es la misma que la que se obtiene tomando  $w_i = 1$  para las correspondencias del conjunto de consenso con mayor soporte, y  $w_i = 0$  para las demás. Para poder resolver (3.17), es necesario disponer *a priori* del vector de pesos  $\mathbf{w} = (w_1, \dots, w_N)$ ; su elección requiere de información sobre la calidad de las correspondencias, que depende de la forma de medir cómo la homografía se acomoda a ellas, manifestada como una función de coste robusta a *outliers*. Un procedimiento para llegar a una solución es alternar entre actualizar  $\mathbf{w}$  y resolver (3.17) a partir de una estimación inicial hasta llegar a un mínimo, algoritmo que se conoce como Iteratively Reweighted Least Squares (IRLS) [2].

Definidos unos residuos  $\{r_i\}_{i=1}^N$ , donde  $r_i: \mathbb{R}^9 \rightarrow [0, +\infty)$  es una función continua y derivable que cuantifica lo bien que se ajusta una homografía  $\mathbf{H}$  a la correspondencia  $(x_i, x'_i)$  (quedando implícito que las entradas de las homografías a las que se aplica pueden escalarse para que  $\|\mathbf{h}\|_2^2 = 1$  sin cambiar el valor de  $r_i(\mathbf{h})$ ), y dada una función de coste continuamente diferenciable  $\psi: [0, +\infty) \rightarrow \mathbb{R}$ , el minimizador  $\hat{\mathbf{h}}$  de la energía

$$E(\mathbf{h}, \mathbf{w}) = \sum_{i=1}^N w_i r_i(\mathbf{h})^2 \quad (3.18)$$

para la variable  $\mathbf{h} = (h_1, \dots, h_9)$  minimiza también a

$$E_\psi(\mathbf{h}) = \sum_{i=1}^N (\psi \circ r_i)(\mathbf{h}) \quad (3.19)$$

### 3. SEGMENTACIÓN DE TRAYECTORIAS

---

si y solo si el vector  $\hat{\mathbf{w}}$  cumple la igualdad

$$\frac{\partial E(\hat{\mathbf{h}}, \hat{\mathbf{w}})}{\partial h_j} = \frac{\partial E_\psi(\hat{\mathbf{h}})}{\partial h_j} = 0, \quad \forall j \in \{1, \dots, 9\}. \quad (3.20)$$

Suponiendo que ni  $r_i(\hat{\mathbf{h}})$  ni su derivada parcial respecto de  $h_j$  se anulan para ningún  $i \in \{1, \dots, N\}$  y  $j \in \{1, \dots, 9\}$ , desarrollar la ecuación (3.20) para los sumandos de las respectivas energías conduce a una fórmula cerrada para  $\hat{\mathbf{w}}$ :

$$\begin{aligned} 2\hat{w}_i r_i(\hat{\mathbf{h}}) \frac{\partial r_i(\hat{\mathbf{h}})}{\partial h_j} &= \psi'(r_i(\hat{\mathbf{h}})) \frac{\partial r_i(\hat{\mathbf{h}})}{\partial h_j} \\ \iff \hat{w}_i &= \frac{\psi'(r_i(\hat{\mathbf{h}}))}{2r_i(\hat{\mathbf{h}})}, \quad \forall i \in \{1, \dots, N\}. \end{aligned} \quad (3.21)$$

Si además el límite  $\lim_{s \rightarrow 0} \frac{\psi'(s)}{s}$  es finito, a partir de la derivada de  $\psi$  es posible encontrar un vector de pesos para el cual los problemas (3.18) y (3.19) sean equivalentes. Dada esta relación, IRLS parte de una aproximación  $\mathbf{h}^0$  de su minimizador y define las recurrencias

$$w_i^k = \frac{\psi'(r_i(\mathbf{h}^k))}{2r_i(\mathbf{h}^k)} \quad \forall i \in \{1, \dots, N\}, \quad (3.22)$$

$$\mathbf{h}^{k+1} = \arg \min_{\mathbf{h}: \|\mathbf{h}\|_2^2=1} E(\mathbf{h}, \mathbf{w}^k) \quad (3.23)$$

con la intención de que converjan a un mínimo de (3.19) cuando  $k \rightarrow +\infty$ . Utilizando la notación  $r_i^k = r_i(\mathbf{h}^k)$ , si  $\psi(\sqrt{s})$  es cóncava para  $s \geq 0$ , y suponiendo que se mantiene la desigualdad  $E(\mathbf{h}^{k+1}, \mathbf{w}^k) \leq E(\mathbf{h}^k, \mathbf{w}^k)$ , entonces

$$\begin{aligned} \psi\left(\sqrt{r_i^{k+1}}\right) - \psi\left(\sqrt{r_i^k}\right) &\leq \left(\sqrt{r_i^{k+1}} - \sqrt{r_i^k}\right) \psi'\left(\sqrt{r_i^k}\right) \\ \implies \psi\left(r_i^{k+1}\right) - \psi\left(r_i^k\right) &\leq \left(r_i^{k+1} - r_i^k\right) \frac{\psi'(r_i^k)}{2r_i^k} \\ \implies E_\psi(\mathbf{h}^{k+1}) - E_\psi(\mathbf{h}^k) &\leq E(\mathbf{h}^{k+1}, \mathbf{w}^k) - E(\mathbf{h}^k, \mathbf{w}^k) \leq 0, \end{aligned} \quad (3.24)$$

lo que quiere decir que si una iteración de IRLS reduce la energía de (3.18), también lo hace con (3.19). Para asegurar la convergencia del método a un mínimo local, además, el minimizador de (3.18) como función de  $\mathbf{w}$  debe ser continuo, y el conjunto  $\{\mathbf{h} \in \mathbb{R}^9 \mid E_\psi(\mathbf{h}) \leq E_\psi(\mathbf{h}^0)\}$  debe estar acotado [2]. En particular, si (3.19) es convexa, IRLS converge a un mínimo global.

IRLS sufre la desventaja de tener condiciones de convergencia restrictivas que no siempre se cumplen, y de exigir una aproximación inicial suficientemente buena que ha de adquirirse por medio de otro método. Aun así, en la práctica, los resultados que da en combinación con RANSAC y de funciones de coste robustas describen con precisión el movimiento del fondo de la mayoría de imágenes para las que un modelo de homografía es válido, lo que se hace evidente en su segmentación.

### 3.1.2 Extracción de trayectorias del fondo

Modelar el fondo de una secuencia de  $T$  imágenes para segmentar sus objetos requiere de un total de  $T - 1$  homografías, para la estimación de las cuales son necesarias correspondencias de puntos por cada par de fotogramas consecutivos. Mientras que es posible calcular  $T - 1$  campos de flujo óptico, obtener correspondencias a partir de ellos, y aplicar los algoritmos de la Sección 3.1.1 de manera independiente sobre ellas, se ha mostrado más efectivo el uso de trayectorias de puntos para dar coherencia temporal a los resultados. Bajo este planteamiento, Wehrwein y Szeliski [48] proponen un método basado en las trayectorias semidensas de [45], que permite ajustar las homografías de manera más precisa al movimiento del fondo utilizando información de todo el vídeo. Dicho método otorga un peso a cada trayectoria, que se usa para clasificarlas como fondo u objeto.

Sea  $\mathcal{V}: \Omega \times [1, T] \rightarrow \mathbb{R}^d$  un vídeo y  $\mathcal{C} = \{c_i: [t_i^1, t_i^2] \subseteq [1, T] \rightarrow \Omega\}_{i=1}^N$  el conjunto de las trayectorias de sus puntos obtenidas con el *tracker* de la Sección 2.3.2. El objetivo de la segmentación de trayectorias es obtener un vector  $\mathbf{w} = (w_1, \dots, w_N)$ , con  $w_i \in [0, 1]$  para  $i \in \{1, \dots, N\}$ , de modo que todos los píxeles de  $c_i$  se categoricen como objeto si  $w_i$  es menor que un umbral, y como fondo en otro caso. En un instante temporal  $t$ , el conjunto  $C_t = \{(c_i(t), c_i(t+1)) \mid t \in [t_i^1, t_i^2], 1 \leq i \leq N\}$  contiene pares de puntos correspondientes entre los fotogramas  $t$  y  $t+1$  de  $\mathcal{V}$ ; con ellos se estiman para  $t \in \{1, \dots, T-1\}$  las homografías  $\mathbf{H}_t$  que modelan el movimiento del fondo  $M = \{\mathbf{H}_t \mid 1 \leq t \leq T-1\}$ . Definir las correspondencias a partir de  $\mathcal{C}$  corrige el principal problema que aparece al determinar dicho modelo: que el peso  $w_i$  de una trayectoria  $c_i$  sea compartido entre todos los puntos por los que esta pasa ayuda a detectar valores atípicos que podrían confundirse con parte del fondo si su movimiento se analizase localmente en el tiempo.

El número de trayectorias juega un papel importante en el proceso de segmentación, ya que debe haber suficientes de ellas tanto para que el modelo de movimiento sea preciso como para proporcionar información suficiente de todo el vídeo para su posterior densificación. Es por este motivo que se favorece un *tracker* semidenso, del que se descartan trayectorias demasiado cortas (de duración menor que 5 fotogramas) cuya brevedad puede deberse al ruido. La estrategia expuesta en [48] inicializa el modelo en primer lugar aplicando RANSAC individualmente sobre  $C_t$  para todo  $t \in \{1, \dots, T-1\}$  con la finalidad de obtener una aproximación inicial de  $M$ , y a continuación la refina minimizando una función de coste robusta a *outliers* que tiene en cuenta el transcurso entero de las trayectorias para evaluar su adaptación al modelo. Los pesos derivados de la función de coste para los cuales esta se minimiza se consideran como la solución al problema de segmentación de trayectorias.

Sean  $\mathbf{x}_{i,t} = c_i(t)$  la posición de la  $i$ -ésima trayectoria en el tiempo  $t$  y  $\tilde{\mathbf{x}}_{i,t}$  sus coordenadas homogéneas. Para cada  $i \in \{1, \dots, N\}$  se define el ajuste de  $c_i$  al modelo de movimiento  $M$  como

$$r_i = \max_t \|\mathbf{H}_t \tilde{\mathbf{x}}_{i,t} - \tilde{\mathbf{x}}_{i,t+1}\|_2, \quad (3.25)$$

que es cercano a 0 solamente si todos los puntos de  $c_i$  se amoldan a las homografías del modelo. La elección del máximo de los errores de proyección de los fotogramas por los que pasa la trayectoria es efectiva para clasificar correctamente píxeles de objetos que permanecen estáticos durante ciertos períodos de tiempo. Por ejemplo, una persona sentada muestra el mismo movimiento que el fondo en los fotogramas

### 3. SEGMENTACIÓN DE TRAYECTORIAS

---

donde permanece parada, pero se distingue de él en el momento en el que se levanta; el residuo (3.25) permite que esta información se propague al resto de imágenes y se detecte a la persona incluso en aquellas donde está quieta.

Una vez estimado un modelo inicial  $M^0$  con RANSAC, los residuos de (3.25) especifican para las trayectorias unos valores de error no acotados superiormente, insuficientes por sí solos para identificar *outliers*. Para mejorar el modelo, se utiliza una función de coste cuadrática truncada  $\psi: [0, +\infty) \rightarrow \mathbb{R}$ , que penaliza residuos elevados y se satura en función de un parámetro  $\tau$ . Para cada  $i \in \{1, \dots, N\}$ , el coste de  $r_i$  se fija a

$$\psi(r_i) = \min_w \left\{ w^2 r_i^2 + \frac{\tau^2}{2} (1 - w^2)^2 \right\}, \quad (3.26)$$

donde la variable auxiliar  $w$  limita la penalización a los *outliers* actuando a la vez como un peso atribuido a  $r_i$  (y, por tanto, a la trayectoria  $c_i$ ), y como regularizador para impedir que todos los pesos valgan 0. Los pesos que minimizan esta expresión pueden determinarse explícitamente en función de los residuos derivándola respecto de  $w^2$ , siendo su fórmula

$$w(r_i)^2 = 1 - \frac{r_i^2}{\tau^2} \quad (3.27)$$

para  $r_i^2 \leq \tau^2$ , y  $w(r_i)^2 = 0$  para residuos más grandes. Así, sustituir (3.27) en (3.26) expresa el coste de cada trayectoria en función únicamente su residuo:

$$\psi(r_i) = \begin{cases} r_i^2 \left(1 - \frac{r_i^2}{2\tau^2}\right) & \text{si } r_i^2 \leq \tau^2, \\ \frac{\tau^2}{2} & \text{en otro caso.} \end{cases} \quad (3.28)$$

El sumatorio de  $\psi(r_i)$  para los residuos de cada una de las trayectorias de  $\mathcal{C}$  se considera como el coste del modelo de movimiento.

La aplicación  $\psi$  definida en (3.28) es infinitamente diferenciable, y su derivada  $\psi'$  cumple que

$$\psi'(r_i) = \begin{cases} 2r_i \left(1 - \frac{r_i^2}{\tau^2}\right) & \text{si } r_i^2 \leq \tau^2, \\ 0 & \text{en otro caso,} \end{cases} \quad (3.29)$$

de forma que  $\frac{\psi'(r_i)}{2r_i} = w(r_i)^2$  para todo  $i \in \{1, \dots, N\}$ , incluido el caso  $r_i = 0$ . Además, para  $s \geq 0$ , la función  $\phi: [0, +\infty] \rightarrow \mathbb{R}$  tal que  $\phi(s) = \psi(\sqrt{s})$  es cóncava. En efecto, cuando  $s \leq \tau^2$ ,

$$\begin{aligned} \phi(s) &= s \left(1 - \frac{s}{2\tau^2}\right) \\ \implies \phi'(s) &= 1 - \frac{s}{\tau^2} \\ \implies \phi''(s) &= -\frac{1}{\tau^2}; \end{aligned} \quad (3.30)$$

y cuando  $s > \tau^2$ ,  $\phi''$  se anula siempre. Ya que la derivada segunda de  $\phi$  es no positiva para todo  $s \geq 0$ , se satisface la condición de concavidad de segundo orden. Bajo estas hipótesis, encontrar el conjunto de homografías que minimizan el coste del modelo de movimiento equivale a resolver el problema (3.18) con los residuos (3.25) y pesos (3.27), que puede realizarse numéricamente mediante IRLS.

---

**Algoritmo 3.3:** IRLS con trayectorias y paso adaptativo

**Datos:** Modelo inicial  $M^0 = \{\mathbf{H}_t^0\}_{t=1}^T$ ; trayectorias  $\mathcal{C} = \{c_i : [t_i^1, t_i^2] \rightarrow \Omega \subset \mathbb{R}^2\}_{i=1}^N$

**Parámetros:** Umbral  $\tau$ ; tolerancia  $\epsilon$ ; máximo de iteraciones  $L$

**Resultado:** Pesos de las trayectorias  $\mathbf{w}$

```

1  $s \leftarrow 1$ 
2  $k \leftarrow 0$ 
3  $\mathbf{r}^0 \leftarrow \text{residuos}(M^0, \mathcal{C})$                                 ▷ Fórmula (3.25)
4  $\mathbf{w}^0 \leftarrow \text{pesos}(\mathbf{r}^0, \tau)$                                 ▷ Fórmula (3.27)
5  $E^0 \leftarrow +\infty$ 
6 repetir
7    $M^{k+1} \leftarrow \text{modelo}(\mathbf{w}^k, \mathcal{C})$                                 ▷ Problema (3.17)
8    $\mathbf{r}^{k+1} \leftarrow \text{residuos}(M^{k+1}, \mathcal{C})$ 
9    $\mathbf{w}^{k+1} \leftarrow (1-s)\mathbf{w}^k + s \cdot \text{pesos}(\mathbf{r}^{k+1}, \tau)$ 
10   $E^{k+1} \leftarrow \text{coste}(\mathbf{r}^{k+1}, \tau)$                                 ▷ Suma de (3.28) para todo  $r_i$ 
11  si  $E^{k+1} < E^k$  entonces
12     $k \leftarrow k + 1$ 
13     $s \leftarrow \min(4s, 1)$ 
14  en otro caso
15     $s \leftarrow s/4$ 
16  fin
17 hasta que  $|1 - E^{k+1}/E^k| < \epsilon$  ó se llega a  $L$  iteraciones
18 devolver  $\mathbf{w}^k$ 

```

---

Minimizar (3.18) es una tarea compleja porque los residuos de las trayectorias dependen de todo el modelo de movimiento de manera no lineal. En su lugar, el problema (3.17) sirve como relajación suya, lo que facilita su resolución. Definiendo la matriz  $\mathbf{A}$  a partir de los conjuntos de puntos  $C_t$  para cada imagen de la secuencia, y dando pesos a dichas correspondencias iguales al de la trayectoria a la que pertenecen, el minimizador de (3.17) para cada iteración de IRLS es una aproximación del de (3.18), que se corrige a medida que se actualizan los pesos de las trayectorias. Como la optimización de este problema relajado no garantiza una bajada del coste respecto de un modelo anterior, en [48] se propone el uso de IRLS con paso adaptativo, detallado en el Algoritmo 3.3, que se particulariza por actualizar los valores de los pesos en cada iteración haciendo una media ponderada entre los de la iteración anterior y los calculados a partir de (3.27). En caso de darse una iteración en la que no baje el coste, esta se repite sin cambiar de modelo ponderando en mayor medida los pesos anteriores; si el coste sí disminuye, el modelo se actualiza y la siguiente iteración valora más sus nuevos pesos. El algoritmo acaba cuando la diferencia de coste entre iteraciones se reduce pasado un umbral, lo que siempre sucede: en el mejor caso, el coste siempre baja e IRLS converge al minimizador del problema rápidamente; en el peor, el coste deja de bajar a partir de cierto momento, y llega una iteración  $k$  en la que  $\mathbf{w}^k = \mathbf{w}^{k-1}$  (el tamaño del paso se vuelve insignificante), igualando el coste de la iteración anterior y finalizando. Para evitar tiempos de ejecución elevados, es posible establecer un límite de iteraciones al algoritmo sin perjudicar en gran medida los resultados.

La salida del Algoritmo 3.3 es el vector  $\mathbf{w} = (w_1, \dots, w_N)$  de pesos que proviene del

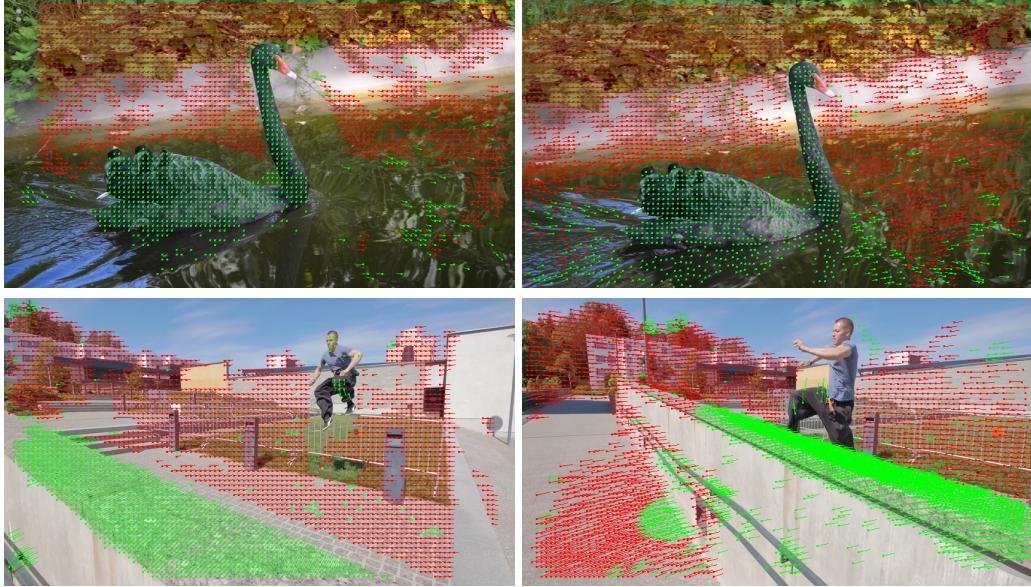
### 3. SEGMENTACIÓN DE TRAYECTORIAS

---



**Figura 3.1:** Trayectorias segmentadas por modelado de fondo de las secuencias BMX-BUMPS, BEAR, HIKE, SOAPBOX, y TRAIN (en orden de filas) de la base de datos DAVIS. En verde, objetos; en rojo, fondo. Primera columna: primer fotograma de las secuencias. Segunda columna: fotograma número 21 de las secuencias.

### 3.1. Modelado geométrico de fondo



**Figura 3.2:** Secuencias de DAVIS que no se segmentan correctamente con un modelo de fondo basado en homografías. Primera fila: fotogramas número 1 y 21 de la secuencia BLACKSWAN. Segunda fila: fotogramas número 1 y 21 de la secuencia PARKOUR.

modelo de fondo de mínimo coste, cuyas componentes toman valores entre 0 y 1. Dado un valor de corte  $a \in [0, 1]$ , los conjuntos  $\mathcal{F}_a = \{c_i \in \mathcal{C} \mid w_i \geq a\}$  y  $\mathcal{O}_a = \{c_i \in \mathcal{C} \mid w_i < a\}$  son una segmentación de las trayectorias, y corresponden respectivamente al fondo y a los objetos del vídeo. La segmentación semidensa de puntos resultante para la secuencia de imágenes consiste en todos aquellos por los que pasan las trayectorias de  $\mathcal{F}_a$  y  $\mathcal{O}_a$ . En la Figura 3.1 se visualiza dicha segmentación para varios vídeos de la base de datos DAVIS [39]. Los resultados se han obtenido con un corte  $a = 0.5$  después de aplicar IRLS con  $\tau = 4$  y RANSAC con quinientas iteraciones y tolerancia de dos píxeles.

Todos los vídeos mostrados en la Figura 3.1 tienen en común que el fondo de las escenas se mantiene estático y que la cámara sigue primariamente a los objetos, causando que el movimiento aparente del resto de puntos sea rígido y discernible. En el caso de TRAIN, además, el fondo es mayormente planar, por lo que su modelo de movimiento (y, consecuentemente, su segmentación) se ajusta a la realidad con mínimo error. La causa principal del ocasional mal etiquetado de trayectorias, manifiesto en BMX-BUMPS y SOAPBOX, tiene origen en el propio rastreo de puntos y no en el modelado de fondo: aquellas que se acercan a los bordes de objetos pueden saltar incorrectamente a una zona del vídeo que inicialmente no seguían, y ser consideradas como objeto debido a la ecuación (3.25) incluso en los fotogramas en los que sus puntos forman parte del fondo. La restricción (2.27) ayuda a disminuir la presencia de este problema, pero esta no es infalible y un vector de flujo óptico impreciso puede eventualmente satisfacerla. Este tipo de errores son de poca importancia, ya que el proceso de densificación que usualmente sigue a la segmentación de trayectorias es capaz de corregirlos.

Finalmente, las limitaciones de la segmentación de trayectorias por modelado de

### 3. SEGMENTACIÓN DE TRAYECTORIAS

---

fondo se evidencian en vídeos cuyas escenas no pueden modelarse con homografías. En la Figura 3.2 se ilustran dos ejemplos de ellas. Por un lado, BLACKSWAN es una secuencia en la que el nado de un cisne genera ondas en el agua, un movimiento no rígido que provoca que el método falle y considere el agua como objeto. Por otro, PARKOUR es una toma con paralaje exagerado en la que los puntos del fondo más cercanos a la cámara aparecen desplazarse más rápidamente que los más alejados, de modo que una única homografía por fotograma es insuficiente para modelarlo. En este último vídeo, el movimiento de la cámara también hace que el *tracker* elimine gran parte de las trayectorias que comienzan en el hombre por ser poco fiables, lo que dificulta su delimitación. A pesar de sus limitaciones, menos de un quinto de la totalidad de los vídeos de DAVIS incumple los criterios para poder ser modelado de esta manera [48], suscitando que el método sea adecuado para segmentar una gran variedad de escenas diferentes.

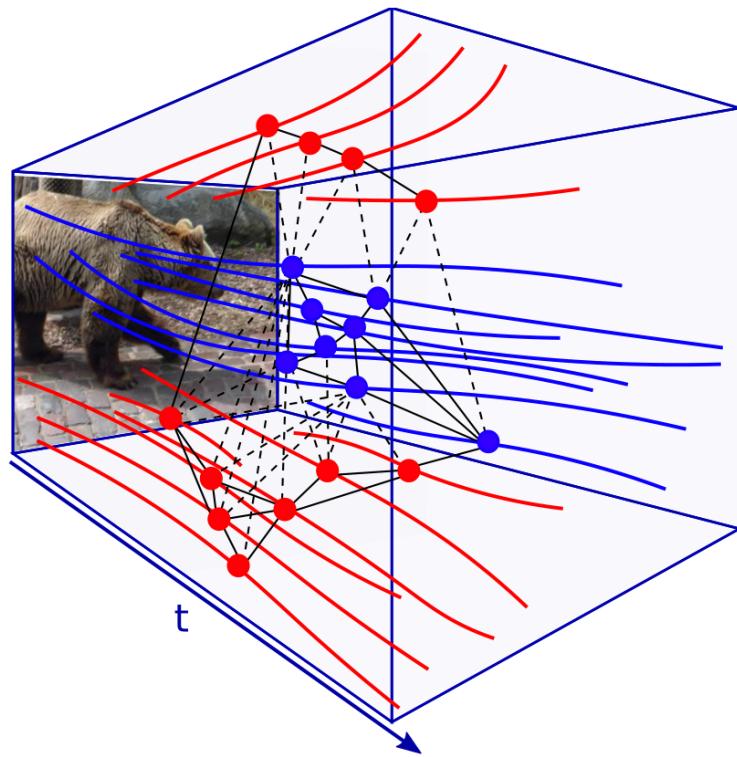
Las imágenes de las figuras 3.1 y 3.2 son los resultados de la ejecución de una implementación propia de los algoritmos explicados en la sección actual. El código fuente de esta se puede consultar en el material complementario.

## 3.2 Grafos de trayectorias

El dominio espacial de una imagen digital es un conjunto finito de elementos reales con configuración de rejilla. Cada píxel es una unidad individual de la imagen con su propio valor de color, que toma una forma rectangular y está conectado con hasta otros ocho píxeles (habitualmente cuatro) tanto por sus lados como por sus esquinas. Esta disposición induce a que se pueda representar una imagen como un grafo, siendo los píxeles sus nodos y existiendo aristas entre pares de píxeles adyacentes. Desde este punto de vista, separar los píxeles en fondo y objeto equivale a dividir los nodos del grafo en conjuntos disjuntos, que se consigue cortando las aristas que los separan. Diversos autores [5, 15, 21, 35] utilizan estrategias de corte en grafos con pesos para segmentar imágenes individuales; el criterio del corte depende de los pesos de las aristas a cortar, y puede aceptar restricciones de pertenencia de ciertos nodos para adaptarse a información conocida.

Definir grafos individuales para cada fotograma de un vídeo desatiende la existencia de coherencia temporal y de movimiento a largo plazo. En su lugar, [24, 37, 38] optan por conectar trayectorias con fotogramas en común, construyendo un único grafo que abarca todo el dominio temporal (ejemplificado en la Figura 3.3 [24, Fig. 2]). Partitionar este grafo lleva a una segmentación de las trayectorias (y, por tanto, de todos los puntos por los que pasan), que tiene la ventaja de determinarse globalmente. Puesto que estas no siguen todos los puntos que hay en un vídeo, la segmentación resultante no es densa; aun así, si el conjunto de ellas es suficientemente grande (como el obtenido por un *tracker* semidenso), un proceso de densificación (visto en el Capítulo 5) es aplicable para cubrir todo el dominio espacial.

Delinear manualmente zonas de algunos fotogramas de un vídeo permite clasificar trayectorias enteras como parte de un objeto o del fondo, aunque no se haga con todos los puntos por los que pasan. En el grafo de trayectorias esto se traduce en etiquetar un nodo antes de cortar, lo que influye en cómo agrupar el resto de vértices. Una segmentación de vídeo interactiva hace uso de *semillas* dadas por un usuario en



**Figura 3.3:** Grafo de trayectorias de una secuencia de imágenes. Cada trayectoria se identifica como un nodo, y forma una arista con todas aquellas con las que comparte fotogramas, con un peso que depende de su similitud. El corte del grafo (aristas discontinuas) separa los nodos en conjuntos disjuntos, que induce a una segmentación de las trayectorias.

determinados fotogramas, que se difunden a todas las trayectorias a través del grafo; realizar un corte sobre él manteniendo el etiquetado inicial, pues, es un problema ligado con la forma óptima de esparcir las semillas.

En esta sección se estudia cómo relacionar las trayectorias en un grafo para que un corte las agrupe correctamente acorde a la visión humana. También se presenta el criterio de corte mínimo normalizado como principio para separar nodos del grafo, así como su vínculo con paseos aleatorios sobre él. Finalmente, se expone un método de segmentación interactiva de trayectorias basado en la propagación de semillas por medio de paseos aleatorios.

### 3.2.1 Afinidad de trayectorias

Partitionar un grafo cortando sus aristas requiere otorgar a cada una de ellas un valor numérico que favorezca el corte en mayor o menor medida. Idealmente, las aristas entre nodos de diferentes agrupaciones deben ser fáciles de cortar, mientras que ha de penalizarse el corte de aquellas que unen nodos estrechamente relacionados entre sí. En un grafo donde los nodos son trayectorias de puntos, cuantificar la similitud que hay entre ellas cumple esta función. Trayectorias afines favorecen clasificarse igual;

### 3. SEGMENTACIÓN DE TRAYECTORIAS

---

trayectorias dispares, de manera separada.

El conjunto de puntos que son rastreados por un *tracker* en una secuencia larga de imágenes forma trayectorias asíncronas, que cubren una variedad de intervalos temporales distintos. Las trayectorias que sobreviven la duración entera de un vídeo son escasas o incluso inexistentes, y pares de ellas pueden no coincidir en el tiempo. La evaluación de la similitud de trayectorias solamente tiene sentido si estas coexisten en un fotograma o más, de modo que el grafo que forman no puede contener aristas entre aquellas que no se solapan en el tiempo; esto equivale a que tengan una afinidad nula. Si el conjunto de trayectorias es suficientemente grande, sin embargo, cada trayectoria tiene una arista con un número elevado de otras de ellas, facilitando la existencia de caminos entre aquellas inicialmente no conectadas. Establecer pesos a las aristas del grafo, por tanto, sirve para relacionar globalmente todas las trayectorias entre sí.

Generalmente, los puntos que se mueven conjuntamente en una escena suelen poder agruparse visualmente como parte de una misma entidad: por ejemplo, los puntos de una rueda muestran un distintivo movimiento rotatorio que los identifica como tales. No obstante, existen abundantes situaciones en las que este principio es ineficaz para segmentar objetos. Dos personas caminando una al lado de la otra comparten el mismo movimiento a pesar de ser objetos distintos, así como un depredador acechando inmóvil a su presa se funde con el fondo. Es en casos como estos que es importante la faceta a largo plazo de las trayectorias de puntos, no siendo necesario comparar sus recorridos en fotogramas individuales para clasificarlas. En su lugar, para cada par de trayectorias, identificar el instante de tiempo en el que su movimiento es maximalmente diferente proporciona la mayor evidencia de que sus puntos pertenecen o no a la misma entidad: las dos personas se disciernen en el momento en el que se separan, y el depredador se percibe como tal cuando comienza a dar caza a la presa.

Sean  $\mathcal{V}: \Omega \times [1, T] \rightarrow \mathbb{R}^d$  un vídeo y  $c_i: [t_i^1, t_i^2] \rightarrow \Omega$  con  $i \in \{1, 2\}$  dos trayectorias tales que  $[t_1^1, t_1^2] \cap [t_2^1, t_2^2] \neq \emptyset$ , de modo que comparten como mínimo un fotograma en común. Para cuantificar su disimilitud, con tal de compararlas por su momento de mayor diferencia, [38] propone la aplicación  $d: \mathcal{C} \times \mathcal{C} \rightarrow [0, +\infty)$  tal que

$$d(c_1, c_2) = \max_t d_t(c_1, c_2), \quad (3.31)$$

donde  $\mathcal{C}$  es el conjunto de todas las trayectorias y  $d_t: \mathcal{C} \times \mathcal{C} \rightarrow [0, +\infty)$  es una pseudodistancia que actúa en ellas en el tiempo  $t$ . Como los pesos de las aristas del grafo de trayectorias deben ser una medida de la afinidad entre ellas, la desemejanza (3.31) se transforma en un valor de semejanza como

$$w(c_1, c_2) = \exp(-\lambda d(c_1, c_2)^2), \quad (3.32)$$

donde  $\exp$  es la función exponencial y  $\lambda > 0$  es un parámetro de escala. De esta manera, trayectorias menos distantes tienen una mayor afinidad entre sí.

La pseudodistancia  $d_t$  en (3.31) ha de establecerse en función de las características que diferencian los puntos de las trayectorias en un fotograma determinado. Una de ellas es el desplazamiento inmediato de los puntos, que señala el sentido y la magnitud de su movimiento, y viene determinado por la derivada en el tiempo  $t$  de las curvas que definen las trayectorias. Dada la desviación estándar  $\sigma_t$  del flujo óptico en el fotograma  $t$ , [38] especifica la diferencia de movimiento entre ellas como

$$d_t^{\text{mo}}(c_1, c_2) = \frac{\|\partial c_1(t) - \partial c_2(t)\|_2}{\sigma_t}, \quad (3.33)$$

donde

$$\partial c_i(t) = \frac{1}{F} (c_i(t+F) - c_i(t)) \quad \forall i \in \{1, 2\} \quad (3.34)$$

es la aproximación de la derivada de  $c_i$  en  $t$  por diferencias avanzadas. La necesidad de utilizar (3.34) como media del desplazamiento a lo largo de  $F$  fotogramas surge de que el dominio temporal de  $\mathcal{V}$  es discreto. Una elección de  $F$  demasiado pequeña puede causar que haya errores debidos al ruido en la aproximación de la derivada; si esta es demasiado grande, en cambio, pueden perderse matrices relevantes en el movimiento de las curvas. A 30 FPS,  $F = 5$  (o el número *frames* en común del par de trayectorias, si es menor) supone un intervalo temporal de menos de 170 ms, lo que es aceptable para no descuidar cambios repentinos en el movimiento de objetos comunes. Por otro lado, el uso de  $\sigma_t$  en (3.33) normaliza la distancia según la cantidad de movimiento en la imagen: de esta forma, un desplazamiento de unos pocos píxeles puede ser significativo si la escena está estática, y uno largo puede no serlo si la cámara se mueve bruscamente.

Aunque el uso de las desemejanzas (3.33) entre pares de trayectorias asume que el movimiento de los objetos es translacional, lo que puede no ser cierto, localmente esta aproximación es usualmente válida a altas frecuencias de captura. Aun así, el efecto del ruido puede ser notorio en escenas con desplazamientos complejos y rápidos; [38] sugiere limitar su impacto modificando (3.33) para que dependa de la distancia espacial media  $d^{\text{sp}}$  de los píxeles en los fotogramas en común de las trayectorias, de modo que sean afines únicamente aquellas con el mismo movimiento y cercanas espacialmente. Con esto, las pseudodistancias en el tiempo  $t$  de (3.31) se definen como

$$d_t(c_1, c_2)^2 = d^{\text{sp}}(c_1, c_2) d_t^{\text{mo}}(c_1, c_2)^2. \quad (3.35)$$

Por otra parte, en [24], en lugar de tratar de sofocar el ruido multiplicando la distancia espacial, se opta por un modelo aditivo lineal. Tomando  $d^c$  como la diferencia media de color en  $L^*a^*b^*$  entre trayectorias, la pseudodistancia

$$d_t(c_1, c_2) = \beta_0 + \beta_1 d^{\text{sp}}(c_1, c_2) + \beta_2 d^c(c_1, c_2) + \beta_3 d_t^{\text{mo}}(c_1, c_2) \quad (3.36)$$

es una segunda opción para cuantificar las diferencias entre ellas, donde  $\beta_i$  son coeficientes reales para todo  $i \in \{0, \dots, 3\}$ . Además, [24] establece un umbral para limitar el peso de las distancias espacial y de color en (3.36), contemplando la posibilidad de que las trayectorias puedan ser partes distantes del mismo objeto diferentes en apariencia. Consecuentemente, tanto (3.35) como (3.36) sustituidas en (3.31) resultan en afinidades con la fórmula (3.32), que relacionan entre sí la totalidad de las trayectorias.

### 3.2.2 Cortes mínimos normalizados

Partitionar un grafo  $G = (V, E)$  puede realizarse simplemente eliminando las aristas que unen las diferentes partes. Elegir qué aristas suprimir requiere de un criterio subjetivo que depende de cómo han de ser las divisiones del grafo para ajustarse a la solución real del problema que este modela. Suponiendo pesos entre nodos, el grado de similitud de dos conjuntos  $A, B \subset V$  tales que  $A \cup B = V$  y  $A \cap B = \emptyset$  puede medirse como el *corte* [21]

$$\text{cut}(A, B) = \sum_{\substack{i \in A \\ j \in B}} w(i, j), \quad (3.37)$$

### 3. SEGMENTACIÓN DE TRAYECTORIAS

---

cuya minimización lleva a una bipartición óptima en el sentido de que los conjuntos se separan por su máxima disimilitud. Agrupar de esta manera los nodos, sin embargo, favorece el corte de pequeños subconjuntos aislados del grafo, puesto que (3.37) crece con el número de aristas cortadas. Por tanto, el corte mínimo no solo trata de separar conjuntos dispares, sino que lo hace evitando cortar demasiadas aristas. En un grafo de trayectorias, dicho criterio puede conducir a una segmentación errónea si los objetos quedan cubiertos por suficientes de ellas, lo que es inaceptable.

Shi y Malik [21] proponen en su lugar una medida de asociación entre subconjuntos que toma en consideración la cercanía de los nodos dentro de una misma agrupación. Normalizando el valor del corte por la suma de los pesos en cada partición, se previene el sesgo que beneficia la supresión de pocas aristas, y se da opción a que los tamaños de los conjuntos resultantes sean más equilibrados. El corte normalizado sobre un grafo se define como

$$\text{Ncut}(A, B) = \left( \frac{1}{\text{vol}_V A} + \frac{1}{\text{vol}_V B} \right) \text{cut}(A, B), \quad (3.38)$$

siendo  $\text{vol}_V X$  el volumen de  $X$  para todo subconjunto de nodos  $X \subseteq V$ , con expresión

$$\text{vol}_V X = \sum_{\substack{i \in X \\ j \in V}} w(i, j). \quad (3.39)$$

El corte resultante de minimizar (3.38) difícilmente contiene pocos nodos aislados, ya que volúmenes reducidos son penalizados; en cambio, se fomenta que sea proporcionalmente pequeño al tamaño de las particiones. La minimización del corte normalizado también tiene la propiedad de maximizar la asociación relativa

$$\text{Nassoc}(A, B) = \frac{\text{vol}_V A}{\text{vol}_V A} + \frac{\text{vol}_V B}{\text{vol}_V B}, \quad (3.40)$$

que refleja en qué medida son afines entre sí los interiores de las agrupaciones. En efecto, notando que  $\text{vol}_V A$  se descompone en los pesos de las aristas que se mantienen en  $A$  y en los que salen a  $B$  (correspondientes al corte) e igualmente para  $\text{vol}_V B$  (puesto que el corte es simétrico), se obtiene

$$\begin{aligned} \text{Ncut}(A, B) &= \frac{\text{cut}(A, B)}{\text{vol}_V A} + \frac{\text{cut}(A, B)}{\text{vol}_V B} \\ &= \frac{\text{vol}_V A - \text{vol}_A A}{\text{vol}_V A} + \frac{\text{vol}_V B - \text{vol}_B B}{\text{vol}_V B} \\ &= 2 - \left( \frac{\text{vol}_A A}{\text{vol}_V A} + \frac{\text{vol}_B B}{\text{vol}_V B} \right) \\ &= 2 - \text{Nassoc}(A, B), \end{aligned} \quad (3.41)$$

de donde se deduce que una disminución de  $\text{Ncut}(A, B)$  comporta un aumento de  $\text{Nassoc}(A, B)$ . Consecuentemente, el criterio del corte normalizado simultáneamente minimiza la similitud entre los conjuntos y maximiza la asociación en su interior.

Conociendo la estructura del grafo es posible obtener la partición óptima de manera algebraica. Dados los conjuntos  $A \subset V$  y  $B = V \setminus A$ , sea  $\mathbf{x}$  un vector de dimensión  $N = |V|$  tal que su  $i$ -ésima componente toma valor  $x_i = 1$  si el nodo  $i$  se encuentra en  $A$ , y  $x_i = -1$  si se halla en  $B$ . Utilizando la notación  $w_{ij} = w(i, j)$  para los pesos

### 3.2. Grafos de trayectorias

de las aristas, y  $d_i = \sum_{j \in V} w_{ij}$  para los grados de los nodos, la expresión (3.38) puede reescribirse como

$$\begin{aligned} \text{Ncut}(\mathbf{x}) &= \frac{\sum_{x_i>0, x_j<0} w_{ij}}{\sum_{x_i>0} d_i} + \frac{\sum_{x_i<0, x_j>0} w_{ij}}{\sum_{x_i<0} d_i} \\ &= \frac{\sum_{x_i>0} d_i - \sum_{x_i>0, x_j>0} w_{ij}}{\sum_{x_i>0} d_i} + \frac{\sum_{x_i<0} d_i - \sum_{x_i<0, x_j<0} w_{ij}}{\sum_{x_i<0} d_i}. \end{aligned} \quad (3.42)$$

Sea  $\mathbf{D} = \text{diag}(d_1, \dots, d_N)$ , y  $\mathbf{W}$  una matriz simétrica de dimensión  $N \times N$  con los pesos  $w_{ij}$  por entradas y ceros en la diagonal. Tomando

$$k = \frac{\sum_{x_i>0} d_i}{\sum_{i \in V} d_i} \neq 1 \quad (3.43)$$

y teniendo en cuenta que los vectores  $\frac{1}{2}(\mathbf{1} + \mathbf{x})$  y  $\frac{1}{2}(\mathbf{1} - \mathbf{x})$  son indicadores de las condiciones  $x_i > 0$  y  $x_i < 0$  respectivamente, los sumatorios de (3.42) pueden reformularse como productos de matrices:

$$\begin{aligned} 4 \text{Ncut}(A, B) &= \frac{(\mathbf{1} + \mathbf{x})^T (\mathbf{D} - \mathbf{W}) (\mathbf{1} + \mathbf{x})}{k \mathbf{1}^T \mathbf{D} \mathbf{1}} + \frac{(\mathbf{1} - \mathbf{x})^T (\mathbf{D} - \mathbf{W}) (\mathbf{1} - \mathbf{x})}{(1-k) \mathbf{1}^T \mathbf{D} \mathbf{1}} \\ &= \frac{\mathbf{x}^T (\mathbf{D} - \mathbf{W}) \mathbf{x} + \mathbf{1}^T (\mathbf{D} - \mathbf{W}) \mathbf{1}}{k(1-k) \mathbf{1}^T \mathbf{D} \mathbf{1}} + \frac{2(1-2k) \mathbf{1}^T (\mathbf{D} - \mathbf{W}) \mathbf{x}}{k(1-k) \mathbf{1}^T \mathbf{D} \mathbf{1}}. \end{aligned}$$

Denotando

$$\begin{aligned} \alpha(\mathbf{x}) &= \mathbf{x}^T (\mathbf{D} - \mathbf{W}) \mathbf{x}, & \beta(\mathbf{x}) &= \mathbf{1}^T (\mathbf{D} - \mathbf{W}) \mathbf{x}, \\ \gamma &= \mathbf{1}^T (\mathbf{D} - \mathbf{W}) \mathbf{1} = 0, & M &= \mathbf{1}^T \mathbf{D} \mathbf{1}, \end{aligned}$$

la expresión del corte normalizado se expande como

$$\begin{aligned} C = 4 \text{Ncut}(A, B) &= \frac{(\alpha(\mathbf{x}) + \gamma) + 2(1-2k)\beta(\mathbf{x})}{k(1-k)M} \\ &= \frac{(\alpha(\mathbf{x}) + \gamma) + 2(1-2k)\beta(\mathbf{x})}{k(1-k)M} - \frac{2(\alpha(\mathbf{x}) + \gamma)}{M} + \frac{2\alpha(\mathbf{x})}{M} + \frac{2\gamma}{M} \\ &= \frac{[k^2 + (1-k)^2](\alpha(\mathbf{x}) + \gamma) + 2(1-2k)\beta(\mathbf{x})}{k(1-k)M} + \frac{2\alpha(\mathbf{x})}{M}, \end{aligned}$$

y dividiendo numerador y denominador entre  $(1-k)^2$ , fijando  $b = \frac{k}{1-k}$ , y recordando que  $\gamma = 0$ , se convierte en

$$\begin{aligned} C &= \frac{(1+b^2)(\alpha(\mathbf{x}) + \gamma) + 2(1-b^2)\beta(\mathbf{x})}{bM} + \frac{2b\alpha(\mathbf{x})}{bM} - \frac{2b\gamma}{bM} \\ &= \frac{(\mathbf{1} + \mathbf{x})^T (\mathbf{D} - \mathbf{W}) (\mathbf{1} + \mathbf{x})}{b \mathbf{1}^T \mathbf{D} \mathbf{1}} + \frac{b^2 (\mathbf{1} - \mathbf{x})^T (\mathbf{D} - \mathbf{W}) (\mathbf{1} - \mathbf{x})}{b \mathbf{1}^T \mathbf{D} \mathbf{1}} - \frac{2b (\mathbf{1} - \mathbf{x})^T (\mathbf{D} - \mathbf{W}) (\mathbf{1} + \mathbf{x})}{b \mathbf{1}^T \mathbf{D} \mathbf{1}} \\ &= \frac{[(\mathbf{1} + \mathbf{x}) - b(\mathbf{1} - \mathbf{x})]^T (\mathbf{D} - \mathbf{W}) [(\mathbf{1} + \mathbf{x}) - b(\mathbf{1} - \mathbf{x})]^T}{b \mathbf{1}^T \mathbf{D} \mathbf{1}}. \end{aligned} \quad (3.44)$$

En este punto, ya que  $b = \frac{k}{1-k} = \frac{\sum_{x_i>0} d_i}{\sum_{x_i<0} d_i}$ , el cambio de variable  $\mathbf{y} = \frac{1}{2}[(\mathbf{1} + \mathbf{x}) - b(\mathbf{1} - \mathbf{x})]$  lleva a las igualdades

$$\mathbf{y}^T \mathbf{D} \mathbf{1} = \sum_{i \in V} d_i \left[ \frac{1}{2}(1+x_i) - b \frac{1}{2}(1-x_i) \right] = \sum_{x_i>0} d_i - b \sum_{x_i<0} d_i = 0 \quad (3.45)$$

### 3. SEGMENTACIÓN DE TRAYECTORIAS

---

y

$$b\mathbf{1}^T \mathbf{D}\mathbf{1} = b \sum_{i \in V} d_i = b \left( \sum_{x_i > 0} d_i + \sum_{x_i < 0} d_i \right) = \sum_{x_i > 0} d_i + b^2 \sum_{x_i < 0} d_i = \mathbf{y}^T \mathbf{D}\mathbf{y}. \quad (3.46)$$

Finalmente, juntar (3.44), (3.45), y (3.46) comporta que la minimización del corte normalizado cumple que

$$\min_{\mathbf{x}} \text{Ncut}(\mathbf{x}) = \min_{\mathbf{y}} \frac{\mathbf{y}^T (\mathbf{D} - \mathbf{W}) \mathbf{y}}{\mathbf{y}^T \mathbf{D} \mathbf{y}} \quad (3.47)$$

bajo las restricciones  $\mathbf{y}^T \mathbf{D}\mathbf{1} = 0$  e  $y_i \in \{1, -b\}$  para cada componente de  $\mathbf{y}$ .

La expresión de la derecha de (3.47) es invariante a cambios de escala. Relajando el problema para que pueda tomar valores reales, por tanto, basta estudiar el caso en el que el denominador es la unidad. En esas circunstancias, su optimización tiene las mismas características que el problema (3.8): como  $\mathbf{D}$  es diagonal, tomar  $\mathbf{z} = \mathbf{D}^{\frac{1}{2}} \mathbf{y}$  permite reescribirlo como

$$\begin{aligned} \min_{\mathbf{z}} \quad & \mathbf{z}^T \mathbf{D}^{-\frac{1}{2}} (\mathbf{D} - \mathbf{W}) \mathbf{D}^{-\frac{1}{2}} \mathbf{z} \\ \text{sujeto a} \quad & \mathbf{z}^T \mathbf{z} = 1, \\ & \mathbf{z}^T \mathbf{D}^{\frac{1}{2}} \mathbf{1} = 0, \end{aligned} \quad (3.48)$$

donde la última restricción es la equivalente a  $\mathbf{y}^T \mathbf{D}\mathbf{1} = 0$ . De (3.13), además, se sabe que el problema sin la última restricción (el cociente (3.47)) es minimizado por el vector propio de valor propio más pequeño de  $\mathbf{D}^{-\frac{1}{2}} (\mathbf{D} - \mathbf{W}) \mathbf{D}^{-\frac{1}{2}}$ , con mínimo igual a su correspondiente valor propio. Dicho vector propio es  $\mathbf{z}_0 = \mathbf{D}^{\frac{1}{2}} \mathbf{1}$ , con valor propio  $\lambda_0 = 0$ , cuya obtención es inmediata al ver que filas y columnas de  $\mathbf{D} - \mathbf{W}$  suman cero y que la matriz es simétrica semidefinida positiva. Por esta última propiedad, los vectores propios de  $\mathbf{D}^{-\frac{1}{2}} (\mathbf{D} - \mathbf{W}) \mathbf{D}^{-\frac{1}{2}}$  son perpendiculares entre sí; en particular, aunque  $\mathbf{z}_0$  no es óptimo en (3.48) porque  $\mathbf{z}_0^T \mathbf{D}^{\frac{1}{2}} \mathbf{1} = \mathbf{1}^T \mathbf{D}\mathbf{1} \neq 0$ , el siguiente vector propio  $\mathbf{z}_1$  (en orden creciente de valores propios) sí cumple que  $\mathbf{z}_1^T \mathbf{D}^{\frac{1}{2}} \mathbf{1} = \mathbf{z}_1^T \mathbf{z}_0 = 0$ . Este último, de hecho, es el minimizador de (3.48) [21], y por consiguiente

$$\begin{aligned} \mathbf{z}_1 &= \underset{\mathbf{z}^T \mathbf{z}_0 = 0}{\operatorname{argmin}} \frac{\mathbf{z}^T \mathbf{D}^{-\frac{1}{2}} (\mathbf{D} - \mathbf{W}) \mathbf{D}^{-\frac{1}{2}} \mathbf{z}}{\mathbf{z}^T \mathbf{z}} \\ \implies \mathbf{y}_1 &= \underset{\mathbf{y}^T \mathbf{D}\mathbf{1}=0}{\operatorname{argmin}} \frac{\mathbf{y}^T (\mathbf{D} - \mathbf{W}) \mathbf{y}}{\mathbf{y}^T \mathbf{D} \mathbf{y}}, \end{aligned} \quad (3.49)$$

donde  $\mathbf{y}_1$  es el vector propio con el segundo valor propio más pequeño del problema de valores propios generalizado

$$(\mathbf{D} - \mathbf{W}) \mathbf{y} = \lambda \mathbf{D} \mathbf{y}. \quad (3.50)$$

La solución  $\mathbf{y}_1$  de (3.49) no es necesariamente exacta a la de (3.47), ya que las componentes del vector pueden tomar cualquier valor real en lugar de únicamente dos de ellos. Sin embargo, estas tienden a formar dos grupos distintivos, que inducen a una partición del grafo. Aplicar un algoritmo de *clustering* sobre el vector  $\mathbf{y}_1$ , de dimensión  $N$ , es una manera de separar los nodos llamada *agrupamiento espectral*: dados dos *clusters*  $A$  y  $B$  con centros  $\mu_A$  y  $\mu_B$ , la relación  $i \in A \iff |y_i - \mu_A| < |y_i - \mu_B|$

clasifica cada nodo  $i \in V$  en uno de los dos conjuntos disjuntos. Por otro lado, si se conoce previamente la partición a la que pertenecen algunos nodos de  $V$ , agrupar el resto de ellos según la distancia mínima a los ya etiquetados también lleva a una clasificación completa de ellos.

El proceso de minimización del corte normalizado puede extenderse para subdividir el grafo en más de dos conjuntos. En el grafo de trayectorias de puntos de un vídeo, esto significa que la segmentación no se ve limitada únicamente a fondo y objetos, sino que distintos objetos pueden clasificarse independientemente. Un argumento similar al de la optimización de (3.48) conduce a que el tercer vector propio en orden creciente de valores propios subpartitiona óptimamente las primeras dos partes, así como los subsiguientes vectores propios también lo hacen con los que les preceden. Sin embargo, en la práctica, resulta más preciso volver a resolver el problema (3.48) sobre los subgrafos resultantes de la primera bipartición [21], o formular un problema de minimización de cortes múltiples en su lugar [24].

### 3.2.3 Paseos aleatorios

En la Sección 3.2.2 se ha visto que el criterio de la minimización del corte normalizado sirve para biparticionar un grafo en conjuntos de alto volumen  $A$  y  $B$ , mínimamente similares entre sí. A partir de la resolución de un problema de valores propios generalizado (3.50), se espera que las componentes del segundo vector propio  $y_1$  sean aproximadamente constantes a trozos, de modo que

$$y_{1i} \approx \begin{cases} \mu_A & \text{si } i \in A \\ \mu_B & \text{si } i \in B \end{cases} \quad (3.51)$$

y cada nodo  $i \in V$  se clasifique en función de su entrada en  $y_1$ . Meilă y Shi [35] muestran que este método tiene una interpretación probabilística, que da una explicación intuitiva de por qué secciona el grafo eficazmente.

Normalizar la matriz de pesos  $W$  del grafo por los grados de cada nodo lleva a la matriz estocástica de una cadena de Markov

$$\mathbf{P} = \mathbf{D}^{-1} \mathbf{W}, \quad (3.52)$$

cuyas filas y columnas suman la unidad.  $\mathbf{P}$  es la matriz de probabilidades de un paseo aleatorio sobre los nodos del grafo: la entrada  $P_{ij}$  representa la probabilidad de saltar al nodo  $j$  en un único movimiento si la ubicación anterior es el nodo  $i$ , y es igual a la proporción que supone la afinidad  $w_{ij}$  respecto de la suma de los pesos de las aristas salientes del nodo  $i$ . El paseo aleatorio que viene dado por  $\mathbf{P}$  está estrechamente relacionado con la minimización del corte normalizado; específicamente, los vectores propios de  $\mathbf{P}$  son exactamente los mismos que los de (3.50), y sus valores propios se obtienen de sustraer a 1 los de este último:

$$\begin{aligned} (\mathbf{D} - \mathbf{W})\mathbf{y} &= \lambda \mathbf{D}\mathbf{y} \\ \iff \mathbf{y} - \mathbf{D}^{-1} \mathbf{W}\mathbf{y} &= \lambda \mathbf{y} \\ \iff \mathbf{P}\mathbf{y} - \mathbf{D}^{-1} \mathbf{W}\mathbf{y} &= (1 - \lambda)\mathbf{y}. \end{aligned} \quad (3.53)$$

Esta característica establece una equivalencia entre el comportamiento del paseo sobre el grafo y la forma de particionar sus vértices. Es más, desde el punto de vista

### 3. SEGMENTACIÓN DE TRAYECTORIAS

---

probabilístico, el primer vector propio de (3.50),  $\mathbf{y}_0 = \mathbf{1}$ , se corresponde con un estado de igual probabilidad para todos los nodos, lo que explica que no aporte información sobre cómo clasificarlos. Por otro lado, el segundo valor propio revela cuán estrecha es la conectividad del grafo, marcando el ritmo al que el paseo se estabiliza.

Si se comienza un paseo aleatorio en el nodo  $i$  con probabilidad  $q_i^0$ , su primer salto se corresponde con el producto del vector fila  $\mathbf{q}^0$ , que contiene la distribución inicial de probabilidades en sus componentes, por la matriz  $\mathbf{P}$ . Más generalmente, la distribución de probabilidades  $\mathbf{q}^k$  en el salto  $k$  depende de las distribuciones anteriores como

$$\mathbf{q}^k = \mathbf{q}^{k-1} \mathbf{P} = \mathbf{q}^0 \mathbf{P}^k. \quad (3.54)$$

Si una cadena de Markov es irreducible (todos los nodos son accesibles) y aperiodica (el paseo puede volver a un nodo en cualquier momento), entonces tiene también una única distribución estacionaria  $\boldsymbol{\pi}$ , que no cambia en subsiguientes saltos; esta es la distribución a la que limita  $\mathbf{q}^k$  cuando  $k \rightarrow \infty$ , y estudiarla expone el comportamiento global del paseo. El grafo de trayectorias construido en la Sección 3.2.1 es conexo siempre y cuando no se pierda la totalidad de trayectorias entre dos fotogramas, y por tanto siempre existe un camino de ida y vuelta entre dos de ellas, haciendo que la cadena sea irreducible. Además, dicho grafo no es bipartito si coinciden al menos tres trayectorias en un mismo fotograma, ya que entre ellas forman un ciclo de longitud impar. Un camino que empieza en el nodo  $i$  puede volver a él en dos pasos a través de la misma arista, así como en un número de saltos impar a través de uno de los ciclos que hacen que el grafo no sea bipartito; consecuentemente, el paseo es también aperiodico y existe  $\boldsymbol{\pi}$  tal que  $\boldsymbol{\pi} \mathbf{P} = \boldsymbol{\pi}$ .

Sea  $\boldsymbol{\pi}$  el vector cuya  $i$ -ésima componente es el cociente entre el grado del nodo  $i$  y la suma de los grados de todos los vértices del grafo:

$$\pi_i = \frac{d_i}{\text{vol}_V V}. \quad (3.55)$$

Este es precisamente el vector que contiene la distribución estacionaria de la cadena de Markov. Como  $\mathbf{P}$  es simétrica, la suma de los pesos de las aristas que entran en un nodo es igual a la suma de las que salen, lo que lleva a la equivalencia

$$\pi_i P_{ij} = \pi_j P_{ji} = \frac{w_{ij}}{\text{vol}_V V}, \quad (3.56)$$

por la cual la cadena es reversible. Por ese mismo motivo, un paso desde ese estado cumple

$$\begin{aligned} \boldsymbol{\pi} \mathbf{P} &= \left[ \frac{d_1}{\text{vol}_V V} \quad \frac{d_2}{\text{vol}_V V} \quad \dots \quad \frac{d_N}{\text{vol}_V V} \right] \begin{bmatrix} \frac{w_{11}}{d_1} & \frac{w_{12}}{d_1} & \dots & \frac{w_{1N}}{d_1} \\ \frac{w_{21}}{d_2} & \frac{w_{22}}{d_2} & \dots & \frac{w_{2N}}{d_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{w_{N1}}{d_N} & \frac{w_{N2}}{d_N} & \dots & \frac{w_{NN}}{d_N} \end{bmatrix} \\ &= \left[ \frac{\sum_{i \in V} w_{j1}}{\text{vol}_V V} \quad \frac{\sum_{i \in V} w_{j2}}{\text{vol}_V V} \quad \dots \quad \frac{\sum_{i \in V} w_{jN}}{\text{vol}_V V} \right] \\ &= \left[ \frac{d_1}{\text{vol}_V V} \quad \frac{d_2}{\text{vol}_V V} \quad \dots \quad \frac{d_N}{\text{vol}_V V} \right] = \boldsymbol{\pi}, \end{aligned} \quad (3.57)$$

lo que demuestra la estacionariedad de  $\pi$  respecto del paseo.

Conociendo  $\pi$ , observar el comportamiento del paseo una vez ha entrado en su distribución estacionaria muestra que favorece recorrer conjuntos de nodos similares entre sí. Sean  $A \subset V$  y  $B = V \setminus A$  dos conjuntos que forman una bipartición de  $V$ , y  $P_{AB} = \Pr(A \rightarrow B | A)$  la probabilidad de que un salto del paseo aleatorio transicione a un nodo de  $B$  estando en  $A$ . Como la probabilidad condicionada de la cadena solamente depende del estado anterior, esta se calcula como el cociente entre la probabilidad de llegar a un nodo de  $B$  por una arista que interconecta ambos conjuntos, y la de que el nodo inicial pertenezca a  $A$ , de manera que

$$P_{AB} = \frac{\sum_{i \in A, j \in B} \pi_i P_{ij}}{\sum_{i \in A} \pi_i} = \frac{\sum_{i \in A, j \in B} \frac{d_i}{\text{vol}_V V} \frac{w_{ij}}{d_i}}{\sum_{i \in A} \frac{d_i}{\text{vol}_V V}} = \frac{\sum_{i \in A, j \in B} w_{ij}}{\sum_{i \in A} d_i} = \frac{\text{cut}(A, B)}{\text{vol}_V A}. \quad (3.58)$$

Aplicando la ecuación (3.58) en ambos sentidos, y teniendo en cuenta que el corte de aristas entre los conjuntos es simétrico, la probabilidad de que un paseo aleatorio no se mantenga en la agrupación de la que parte es

$$P_{AB} + P_{BA} = \left( \frac{1}{\text{vol}_V A} + \frac{1}{\text{vol}_V B} \right) \text{cut}(A, B) = \text{Ncut}(A, B). \quad (3.59)$$

Con este resultado, se deduce que minimizar el corte normalizado equivale a minimizar la probabilidad de que un paseo aleatorio en su distribución estacionaria salga de los límites del conjunto en el que comienza. En otras palabras, el proceso de recorrido del grafo permanece con mayor probabilidad dentro de la agrupación en la que origina. Segmentar trayectorias de puntos en un grafo bajo el criterio del corte normalizado, pues, puede entenderse como distribuir la evidencia de que estas pertenezcan a un objeto o al fondo por medio de un paseo aleatorio, que con suficientes pasos constituye agrupaciones distintivas.

En el caso de segmentar un grafo en más de dos particiones, en [35] se argumenta que una solución óptima del problema del corte normalizado en  $K$  grupos (es decir, la existencia de  $K$  vectores propios de  $\mathbf{P}$  constantes a trozos con valores propios ordenados y no nulos) se da si y solo si las sumas de las probabilidades de transición a los nodos de una agrupación son constantes para todos los vértices de un mismo conjunto, y estas se pueden condensar en una única matriz no singular de transición entre agrupaciones. En efecto, sea  $\Delta = (A_1, \dots, A_K)$  una partición de  $V$  derivada de los vectores propios de  $\mathbf{P}$  constantes a trozos  $\mathbf{x}^1, \dots, \mathbf{x}^K$ , con respectivos valores propios  $\lambda^1, \dots, \lambda^K$ , y la aplicación inyectiva  $\mathbf{x} \mapsto \mathbf{y}$  que para  $l = 1, \dots, K$  asocia a cada vector propio  $\mathbf{x}^l$  el vector  $K$ -dimensional  $\mathbf{y}^l$  cuyas componentes son las  $K$  diferentes constantes de los segmentos. Fijando  $i, i' \in A_s$  para  $s \in \{1, \dots, K\}$ , se tiene que

$$(\mathbf{Px}^l)_i = \sum_{n=1}^N P_{in} x_n^l = \sum_{s'=1}^K \left( \sum_{j \in A_{s'}} P_{ij} \right) y_{s'}^l = \lambda^l x_i^l \quad (3.60)$$

y  $(\mathbf{Px}^l)_{i'} = \lambda^l x_{i'}^l$  son las componentes  $i$  e  $i'$ -ésima de  $\mathbf{Px}^l$ . Como  $\mathbf{x}^l$  es constante a trozos y tanto  $i$  como  $i'$  pertenecen al mismo segmento, ambas componentes toman el mismo valor. Denotando  $\hat{P}_{is'} = \sum_{j \in A_{s'}} P_{ij}$  y restando  $(\mathbf{Px}^l)_i$  y  $(\mathbf{Px}^l)_{i'}$ , esto lleva a

$$\sum_{s'=1}^K (\hat{P}_{is'} - \hat{P}_{i's'}) y_{s'}^l = 0 \quad \forall l \in \{1, \dots, K\}, \quad (3.61)$$

### 3. SEGMENTACIÓN DE TRAYECTORIAS

---

un sistema homogéneo de  $K$  ecuaciones y  $K$  incógnitas con coeficientes  $y_s^l$ . Las entradas de la matriz del sistema están formadas por los  $K$  valores no repetidos de los  $K$  vectores propios de  $\mathbf{P}$ ; al ser estos linealmente independientes, dicha matriz es no singular, y el sistema de ecuaciones admite únicamente la solución trivial  $\hat{P}_{is'} - \hat{P}_{i's'} = 0$  para  $s' \in \{1, \dots, K\}$ . Como consecuencia de haber escogido  $i$  e  $i'$  arbitrariamente, se sostiene que para todo  $i \in A_s$ , las sumas  $\hat{P}_{is'}$  son constantes hacia cualquier segmento  $A_{s'}$ . Además, sea  $\hat{P}_{ss'}$  la notación de dicha probabilidad constante de transicionar de  $A_s$  a  $A_{s'}$ , la matriz  $\hat{\mathbf{P}} = [\hat{P}_{ss'}]_{s,s' \in \{1, \dots, K\}}$  cumple para todo  $l \in \{1, \dots, K\}$  que

$$\hat{\mathbf{P}}\mathbf{y}^l = \begin{bmatrix} \sum_{s=1}^K \hat{P}_{1s} y_1^l \\ \sum_{s=1}^K \hat{P}_{2s} y_2^l \\ \vdots \\ \sum_{s=1}^K \hat{P}_{Ks} y_K^l \end{bmatrix} = \begin{bmatrix} \sum_{s=1}^K (\sum_{j \in A_s} P_{ij} x_i^l) \\ \sum_{s=1}^K (\sum_{j \in A_s} P_{ij} x_i^l) \\ \vdots \\ \sum_{s=1}^K (\sum_{j \in A_s} P_{ij} x_i^l) \end{bmatrix} = \begin{bmatrix} \lambda^l x_i^l \\ \lambda^l x_i^l \\ \vdots \\ \lambda^l x_i^l \end{bmatrix} = \begin{bmatrix} \lambda^l y_1^l \\ \lambda^l y_2^l \\ \vdots \\ \lambda^l y_K^l \end{bmatrix} = \lambda^l \mathbf{y}^l, \quad (3.62)$$

donde el subíndice  $i$  indica un elemento cualquiera de la partición de la fila correspondiente. Por tanto,  $\hat{\mathbf{P}}$  tiene  $\mathbf{y}^1, \dots, \mathbf{y}^K$  por vectores propios,  $\lambda^1, \dots, \lambda^K$  por valores propios, y es no singular porque estos últimos no son nulos. Consecuentemente, una solución óptima del corte normalizado induce un paseo aleatorio con matriz  $\hat{\mathbf{P}}$  entre las  $K$  agrupaciones que divide. De la misma manera, si dicha  $\hat{\mathbf{P}}$  existe y es no singular, dada la aplicación exhaustiva  $\mathbf{y} \mapsto \mathbf{x}$  tal que  $x_i^l = y_s^l$  si  $i \in A_s$  para cada vector propio  $\mathbf{y}^l$  de  $\hat{\mathbf{P}}$  e  $i \in \{1, \dots, N\}$ , entonces

$$(\mathbf{Px}^l)_i = \sum_{s'=1}^K \left( \sum_{j \in A_{s'}} P_{ij} \right) y_{s'}^l = \sum_{s'=1}^K \hat{P}_{is'} y_{s'}^l = \lambda^l y_{s'}^l = \lambda^l x_i^l \quad \forall l \in \{1, \dots, K\}, \quad (3.63)$$

por lo que  $\mathbf{x}^1, \dots, \mathbf{x}^K$  son vectores propios de  $\mathbf{P}$  constantes a trozos con valores propios  $\lambda^1, \dots, \lambda^K$ , y del paseo aleatorio surge naturalmente un corte mínimo normalizado sobre el grafo.

La explicación intuitiva de la propiedad anterior es que un corte normalizado óptimo es el que permite condensar un paseo aleatorio sobre un grafo de  $N$  nodos en uno nuevo sobre  $K$  bloques de nodos sin perder información de su estructura original. La probabilidad de transicionar entre bloques en este último paseo, además, es mínima y no depende de los nodos originales sino del segmento al que pertenecen; de esta manera, el corte que crea las particiones es el que elimina todas las aristas del nuevo grafo de  $K$  vértices. Como consecuencia inmediata de este hecho, si se conoce el conjunto al que pertenecen algunos nodos de los  $K$  bloques, calcular la probabilidad de que un paseo que se inicia desde ellos llegue a cada uno de los demás nodos desconocidos conduce a una segmentación de todo el grafo.

#### 3.2.4 Segmentación interactiva

La interactividad en la segmentación de vídeo tiene el beneficio de poder adaptarse a la visión de un usuario o a una tarea sin necesidad de hacer suposiciones sobre ella. Admitir información externa sobre una secuencia de imágenes, por tanto, es una forma de afinar la precisión de su segmentación a cambio de una pérdida de automatización en su ejecución. Mientras que no existe un límite en la cantidad de interacción más allá de la realización de una segmentación manual, idealmente el proceso de segmentación debe dar buenos resultados con la menor interacción posible; un video puede estar

formado por millares de fotogramas, inviables de anotar en su totalidad, por lo que es deseable que únicamente sea necesario hacerlo sobre unos pocos. Así, en la práctica, un algoritmo de segmentación interactivo debe poder propagar información reducida tanto dentro de las imágenes parcialmente anotadas como a través del tiempo, y llegar a resultados coherentes respecto a ella.

La información adicional que otorga un usuario sobre una o varias imágenes de una secuencia se proporciona a través de *anotaciones*, o *semillas*. Definiendo el grafo de trayectorias de la Sección 3.2.1 y marcando inicialmente los nodos que pasan por un punto anotado de modo que queden preclasiificados, un algoritmo basado en paseos aleatorios es capaz de propagar dicha información al resto de trayectorias en forma de probabilidades de pertenencia a los diferentes conjuntos [37]. Para cada nodo sin marcar, iniciar un paseo aleatorio en él y etiquetarlo de la misma manera que el primer nodo marcado por el que este último pasa produce una segmentación completa de las trayectorias, que se agrupan según la tendencia del paseo a permanecer en sus particiones. Según lo visto en la Sección 3.2.3, esta segmentación se asemeja a la de un corte mínimo normalizado (sesgado por las semillas), y admite tantas divisiones como se desee. Grady [15] demuestra que para llegar a ella no es necesaria una simulación del paseo, sino que se puede formular el problema como un sistema de ecuaciones lineales con solución única.

Dado el conjunto de trayectorias  $\mathcal{C}$  y el grafo parcialmente anotado  $G = (V, E)$  que estas forman, el objetivo de la propagación de semillas es encontrar la matriz  $\mathbf{X}$  de probabilidades de que los nodos lleguen en primer lugar a cada conjunto a través de un paseo aleatorio. Sea  $\mathbf{x}^s$  la  $s$ -ésima columna de  $\mathbf{X}$ , correspondiente a las probabilidades de que el paseo se encuentre primero con la semilla  $s$ , el problema de obtener sus valores se puede resolver interpretando el resto de semillas como una sola, ya que la suma de todas ellas (incluida  $s$ ) debe ser la unidad. Sabiendo que  $x_i^s = 1$  para todo nodo  $i$  marcado que pertenece al conjunto  $s$ , y  $x_i^s = 0$  para aquellos que no,  $\mathbf{x}^s$  equivale al minimizador de la función  $D: [0, 1]^N \rightarrow \mathbb{R}$  definida como

$$D(\mathbf{x}) = \mathbf{x}^T (\mathbf{D} - \mathbf{W}) \mathbf{x} = \sum_{i, j \in V} w_{ij} (x_i - x_j)^2, \quad (3.64)$$

que se identifica como la versión combinatoria de la integral de Dirichlet [15]. En el caso de minimizar (3.64) sin las restricciones que vienen dadas por las semillas, las soluciones son aquellas que cumplen  $(\mathbf{D} - \mathbf{W})\mathbf{x} = \mathbf{0}$  (equivalentemente,  $\mathbf{x} = \mathbf{D}^{-1}\mathbf{W}\mathbf{x}$ ); por analogía a la versión continua, minimizada por funciones armónicas, se llama a  $\mathbf{L} = \mathbf{D} - \mathbf{W}$  el *laplaciano* del grafo  $G$ . La matriz  $\mathbf{L}$  es simétrica y semidefinida positiva ya que los pesos del grafo son no negativos, así que los únicos puntos críticos de (3.64) son mínimos. Si se dividen los vértices del grafo en dos conjuntos  $V_M$  y  $V_U$  que agrupan respectivamente los nodos que están marcados (independientemente de su etiqueta) y los que no, y suponiendo sin pérdida de generalidad que  $\mathbf{L}$  y  $\mathbf{x}$  están construidos con los nodos de  $V_M$  ordenados en primer lugar, la ecuación (3.64) se descompone como

$$\begin{aligned} D(\mathbf{x}_U) &= [\mathbf{x}_M^T \quad \mathbf{x}_U^T] \begin{bmatrix} \mathbf{L}_M & \mathbf{B} \\ \mathbf{B}^T & \mathbf{L}_U \end{bmatrix} \begin{bmatrix} \mathbf{x}_M \\ \mathbf{x}_U \end{bmatrix} \\ &= \mathbf{x}_M^T \mathbf{L}_M \mathbf{x}_M + 2\mathbf{x}_U^T \mathbf{B}^T \mathbf{x}_M + \mathbf{x}_U^T \mathbf{L}_U \mathbf{x}_U, \end{aligned} \quad (3.65)$$

donde  $\mathbf{x}_M$  y  $\mathbf{x}_U$  son respectivamente las probabilidades de llegada a la semilla correspondiente para nodos marcados y no marcados,  $\mathbf{L}_M$  y  $\mathbf{L}_U$  son los laplacianos de los

### 3. SEGMENTACIÓN DE TRAYECTORIAS

---

subgrafos que estos últimos forman de manera separada, y  $\mathbf{B}$  es la matriz de pesos negados en aristas que unen  $V_M$  y  $V_U$ . Como  $\mathbf{x}_M$  es conocido, diferenciar (3.65) con respecto de  $\mathbf{x}_U$  conduce a que la solución óptima satisface el sistema lineal de  $|V_U|$  ecuaciones

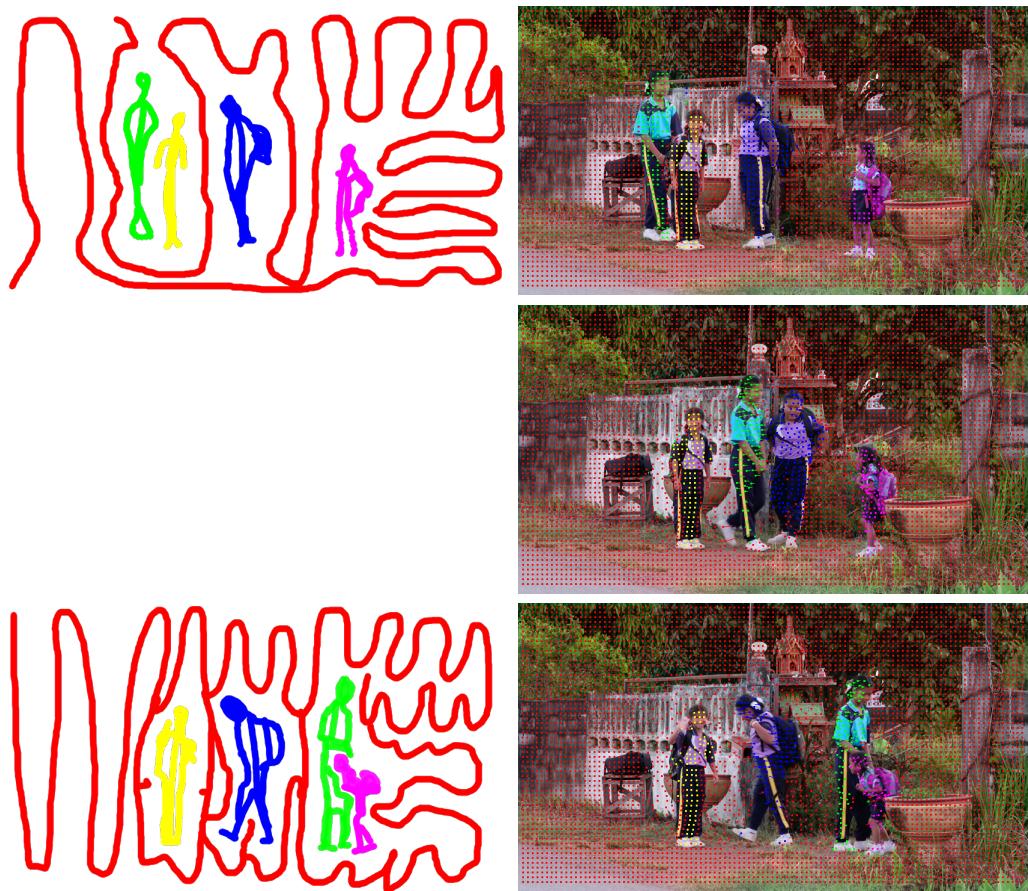
$$\mathbf{L}_U \mathbf{x}_U = -\mathbf{B}^T \mathbf{x}_M. \quad (3.66)$$

Si el grafo contiene como mínimo una semilla,  $\mathbf{L}_U$  es no singular y (3.66) tiene solución única [15]. De esta manera, utilizando que  $\sum_s x_i^s = 1$  para todo  $i \in V$ , todos los vectores  $\mathbf{x}^s$  (con  $s \in \{1, \dots, K\}$ ) se pueden obtener de manera independiente resolviendo dicho sistema  $K - 1$  veces. Ya que muchos pares de nodos no tienen aristas entre sí,  $\mathbf{L}_U$  es una matriz simétrica semidefinida positiva con una gran proporción de entradas nulas, por lo que (3.66) puede resolverse de manera eficiente con métodos iterativos como el del gradiente conjugado.

El proceso de segmentación interactiva de [37] comienza por la lectura de varios tipos de pinceladas proporcionadas por un usuario sobre un vídeo, que sirven para inicializar las  $K$  semillas y forzar que las zonas en las que se encuentran se clasifiquen como fondo u objeto. Cada trayectoria que pasa por al menos un punto anotado es asignada en su totalidad a su correspondiente conjunto. Si se da el caso de que haya anotaciones contradictorias para una trayectoria (es decir, que en tiempos distintos pase por semillas diferentes), esta se deja sin marcar; de esta manera, se evitan tanto errores humanos como los provocados por occlusiones o una estimación imprecisa del flujo óptico. A continuación, se construye un grafo a partir de las trayectorias, con pesos en las aristas dados por la ecuación (3.32), y se resuelve repetidamente el sistema (3.66) para obtener las probabilidades de pertenencia a cada partición de las trayectorias inicialmente no marcadas. Sea  $x_i^s$  la probabilidad de que la  $i$ -ésima trayectoria pertenezca a la  $s$ -ésima partición, los conjuntos  $A_s = \{c_i \in \mathcal{C} \mid x_i^s = \max_k x_i^k\}$  (con  $s \in \{1, \dots, K\}$ ) son una segmentación del conjunto de trayectorias  $\mathcal{C}$ . Específicamente, si las semillas están ordenadas con el fondo en primer lugar,  $\mathcal{F} = A_1$  y  $\mathcal{O} = \bigcup_{s=2}^K A_s$  conforman una segmentación binaria que separa el fondo del conjunto de todos los objetos del vídeo.

Las semillas que se difunden por medio del paseo aleatorio pueden llegar a propagarse a lo largo de elevadas cantidades de fotogramas, gracias a que las trayectorias son capaces de seguir puntos durante largos períodos de tiempo. Por el mismo motivo, la calidad de la segmentación final es dependiente de que las trayectorias sean fiables y cubran suficientemente los distintos objetos del vídeo. En particular, la occlusion de objetos, la presencia de grandes superficies sin textura, y un flujo óptico impreciso son los principales causantes de la acumulación de errores en el proceso de segmentación. La posibilidad de interactuar con el algoritmo permite corregir gran parte de estos errores: un mayor número de anotaciones ayuda a relacionar trayectorias distantes, y asegura que las que se reponen debido a occlusiones mantengan su dirección. La Figura 3.4 muestra cómo se etiquetan varios objetos a lo largo de sesenta fotogramas de la secuencia SCHOOLGIRLS de DAVIS [39] partiendo de dos fotogramas anotados; aunque dos de las niñas (verde y azul) se ocultan casi totalmente en un punto del vídeo, el hecho de que los fotogramas anotados ocurran antes y después de la occlusion hace que se segmenten correctamente en todo momento.

De manera general, en secuencias con pocos objetos y movimiento de cámara rígido, un único fotograma anotado es suficiente para que el paseo aleatorio de semillas dé resultados adecuados. En la Figura 3.5 se observan secuencias con un único



**Figura 3.4:** Segmentación interactiva de múltiples objetos sobre la secuencia SCHOOLGIRLS de DAVIS utilizando anotaciones en dos fotogramas. Primera columna: semillas en los fotogramas 1 y 61. Segunda columna: trayectorias de los fotogramas 1, 31, y 61 coloreadas según su clasificación. En rojo, las trayectorias del fondo; en verde, amarillo, azul, y magenta, las distintas niñas que forman los objetos del vídeo.

objeto, cuya segmentación se efectúa de manera precisa y con mínima interacción; comparadas con las de la Figura 3.1, obtenidas geométricamente por modelado de fondo, los errores de segmentación son muy reducidos, bajo el coste de la necesidad de intervenciones puntuales de un usuario. Por otro lado, secuencias más complejas pueden sufrir algunas imprecisiones; por ejemplo, en la Figura 3.6 se observa como partes del reflejo del cisne en BLACKSWAN se clasifican como objeto por desplazarse igual que el ave, y como el movimiento de la cámara en PARKOUR causa una constante pérdida de trayectorias que dificulta la delimitación del deportista. Anotaciones más frecuentes ayudan en casos como estos a corregir errores en las zonas donde se producen, pero suponen una mayor implicación por parte de un usuario para llegar a un resultado ideal, y no son efectivas si por las zonas anotadas no pasan trayectorias. Aun así, la mejora en la calidad de la segmentación respecto a métodos no supervisados es notoria, especialmente en vídeos en los que el movimiento global es complejo.

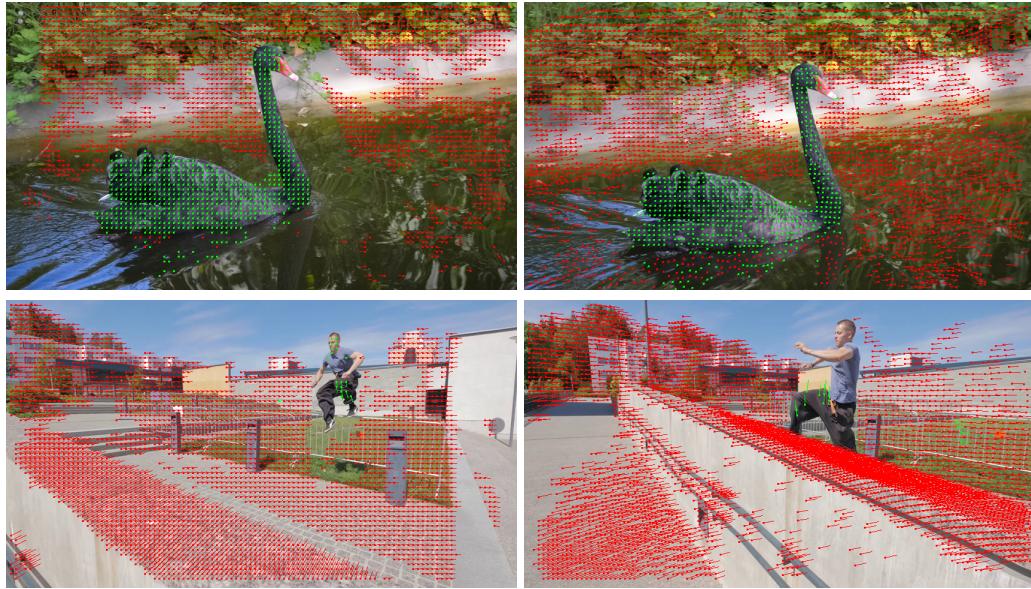
Las imágenes de las figuras 3.4, 3.5, y 3.6 son los resultados de la ejecución de una implementación propia en C++ del algoritmo de segmentación interactiva de [37].

### 3. SEGMENTACIÓN DE TRAYECTORIAS

---



**Figura 3.5:** Trayectorias segmentadas de manera interactiva en secuencias de DAVIS con un solo objeto. De arriba a abajo: BMX-BUMPS, BEAR, HIKE, SOAPBOX, TRAIN. En la primera columna, el primer fotograma de las secuencias; en la segunda, el vigesimoprimero. Un único fotograma ha sido anotado en cada una de ellas.



**Figura 3.6:** Segmentación de trayectorias interactiva en secuencias de DAVIS con movimiento complejo. Primera fila: fotogramas número 1 y 21 de BLACKSWAN. Segunda fila: fotogramas número 1 y 21 de PARKOUR. Un único fotograma ha sido anotado en ambos videos.

Para obtenerlas, se ha utilizado como parámetro  $\lambda = 0.1$  en la ecuación (3.32), y se han admitido como nodos del grafo de trayectorias únicamente aquellas de duración mayor que cinco fotogramas, con objetivo de reducir los efectos negativos del ruido. La biblioteca Eigen [16] se ha empleado para resolver eficientemente el sistema (3.66), así como con OpenCV [6] se han procesado las imágenes de las distintas secuencias.



CAPÍTULO



# 4

## SEGMENTACIÓN DE REGIONES DE PÍXELES

Pese a ser indispensable en la segmentación en vídeo, el movimiento no es la única herramienta para detectar objetos. El fondo, salvo excepciones de carácter artístico, se considera como tal por ser de menor interés visual que el resto de la escena; es natural que una persona sea capaz de distinguir a simple vista la porción de imagen con más protagonismo, incluso observando únicamente un fotograma de una secuencia. La distribución de colores de diferentes regiones de un objeto, así como su textura, son comúnmente distintivas y similares entre sí. Asimismo, las mismas regiones se repiten frecuentemente en varios fotogramas, sometiéndose a cambios mínimos que las hacen reconocibles. Bajo esta premisa, una estrategia para segmentar objetos en un vídeo consiste en identificar regiones candidatas a formar parte de un objeto, y relacionarlas con otras semejantes en el resto del vídeo.

Conectar regiones similares en intervalos de tiempo extensos es insuficiente para clasificarlas, puesto que a largo plazo los objetos tienden a modificar su apariencia. Transformaciones no rígidas, cambios de iluminación, y cambios de escala son ejemplos de alteraciones que hacen que una medida de semejanza se vuelva poco fiable con el tiempo. Por otro lado, señales a corto plazo como el movimiento son imprecisas e incapaces de describir escenas complejas, como olas en el mar. Información a corto y a largo plazo, sin embargo, se pueden combinar para suplir sus carencias. Faktor e Irani [12] proponen un esquema para segmentar vídeos de carácter general de manera no supervisada, que plantea el problema de clasificación como una votación en varias fases, en la que distintas regiones de un vídeo pueden influenciar el voto de aquellas con las que tienen un mayor parecido.

El método de [12], Non-Local Consensus Voting (NLCV), utiliza información local de movimiento para medir de forma aproximada el grado de certeza de que cada píxel de las imágenes de una secuencia forme parte de un objeto. A estos se les asigna un valor inicial (un *voto*), posiblemente impreciso y perturbado por ruido en el vídeo, que cuantifica la evidencia de que no pertenezcan al fondo. Por medio de la información a largo plazo, los votos se corrigen iterativamente hasta llegar a un *consenso*, esencialmente actualizando sus valores en función de los votos de las regiones con apariencia

## 4. SEGMENTACIÓN DE REGIONES DE PÍXELES

---

más cercana a la suya. Estableciendo mínimas restricciones sobre la distancia espacial y temporal entre regiones, la similitud entre ellas toma prioridad al propagar sus votos, lo que posibilita rectificar errores en ellos con la información de la totalidad del vídeo. Al contrario que los métodos basados en trayectorias, las correspondencias que se obtienen de esta manera no son secuenciales en el tiempo, y su exactitud no se ve afectada por la calidad de la estimación del flujo óptico, que tiende a deteriorarse con la extensión de la trayectoria. En cambio, estas se definen de forma probabilística a partir de una medida de semejanza, evitando así la acumulación de errores que proviene del rastreo de puntos. Como las regiones son independientes del flujo óptico, además, NLCV permite asociar zonas sin textura entre sí, no existiendo áreas intratables; eligiendo adecuadamente dichas agrupaciones de píxeles, por tanto, este método da como resultado una segmentación densa.

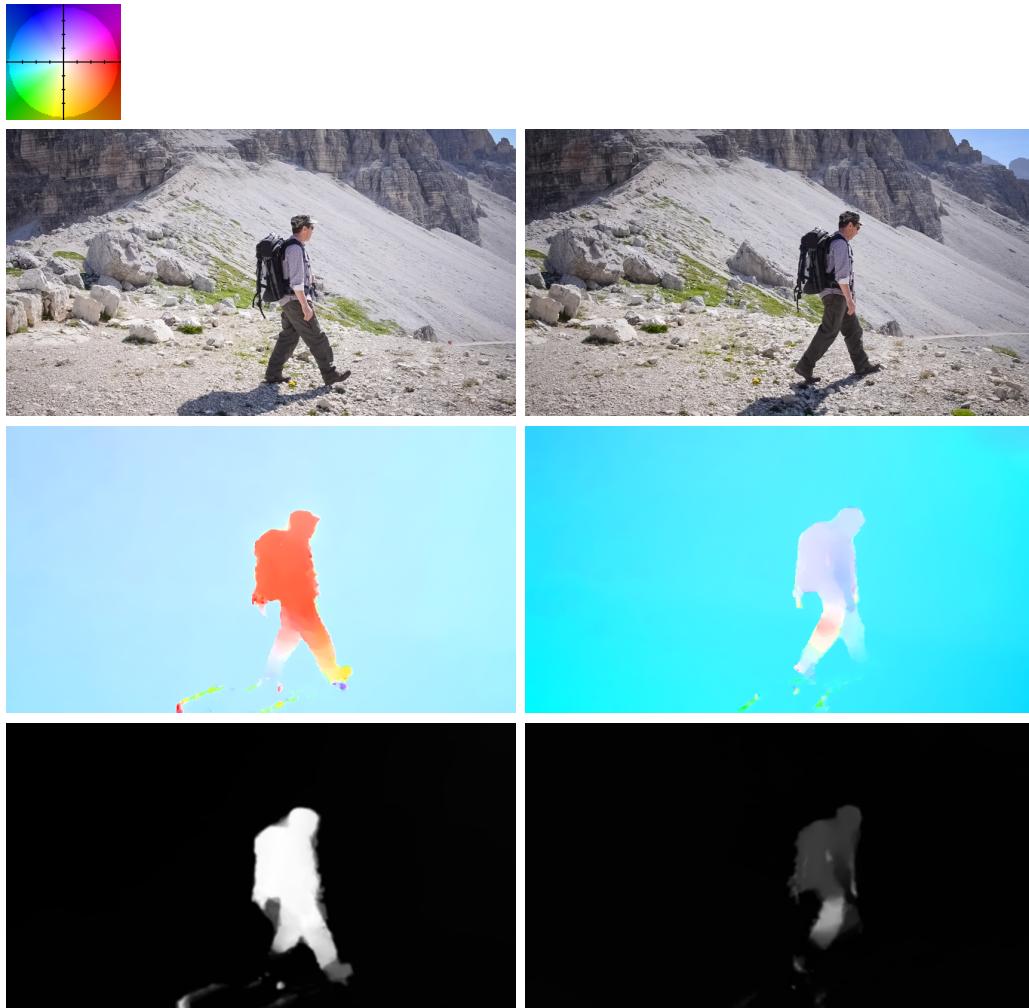
NLCV se divide en cuatro etapas. Primero, a partir de los campos de flujo óptico, se obtienen *mapas de prominencia* (*saliency maps*) que cuantifican el interés visual de los píxeles de cada fotograma. A continuación, las imágenes de la secuencia se dividen en regiones, a las que se les atribuye un voto inicial basado en la prominencia de los píxeles que contienen. Dichas regiones se representan por medio de un descriptor, incorporándose en un espacio en el se pueden comparar según su apariencia. Finalmente, se identifican conjuntos de descriptores cercanos a nivel de apariencia, y se actualizan los votos de las regiones conforme a los de sus vecinas.

### 4.1 Mapas de prominencia

Un mapa de prominencia asociado a una imagen es otra imagen de un canal que otorga a cada uno de sus píxeles un valor numérico que mide una cualidad en relación al resto de la imagen. En concreto, un mapa de prominencia de movimiento define esa cualidad como la forma de destacar del desplazamiento de los píxeles entre fotogramas respecto al movimiento global. La elaboración de un mapa de prominencia tiene como objetivo poder localizar de manera sencilla los puntos de una imagen que son de interés para un problema, y cuantificar dicho interés para su posterior uso.

El algoritmo NLCV comienza por el análisis del movimiento local de cada uno de los fotogramas de una secuencia a partir de sus campos de flujo óptico, que busca encontrar patrones de movimiento. Específicamente, son de interés aquellas imágenes que muestran un desplazamiento dominante, interpretado como la concordancia de la magnitud u orientación de sus vectores de flujo en la mayoría de píxeles, que posibilita una separación inicial de objetos y fondo más fiable. Adquirir el mapa de prominencia de un fotograma con movimiento dominante se reduce a detectar los píxeles cuyo flujo óptico no se ajusta al patrón de movimiento; si la mayoría de imágenes lo exhibe, los valores que estos contienen son suficientes para inicializar los votos del algoritmo. Seleccionar los fotogramas previamente al cálculo de los mapas tiene la ventaja de filtrar aquellos que tienen un campo de flujo poco fiable. Al resto de fotogramas no seleccionados se les atribuye un mapa con prominencia nula (y por tanto, voto igual a 0), que se corrige en la fase de consenso siempre y cuando no constituyan la mayoría del vídeo.

El caso más sencillo de movimiento dominante se da cuando la cámara y el fondo se mantienen estáticos. En dicha circunstancia, la magnitud del flujo óptico de la mayoría de píxeles es cercana a 0, por lo que todo punto con un desplazamiento significativo



**Figura 4.1:** Prominencia de movimiento en los fotogramas 1 y 31 de la secuencia HIKE de DAVIS. Fila 1: codificación de color de los campos de vectores de flujo óptico. Fila 2: fotogramas originales. Fila 3: campos de flujo óptico TV-L<sup>1</sup> hacia adelante en el tiempo. Fila 4: mapas de prominencia. Un blanco más intenso indica una mayor prominencia.

destaca por encima de los demás. Bajo la suposición de que la mayoría de fotogramas de un vídeo digital son estáticos, dado el campo de flujo óptico  $\mathbf{u}_t$  del  $t$ -ésimo de ellos, su prominencia  $S_t: \Omega \rightarrow \mathbb{R}$  se mide como

$$S_t(\mathbf{x}) = \begin{cases} \|\mathbf{u}_t(\mathbf{x})\|_2^2 & \text{si } \tilde{u}_t < 1, \\ 0 & \text{en otro caso,} \end{cases} \quad (4.1)$$

donde  $\tilde{u}_t = \text{med}\{\|\mathbf{u}_t(\mathbf{x})\|_2 : \mathbf{x} \in \Omega\}$  es la mediana del conjunto finito de las magnitudes de los vectores de flujo. La condición  $\tilde{u}_t < 1$  exige que la mayoría de los puntos de la imagen no se desplace más de un píxel al pasar a la siguiente; en caso de no cumplirse, la prominencia de todo  $\mathbf{x} \in \Omega$  se toma como nula. Es importante notar que la suposición de movimiento dominante estático quiere decir que  $\tilde{u}_t < 1$  para al menos la mitad de los fotogramas del vídeo.

Un movimiento dominante translacional es el segundo tipo de patrón que intenta detectar NLCV. Si la cámara y el fondo no están fijos, y la mayoría de los vectores de flujo óptico tienen una orientación (no necesariamente magnitud) común, los píxeles que se desplazan en un sentido diferente al dominante son más prominentes. En el caso de que más de la mitad de los fotogramas estén en esta situación, sus mapas de prominencia pueden obtenerse a partir de los histogramas de orientaciones de sus campos de flujo óptico. Sea  $\theta_t \in [0, 2\pi]$  la marca de clase del grupo de orientaciones más frecuente de la imagen  $I_t$  y  $v_t \in [0, 1]$  su frecuencia relativa, la prominencia de los puntos que hay en ella se obtiene como

$$S_t(\mathbf{x}) = \begin{cases} [(\text{Arg } \mathbf{u}_t(\mathbf{x}) - \theta_t) \bmod 2\pi]^2 & \text{si } v_t > 0.5, \\ 0 & \text{en otro caso,} \end{cases} \quad (4.2)$$

siendo  $\text{Arg } \mathbf{u}_t(\mathbf{x})$  el valor principal del ángulo que forma  $\mathbf{u}_t(\mathbf{x})$  con el eje de abscisas, y donde la operación  $\varphi \bmod 2\pi$  mueve el ángulo  $\varphi$  al rango  $[0, 2\pi]$ . La condición  $v_t > 0.5$  requiere que la mayoría de vectores de flujo apunten en el mismo sentido; si  $v_t > 0.5$  para más de la mitad de  $t \in \{1, \dots, T\}$ , se cumple la hipótesis de movimiento dominante translacional y la ecuación (4.2) es utilizada por NLCV. La Figura 4.1 muestra ejemplos de mapas de prominencia de una secuencia con movimiento dominante translacional, junto con la representación de sus campos de flujo óptico.

En escenas con un movimiento más complejo, es posible buscar transformaciones dominantes más generales, como afinidades y homografías, y determinar los mapas de prominencia a partir del ajuste de los píxeles a su modelo. En [12], no obstante, se opta por la prominencia visual si no se detectan suficientes fotogramas con movimiento dominante estático o translacional. Pese a ser menos precisa por no utilizar la información que da el movimiento entre imágenes, esta última mide la desviación respecto de estructuras espaciales que se repiten y es aplicable individualmente a los fotogramas de cualquier tipo de vídeo.

Los valores que dan las fórmulas (4.1) y (4.2) y los de prominencia visual están sujetos al ruido que puede haber en las imágenes, por lo que [12] recomienda suavizar los mapas resultantes con un filtro de media en parches de  $5 \times 5$  píxeles. Además, para transformarlos en votos, conviene que se encuentren en un rango acotado. Como la prominencia de un punto es una cualidad relativa a los valores que toman los demás, NLCV define mapas normalizados  $\tilde{S}_t$  de manera que

$$\tilde{S}_t(\mathbf{x}) = \frac{S_t(\mathbf{x})}{\max_{t, \mathbf{x}} S_t(\mathbf{x})}. \quad (4.3)$$

Así, el píxel con mayor prominencia en todo el vídeo sirve como punto de referencia (con valor igual a 1), y el resto destaca en mayor o menor medida respecto de él. Consecuentemente, un punto  $\mathbf{x} \in \Omega$  de un fotograma  $t$  con  $\tilde{S}_t(\mathbf{x}) \approx 0$  es probablemente parte del fondo, mientras que cuando más se acerque  $\tilde{S}_t(\mathbf{x})$  a 1, hay más indicios de que sea un objeto, siempre manteniendo en consideración que los votos iniciales pueden no corresponderse con la realidad.

## 4.2 Extracción de regiones

El esquema de votación que plantea NLCV está basado en regiones, agrupaciones conexas de píxeles que facilitan la identificación de áreas con características similares y

limitan los efectos del ruido sobre el algoritmo. Los valores de prominencia de cada píxel se promedian dentro de la región en la que se encuentran, constituyendo un voto por cada una de ellas. La adecuación de los resultados del algoritmo a la realidad, como consecuencia, depende de la calidad de la selección de estas regiones. Una región apropiada de una imagen es aquella coherente tanto visual como espacialmente, y que delimita correctamente áreas de apariencia distinta, sin contener al mismo tiempo puntos de objetos y de fondo.

Las regiones se definen como superpíxeles, grupos de píxeles con características en común. Existen distintos métodos para dividir una imagen en ellos; en concreto, [12] utiliza una transformación divisoria al resultado de aplicar sobre la imagen un detector de bordes previamente entrenado. No obstante, Simple Linear Iterative Clustering (SLIC) [1] ha demostrado ser igualmente apto y de implementación más sencilla, de modo que se ha preferido su uso en la implementación de NLCV de este trabajo.

SLIC genera superpíxeles agrupando píxeles según la similitud de sus colores y su proximidad espacial. Sea  $I: \Omega \rightarrow \mathbb{R}^3$  una imagen a color, el algoritmo se fundamenta en aplicar un método de *clustering* sobre los puntos del espacio definido por las componentes  $L, a, b$  del espacio de color  $L^*a^*b^*$ , y las coordenadas  $x, y$  de los píxeles. El uso de CIELab ayuda a que la medida de diferencia entre píxeles sea representativa de la realidad, y que estos se agrupen acorde a la visión humana. Si  $\Omega$  está constituido por  $N$  píxeles y a la entrada del algoritmo se instruye que se desea dividir  $I$  en  $K$  regiones, SLIC comienza inicializando  $K$  centros de *clusters* equidistantiados en intervalos de  $M = \sqrt{N/K}$  píxeles, a los cuales se les asocian puntos vecinos según una distancia que combina diferencias de color y de posición. Como el área media de los superpíxeles es de  $M^2$  píxeles, se asume que aquellos ligados a cada centro se encuentran en una extensión de tamaño  $2M \times 2M$  alrededor de él; el área de búsqueda de píxeles candidatos a formar parte de un *cluster*, por tanto, se limita a esta zona.

La distancia utilizada para agrupar píxeles ha de encontrar un balance entre la diferencia perceptual de los colores de la imagen y la de la posición de sus píxeles. Por un lado, el espacio de color  $L^*a^*b^*$  es perceptiblemente significativo en distancias pequeñas, por lo que interesa penalizar que estas superen cierto valor  $m$ . Por el otro, la distancia euclídea entre las coordenadas de los píxeles se ve afectada por la resolución de la imagen, necesitando normalizar sus coordenadas por el tamaño del área en la que se encuentran. Teniendo esto en consideración, en [1] se define la distancia

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\left(\frac{\|I(\mathbf{x}) - I(\mathbf{y})\|_2}{m}\right)^2 + \left(\frac{\|\mathbf{x} - \mathbf{y}\|_2}{M}\right)^2}, \quad (4.4)$$

que sirve a SLIC para situar los píxeles en los distintos *clusters*. El valor de  $m$  controla la compacidad de los superpíxeles: un valor elevado da más peso a la proximidad espacial en (4.4), mientras que valores más pequeños dan más libertad a las formas que estos pueden tomar. Además de las diferencias de color y posición propuestas en [1], también es posible obligar al algoritmo a que tenga en cuenta otros criterios de disimilitud. En el caso de la implementación de NLCV de este trabajo, se ha utilizado una modificación de (4.4) que pondera la diferencia de valores de prominencia de los píxeles de la imagen, cuyo cuadrado se suma al resto de distancias dentro de la raíz cuadrada. La inclusión de la prominencia en la distancia de SLIC aumenta la precisión de las regiones incorporando la información del movimiento que no existe en fotografías individuales.

#### 4. SEGMENTACIÓN DE REGIONES DE PÍXELES

---



---

**Algoritmo 4.1:** SLIC

---

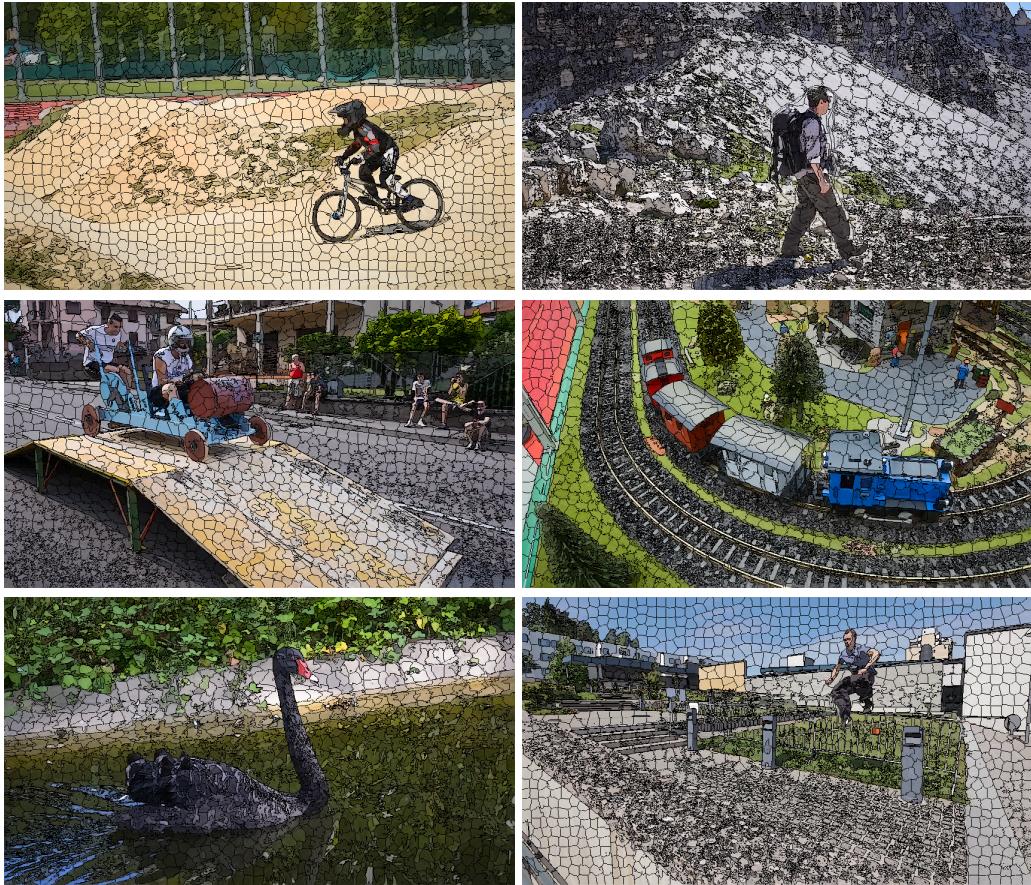
**Datos:** Imagen  $I: \Omega \rightarrow \mathbb{R}^3$  de  $N$  píxeles  
**Parámetros:** Umbral  $\epsilon$ ; número de regiones  $K$   
**Resultado:** Etiquetado de píxeles  $l: \Omega \rightarrow \{1, \dots, K\}$

- 1 Transformar  $I$  a espacio de color  $L^*a^*b^*$
- 2  $M \leftarrow \sqrt{N/K}$
- 3  $c_k \leftarrow (\mathbf{x}_k, I(\mathbf{x}_k))$  con  $k \in \{1, \dots, K\}$  para muestra de puntos cada  $M$  píxeles
- 4 Mover  $c_k$  a su vecino con gradiente más bajo en una ventana de  $3 \times 3$  píxeles
- 5  $D(\mathbf{x}) \leftarrow +\infty \quad \forall \mathbf{x} \in \Omega$
- 6 **repetir**
- 7   **para**  $k \in \{1, \dots, K\}$  **hacer**
- 8     **para** cada píxel  $\mathbf{x}$  en una región de  $2M \times 2M$  alrededor de  $c_k$  **hacer**
- 9        $D' \leftarrow d(\mathbf{x}, c_k)$  ▷ Distancia (4.4)
- 10       **si**  $D' < D(\mathbf{x})$  **entonces**
- 11          $D(\mathbf{x}) \leftarrow D'$  ▷ Píxel cambia de cluster
- 12          $l(\mathbf{x}) \leftarrow k$
- 13       **fin**
- 14     **fin**
- 15   **fin**
- 16    $C_k \leftarrow \{\mathbf{x} \in \Omega \mid l(\mathbf{x}) = k\} \quad \forall k \in \{1, \dots, K\}$
- 17    $r \leftarrow \sum_{k=1}^K \|c_k - \frac{1}{|C_k|} \sum_{\mathbf{x} \in C_k} \mathbf{x}\|_1$
- 18    $c_k \leftarrow \frac{1}{|C_k|} \sum_{\mathbf{x} \in C_k} \mathbf{x} \quad \forall k \in \{1, \dots, K\}$  ▷ Actualización de los centros
- 19 **hasta que**  $r < \epsilon$
- 20 **devolver**  $l$

---

El Algoritmo 4.1 [1] resume el procedimiento completo que sigue SLIC para delimitar las regiones de una imagen. Una vez elegidos los centros de cada *cluster* y definida la distancia (4.4), SLIC itera sobre los vecindarios de  $2M \times 2M$  píxeles de cada uno de los centros para clasificarlos en el mismo *cluster* que su centro más cercano. Después de etiquetar todos los puntos de la imagen, los centros de cada superpíxel se recalculan como el baricentro de los píxeles que contienen, y el proceso se repite hasta que la posición de estos últimos deja de variar. Las zonas con mucha textura pueden causar imprecisiones en el cálculo de las distancias a los centros, exagerándolas; por tanto, con la finalidad de evitar que estos últimos se ubiquen en esquinas, y para reducir la probabilidad de que se inicialicen en un píxel ruidoso, [1] recomienda moverlos previamente a los puntos de gradiente más pequeño de su entorno de  $3 \times 3$  píxeles más cercanos. Además, opcionalmente, es posible imponer que las regiones sean conexas dejando que los píxeles huérfanos de una agrupación sean absorbidos por su *cluster* más cercano espacialmente, lo que puede reducir el número de superpíxeles.

En la Figura 4.2 se muestran ejemplos de la división en regiones de distintas imágenes de DAVIS, en las que se opta por no imponer conectividad. Se puede observar como el algoritmo divide las imágenes de manera perceptiblemente significativa: los superpíxeles demarcán de forma adecuada la frontera de los objetos, así como agrupan píxeles cercanos de apariencia similar. En las áreas con más textura se hace evidente que la diferencia de apariencia toma mayor importancia en (4.4), creando regiones



**Figura 4.2:** Separación de imágenes en regiones utilizando el algoritmo SLIC modificado para considerar prominencia, usando  $m = 10$ ,  $M = 16$  y sin imponer conectividad. De izquierda a derecha, arriba a abajo: BMX-BUMPS, HIKE, SOAPBOX, TRAIN, BLACKSWAN, PARKOUR.

disconexas; en las planas, por contra, la distancia espacial tiene un peso mayor y las regiones no se fraccionan. La escala de apariencia y tamaño de superpíxel que se han utilizado son  $m = 10$  y  $M = 16$ ; superpíxeles más pequeños son más sensibles al ruido y aumentan la complejidad computacional del algoritmo, mientras que superficies grandes disminuyen su precisión al delimitar bordes.

### 4.3 Descripción de regiones

Una vez extraídas las regiones de cada uno de los fotogramas de una secuencia, y disponiendo de sus mapas de prominencia, a cada una de ellas se les otorga un voto inicial. Sea  $\{C_k^t\}_{k=1}^K$  el conjunto de regiones disjuntas tales que  $\bigcup_{k=1}^K C_k^t = \Omega$  para el  $t$ -ésimo fotograma  $I_t: \Omega \rightarrow \mathbb{R}^d$  de un vídeo, y  $\tilde{S}_t: \Omega \rightarrow [0, 1]$  el mapa de prominencia normalizado de este. El voto inicial de cada región  $C_k^t$  se calcula como

$$v^0(C_k^t) = \frac{1}{\int_{C_k^t} dC_k^t} \int_{C_k^t} \tilde{S}_t(\mathbf{x}) dC_k^t, \quad (4.5)$$

#### 4. SEGMENTACIÓN DE REGIONES DE PÍXELES

---

que es la media de los valores de prominencia sobre  $C_k^t$ . En imágenes digitales, (4.5) se simplifica a la suma de las prominencias de la región, dividida entre el número de píxeles que la forman.

Para medir cuantitativamente el grado de similitud de dos regiones distintas, es necesario representarlas dentro de un espacio en el que se puedan comparar sus características distintivas. Es por ello que a cada una se le atribuye un descriptor: un elemento de un espacio vectorial que concatena valores que dan información sobre su apariencia, textura, y posición, entre otros. A partir de ellos, la desemejanza entre dos regiones  $R_1$  y  $R_2$  se mide como la distancia entre sus descriptores  $\mathcal{D}(R_1)$  y  $\mathcal{D}(R_2)$ .

NLCV utiliza tres tipos de descriptores: histogramas de colores, histogramas de gradientes, y coordenadas relativas de los superpíxeles. Un histograma de color en espacio RGB y otro en L\*a\*b\* describen la apariencia de las regiones, cada uno con veinte contenedores por componente, recontando las frecuencias relativas de los colores que aparecen en ellas en un total de ciento veinte valores comprendidos entre 0 y 1. Por otro lado, para describir su textura, se utilizan Histograms of Oriented Gradients (HOGs) sobre nueve celdas de  $5 \times 5$  píxeles alrededor de sus centros, subdividiendo las orientaciones en seis contenedores por celda. El descriptor HOG calcula los gradientes individuales de los píxeles de cada celda, computa las frecuencias absolutas de sus orientaciones dentro de ella, y las normaliza sobre los bloques de  $15 \times 15$  que forman las celdas en su conjunto, resultando en cincuenta y cuatro valores descriptivos. Finalmente, la posición del centro de los superpíxeles se guarda en forma de coordenadas relativas normalizando sus componentes para que tomen valores entre 0 y 1.

Sea  $\mathcal{R}$  el conjunto de todos los superpíxeles de un vídeo, independientemente del fotograma en el que aparezcan, y  $\mathcal{D}: \mathcal{R} \rightarrow [0, 1]^{176}$  el descriptor definido por NLCV que se ha explicado anteriormente. A partir de este último, la aplicación

$$\begin{aligned} d: \mathcal{R} \times \mathcal{R} &\rightarrow [0, +\infty) \\ (R_1, R_2) &\mapsto \|\mathcal{D}(R_1) - \mathcal{D}(R_2)\|_2 \end{aligned} \tag{4.6}$$

es una pseudodistancia que mide la desemejanza entre dos regiones. Sus propiedades de no negatividad, simetría, y desigualdad triangular derivan inmediatamente de la norma euclídea, pero puede darse el caso (poco común) de que dos regiones distintas tengan el mismo descriptor, lo que no es importante para el algoritmo. Es a partir de (4.6) que NLCV busca regiones de características similares, que pueden encontrarse lejos en el tiempo y no ser conexas en el espacio por la naturaleza del descriptor escogido, y propaga sus votos entre ellas hasta llegar a un consenso.

#### 4.4 Propagación de votos

Los votos iniciales que da la fórmula (4.5) sufren dos inconvenientes que los hacen inadecuados para clasificar de manera precisa sus regiones como objetos o fondo. Primero, están basados puramente en la información local de prominencia, ruidosa y dependiente de que exista movimiento continuado en la totalidad de los objetos. Segundo, es posible que algunos de estos votos no se hayan inicializado a un valor significativo por la imposibilidad de obtener los mapas de prominencia de ciertos fotogramas sin movimiento dominante. Propagar iterativamente dichos votos entre regiones similares incita a que regiones vecinas se influencien mutuamente hasta llegar

a un consenso respaldado por la mayoría. Si una fracción suficientemente grande de las regiones tiene un voto inicial coherente con la realidad, por tanto, este procedimiento corrige errores iniciales y conduce a una segmentación refinada.

Es indeseable que el voto de una región altere los votos de zonas con las que no tiene parecido. Es por ello que es importante que estos solamente se propaguen a las regiones vecinas más cercanas según la pseudodistancia (4.6). Para cada  $R \in \mathcal{R}$ , NLCV busca correspondencias en un radio temporal de  $F$  fotogramas hacia adelante y hacia atrás, incluido el propio de  $R$ , las cuales son las únicas que pueden influenciar su voto. Cada superpíxel puede asociarse con varios vecinos en tiempos distintos, ya que al erigirlos no se establece coherencia temporal entre ellos, y diferentes zonas de los mismos objetos (o fondo) tienden a tener descriptores similares. Así, a cada región se le asocian  $L(2F + 1)$  vecinos, donde  $L = 4$  y  $F = 15$  son los valores recomendados en [12]. La notación  $\mathcal{N}_i(R)$  se utiliza para indicar el  $i$ -ésimo vecino más cercano de  $R$ , de manera que

$$d(R, \mathcal{N}_{i-1}(R)) \leq d(R, \mathcal{N}_i(R)) \leq d(R, \mathcal{N}_{i+1}(R)) \quad (4.7)$$

para todo  $i \in \{2, \dots, L(2F + 1) - 1\}$ , y  $\mathcal{N}(R) = \{\mathcal{N}_i(R) \in \mathcal{R} \mid 1 \leq i \leq L(2F + 1)\}$  representa el conjunto de todos los  $L(2F + 1)$  vecinos.

La propagación de los votos se modela como un paseo aleatorio en un grafo dirigido cuyos nodos son las regiones de  $\mathcal{R}$ , con aristas hacia regiones vecinas y un bucle en cada vértice. A partir de la pseudodistancia (4.6), a cada arista que va desde  $R \in \mathcal{R}$  a  $\mathcal{N}_i(R)$  se le atribuye un valor de similitud definido por

$$w(R, \mathcal{N}_i(R)) = \exp\left(-\frac{d(R, \mathcal{N}_i(R))^2}{\sigma^2}\right), \quad (4.8)$$

que expresa la influencia que tiene el voto de  $R$  sobre el de  $\mathcal{N}_i(R)$ , siendo  $\exp$  la función exponencial y  $\sigma^2$  un parámetro de escala que depende de la velocidad del movimiento en el vídeo. De (4.8) se define la matriz de pesos del grafo  $\mathbf{W} = (W_{ij})$ , con  $|\mathcal{R}| \times |\mathcal{R}|$  entradas determinadas por

$$W_{ij} = \begin{cases} w(R_i, R_j) & \text{si } R_j \in \mathcal{N}(R_i), \\ 1 & \text{si } i = j, \\ 0 & \text{en otro caso,} \end{cases} \quad (4.9)$$

y se obtiene la matriz de paseo aleatorio como  $\mathbf{P} = \mathbf{D}^{-1} \mathbf{W}$ , donde  $\mathbf{D} = \text{diag}(\mathbf{W}\mathbf{1})$ . La entrada  $P_{ij}$  de  $\mathbf{P}$  contiene la probabilidad de que un salto aleatorio desde el nodo  $i$  llegue al nodo  $j$ ; desde el punto de vista del grafo construido por NLCV, este valor equivale a la influencia relativa que tiene la región  $R_j$  sobre  $R_i$ : un paso del paseo otorga a  $R_i$  una porción  $P_{ij}$  del voto de  $R_j$ . Aplicar un paseo aleatorio sobre el grafo de regiones, pues, redistribuye los votos de manera que cada región recibe un porcentaje del voto de las regiones para las que es vecina.

Sea  $\mathbf{v}^0$  el vector que contiene los votos iniciales de todas las regiones, obtenidas con la expresión (4.5). El proceso de difusión de votos se sintetiza en aplicar la recursión

$$\mathbf{v}^k = \mathbf{P} \mathbf{v}^{k-1} \quad (4.10)$$

una cantidad  $K$  de veces, normalizando los votos al intervalo  $[0, 1]$  en cada iteración. Como la suma de los elementos de las filas de  $\mathbf{P}$  es igual a 1, cada voto  $v_i$  de  $R_i$  se

actualiza como una media ponderada del resto, de los que solo contribuyen aquellos  $v_j$  de  $R_j \in \mathcal{N}(R_i)$  y el propio  $v_i$ . Cabe notar que  $R_j \in \mathcal{N}(R_i)$  no implica que  $R_i \in \mathcal{N}(R_j)$ , así que la suma de las columnas de  $\mathbf{P}$  no es necesariamente 1; esto significa que durante la propagación, un voto puede repartirse en déficit o en exceso. El paso de normalización que sigue a (4.10) es indispensable para que cada región tenga la misma participación, y no se propaguen errores entre iteraciones derivados de una votación injusta.

De la ecuación (4.10) se ve que, con suficientes iteraciones, si la pluralidad de los votos de una región  $R$  sugiere que forma parte de un objeto,  $R$  acaba adaptando su voto para indicar lo mismo aunque inicialmente no lo hiciese. Asimismo, si la mayoría insinúa lo contrario, el voto de  $R$  pierde fuerza hasta decirlo también.

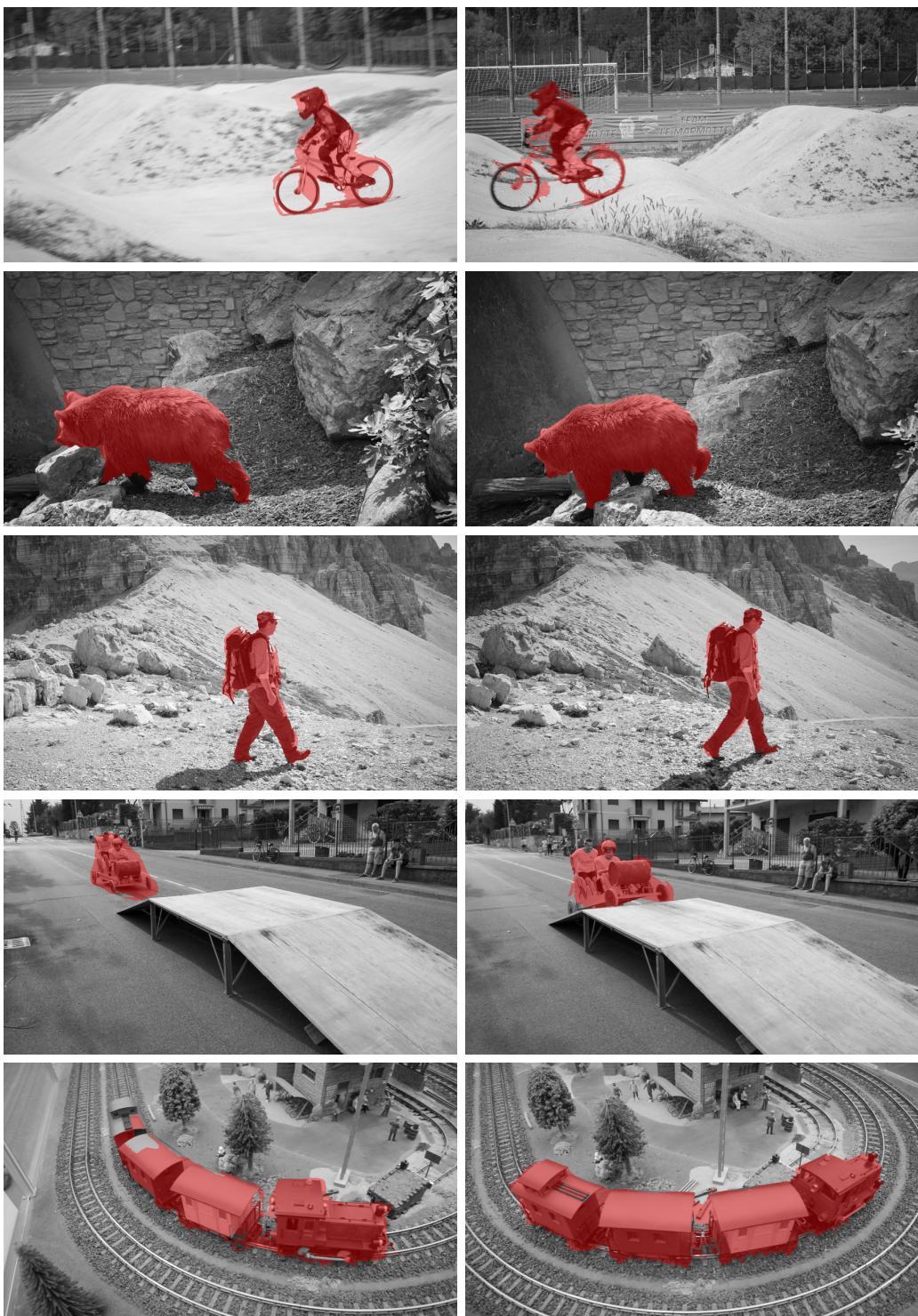
## 4.5 Clasificación de regiones

Ya disponiendo de los votos finales de todas las regiones del vídeo, llegar a una segmentación de este únicamente requiere la definición de un criterio de clasificación de votos. Un voto nulo se asocia con el fondo, mientras que valores más elevados favorecen a la categoría de objeto. Por ello, la elección de un umbral de separación entre ambos induce a una segmentación de las regiones. Sea  $a > 0$  un punto de corte y  $v(R) \in [0, 1]$  el voto final de la región  $R$ , los conjuntos  $\mathcal{F}_a = \{R \in \mathcal{R} \mid v(R) < a\}$  y  $\mathcal{O}_a = \{R \in \mathcal{R} \mid v(R) \geq a\}$  son una segmentación de  $\mathcal{R}$ . Como los elementos de  $\mathcal{R}$  están formados por agrupaciones de píxeles que cubren la totalidad de las imágenes,  $\mathcal{F}_a$  y  $\mathcal{O}_a$  constituyen una segmentación densa del vídeo, siendo respectivamente su fondo y sus objetos.

La no localidad de NLCV provoca que la segmentación que resulta de su aplicación en ocasiones divida los objetos en múltiples componentes conexas. Es posible que esta decisión sea correcta: un objeto puede ser oculto parcialmente en un fotograma y separarse visualmente en varios fragmentos, así como puede haber más de un objeto en el vídeo. No obstante, eventualmente superpíxeles inconexos del fondo de tamaño pequeño son clasificados incorrectamente como objetos por tener un aspecto y textura similar a estos últimos. Como solución a este problema, es recomendable postprocesar los resultados eliminando componentes conexas de área menor a un porcentaje pequeño (por ejemplo, 5%) de la componente más grande de la segmentación. De esta manera, solamente se corrigen pequeños errores y se mantienen intactos los objetos suficientemente grandes, presumiblemente bien segmentados.

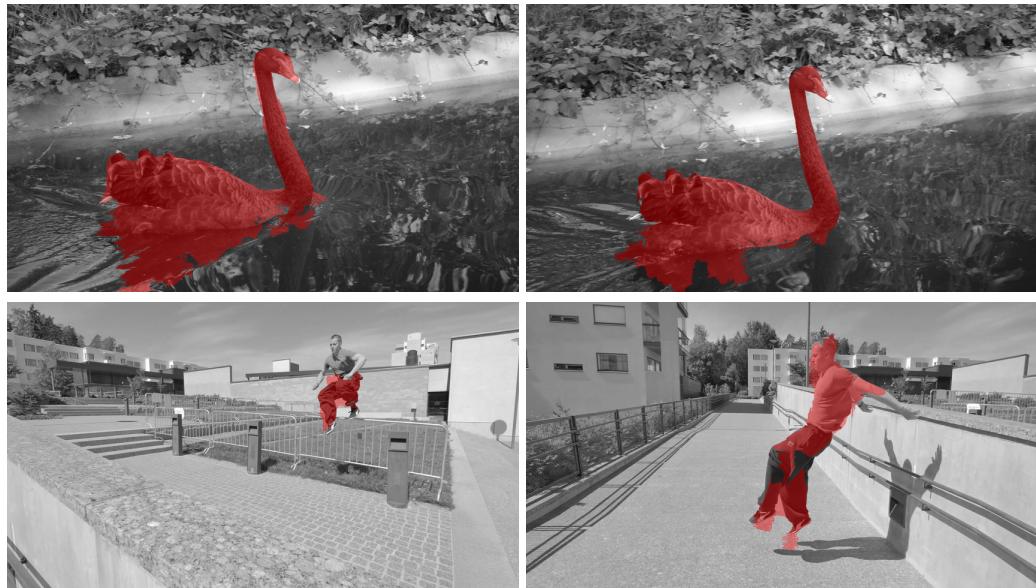
Una implementación propia de NLCV en C++ se facilita junto a este trabajo. Los superpíxeles y sus descriptores (histogramas de colores y HOGs) se obtienen con la implementación de los correspondientes algoritmos en la biblioteca OpenCV [6]. Esta última también proporciona un algoritmo eficiente de búsqueda de vecinos más cercanos, que se utiliza para construir el grafo correspondiente. La biblioteca Eigen [16] también es utilizada para operar con matrices dispersas de grandes dimensiones en el paso de propagación de votos. La Figura 4.3 muestra ejemplos de los resultados que se obtienen con ella sobre secuencias de DAVIS [39]; todas las segmentaciones han sido alcanzadas con quinientas iteraciones de (4.10), un valor de  $\sigma^2 = 0.05$  en (4.8), y eliminación de regiones pequeñas inconexas. Los valores de corte  $a$  han sido elegidos de forma diferente en cada secuencia, situándose entre 0.15 y 0.4.

La calidad de los resultados de NLCV depende de que las estimaciones de los campos de flujo óptico sean fiables. La consecuencia negativa principal de una mala



**Figura 4.3:** Segmentación densa resultante del algoritmo NLCV aplicado sobre secuencias de DAVIS. Los objetos se identifican con un color rojo transparente. De arriba a abajo: BMX-BUMPS, BEAR, HIKE, SOAPBOX, TRAIN.

#### 4. SEGMENTACIÓN DE REGIONES DE PÍXELES



**Figura 4.4:** Segmentación densa con el algoritmo NLCV de secuencias con movimiento complejo de la base de datos DAVIS. Primera fila: fotogramas de BLACKSWAN. Segunda fila: fotogramas de PARKOUR.

estimación del movimiento es una inicialización inexacta de los votos, cuya propagación puede causar errores en todo el vídeo. Por este mismo motivo, objetos con desplazamientos rápidos (como BMX-BUMPS en la Figura 4.3) tienden a segmentarse deficientemente alrededor de sus bordes, mientras que aquellos más lentos (cuyo flujo óptico es más preciso, como BEAR o HIKE) se ven delimitados perfectamente. Secuencias con una gran cantidad de fotogramas sin movimiento dominante, como TRAIN, ven corregidos correctamente la mayoría de sus votos iniciales, pero muestran un mayor número de errores en la clasificación de regiones que pertenecen a objetos.

Vídeos que exhiben movimientos de fondo y de cámara complejos como los de la Figura 4.4 presentan dificultades para ser segmentados correctamente, pero aun así dan mejores resultados que otros métodos no supervisados como el de la Sección 3.1. En la secuencia BLACKSWAN el cisne se identifica adecuadamente, pero su reflejo en el agua se clasifica erróneamente como objeto, ya que presenta movimiento y apariencia similares al ave. Por otro lado, PARKOUR presenta una inicialización ruidosa de los votos de las regiones situadas sobre el hombre, y los movimientos de la cámara dificultan encontrar zonas que se repiten entre fotogramas, lo que hace que la segmentación sea inexacta.

NLCV funciona bajo suposiciones mínimas sobre los vídeos a los que se aplica, lo que hace de él una herramienta de segmentación versátil. Como algoritmo, utiliza técnicas de implementación sencilla y es computacionalmente eficiente, siendo capaz de llegar producir resultados densos en unos pocos segundos por fotograma. Se trata, pues, de un método de segmentación capaz de actuar adecuadamente sobre multitud de vídeos de manera no supervisada incluso en circunstancias en las que otros métodos más especializados fracasan.

CAPÍTULO



# 5

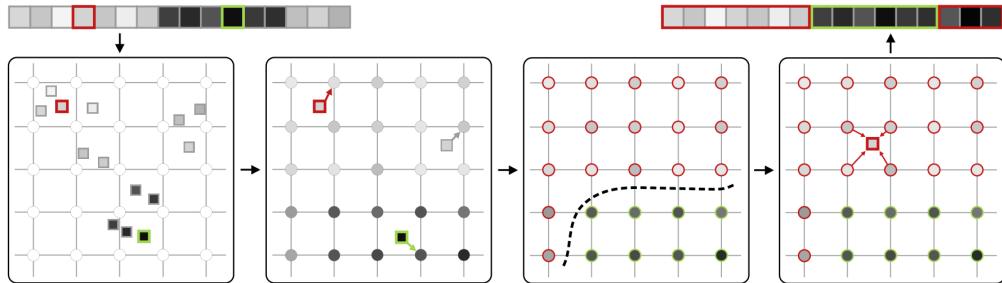
## DENSIFICACIÓN DE TRAYECTORIAS

Las estrategias de segmentación de vídeo vistas en los capítulos 3 y 4 se distinguen por la densidad de los segmentos que separan con respecto al total de puntos de las imágenes. Por un lado, la segmentación de trayectorias clasifica únicamente un conjunto disperso de puntos de interés de los que se dispone de un seguimiento temporal. Por el otro, la segmentación de regiones etiqueta cada uno de los píxeles de una secuencia de imágenes acorde a la entidad a la que pertenecen. Mientras que la primera es usualmente suficiente para el rastreo de objetos, en ocasiones es conveniente que estos se delimiten píxel a píxel en una segmentación densa como la segunda. En este capítulo se expone cómo utilizar la información que proporciona la segmentación de trayectorias para clasificar la totalidad de los píxeles de un vídeo, proceso que se conoce como *densificación*.

Un algoritmo de densificación utiliza información previamente conocida para segmentar una secuencia de imágenes, por lo que sus entradas son las mismas que las de un algoritmo de segmentación semisupervisado. En lugar de aceptar directamente interacción humana, no obstante, son las agrupaciones de trayectorias preliminarmente determinadas las que introducen un sesgo para clasificar el resto de puntos. En la misma línea, del mismo modo que una mayor supervisión contribuye a la exactitud de la segmentación interactiva, una buena cobertura de trayectorias en todas las entidades que se han de separar es imprescindible para que su densificación sea precisa. Además, como estas pueden no ser completamente fiables individualmente, interesa dar opción a que sus puntos puedan corregirse en función de sus alrededores en caso de estar inicialmente mal etiquetados, haciendo énfasis en contornear adecuadamente los bordes de los objetos.

El método de densificación que se presenta en este capítulo está basado en la extensión de los puntos de una secuencia de imágenes sobre su *espacio bilateral*, que complementa sus dominios espacial y temporal con su espacio de color [36]. En un vídeo digital, dicho espacio forma una estructura de rejilla multidimensional, cuya manipulación es eficiente y se traduce sobre el dominio original en un procesamiento que detecta y preserva los bordes de objetos. Entendida como un grafo no dirigido, par-

## 5. DENSIFICACIÓN DE TRAYECTORIAS



**Figura 5.1:** Ejemplo de segmentación de una señal de una dimensión y un canal utilizando su rejilla bilateral bidimensional. Primero, los puntos de la señal parcialmente anotada (izquierda) se levantan al espacio bilateral (primer cuadro), y se adjudican a la correspondiente celda de la rejilla muestreada (segundo cuadro). A continuación, se partitiona la rejilla vista como un grafo (tercer cuadro). Finalmente, el etiquetado de las celdas se transfiere a los puntos originales (cuarto cuadro), y se recupera la señal segmentada (derecha).

ticionar esta rejilla introduciendo previamente un sesgo en los nodos correspondientes a los puntos de las trayectorias ya segmentadas induce a una clasificación de todos los píxeles del vídeo. Así pues, la información parcial que aporta la segmentación de trayectorias se propaga evitando sobrepassar bordes, influenciando a la totalidad de los puntos y posiblemente corrigiendo inicializaciones incorrectas.

En la primera sección de este capítulo se explica de manera teórica el proceso de construcción, segmentación, y recuperación de la información en el espacio bilateral, visto como un algoritmo de segmentación denso semisupervisado. En la segunda, se describe cómo utilizar una segmentación de trayectorias determinada anticipadamente para prescindir de una interacción directa con el algoritmo, así como se muestran ejemplos de su utilización para este propósito y de los resultados que produce en la práctica en combinación con distintos métodos estudiados en el Capítulo 3.

### 5.1 Segmentación en el espacio bilateral

Sea  $\mathcal{V}: \Omega \times [1, T] \rightarrow \mathbb{R}^d$  un vídeo que ha de segmentarse en  $K$  distintas entidades (el fondo y  $K - 1$  objetos), donde  $\Omega \subset \mathbb{R}^2$  y  $d \in \mathbb{N}$  son su dominio espacial y el número de canales de su espacio de color, respectivamente. Su espacio bilateral es una estructura  $(d + 3)$ -dimensional, subconjunto del producto cartesiano de su dominio y su codominio, sobre la que los puntos del vídeo se inmergen de manera natural como una variedad geométrica [10]. En un formato digital, donde estos conjuntos son discretos, el espacio toma una configuración de rejilla, cuya resolución es manipulable submuestreando sus dimensiones; reducir esta última comporta que diferentes píxeles del vídeo se alcen sobre la misma celda, formando clases de equivalencia que imponen una regularización espacio-temporal y de apariencia en sus entornos. Atribuyendo a cada una de las celdas un vector  $(K + 1)$ -dimensional que recuenta los píxeles que albergan y la evidencia de que estas pertenezcan a los  $K$  segmentos, la segmentación de  $\mathcal{V}$  sobre la rejilla se realiza como el corte de un grafo. Esta segmentación es computacionalmente económica debido a que la rejilla mantiene la información de todos los píxeles incluso

a resoluciones bajas (pocos nodos en el grafo), y tiene la ventaja de mantener cercanos los píxeles de apariencia y posición similar, siendo capaz de delimitar precisamente los objetos a lo largo de sus bordes.

La segmentación en la rejilla bilateral puede dividirse en cuatro fases, ejemplificadas en la Figura 5.1 [36, Fig. 2]. Las dos primeras se explican en la Sección 5.1.1, y consisten en el levantamiento de los puntos de  $\mathcal{V}$  sobre su espacio bilateral y su posicionamiento en la rejilla muestrada. Después, con la rejilla ya construida, se utilizan estrategias de corte de grafos para particionarla y agrupar sus celdas en tantos conjuntos como entidades en el vídeo, lo que se ve en la Sección 5.1.2. En la Sección 5.1.3, finalmente, se recuperan los píxeles originales segmentados a partir de la clasificación de sus celdas en la rejilla.

### 5.1.1 Construcción de la rejilla bilateral

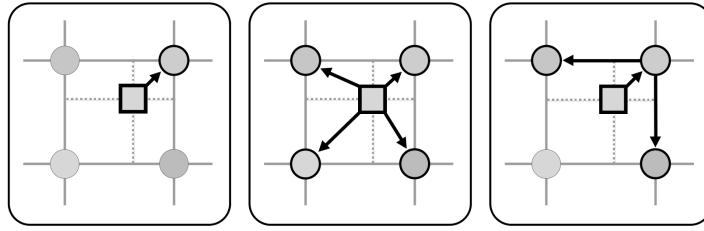
Un punto  $\mathbf{x} = (x, y) \in \Omega$  del  $t$ -ésimo fotograma  $I_t: \Omega \rightarrow \mathbb{R}^d$  de un vídeo se *levanta* a su espacio bilateral inyectándolo sobre  $\mathbb{R}^{d+3}$ , y escalando de manera separada cada una de sus coordenadas en este último. Sean  $I_t^1, \dots, I_t^d$  las componentes de  $I_t$ , y  $s_s, s_t, s_r^1, \dots, s_r^d$  unos parámetros de escala para cada dimensión, la aplicación

$$\begin{aligned} \ell: \Omega \times [1, T] &\rightarrow \mathbb{R}^{d+3} \\ (\mathbf{x}, t) &\mapsto \left( \frac{x}{s_s}, \frac{y}{s_s}, \frac{t-1}{s_t}, \frac{I_t^1(\mathbf{x})}{s_r^1}, \dots, \frac{I_t^d(\mathbf{x})}{s_r^d} \right) \end{aligned} \quad (5.1)$$

es un levantamiento hacia dicho espacio. En el caso de que la imagen esté en escala de grises, dados  $\mathbf{x}_1, \mathbf{x}_2 \in \Omega$  y  $t_1, t_2 \in [1, T]$ , la distancia euclídea entre  $\ell(\mathbf{x}_1, t_1)$  y  $\ell(\mathbf{x}_2, t_2)$  es una medida de la disimilitud entre los píxeles que valora su proximidad e intensidad. Del mismo modo, cuando  $d = 3$ , con ella se diferencian los puntos por su posición y por su color, con el matiz de que esta diferencia depende del rango de valores que toma el espacio de color de las imágenes. Utilizando un espacio de color perceptiblemente uniforme se logra que puntos percibidos como similares en apariencia y ubicación sean cercanos entre sí en el espacio bilateral; específicamente, [48] recomienda el uso de  $L^*u^*v^*$  (aunque  $L^*a^*b^*$  es igualmente válido) para representar el color y hacer que las distancias en el espacio bilateral sean significativas acorde a la visión humana.

El espacio bilateral de un vídeo digital necesariamente ha de ser un conjunto discreto y finito. Concretamente, los parámetros  $s_s, s_t, s_r^1, \dots, s_r^d$  de (5.1) son los períodos de muestreo que se definen en cada una de las dimensiones del espacio para convertirlo en una rejilla  $\Gamma$ . Períodos grandes disminuyen la cantidad de celdas de esta última, restringiendo las posiciones a las que se pueden levantar los puntos. El levantamiento (5.1), sin embargo, está definido para todos los puntos de un vídeo, que no necesariamente caen en la posición exacta de una celda de la rejilla bilateral. Así, si  $\ell(\mathbf{x}, t)$  no tiene las coordenadas de ningún  $\mathbf{v} \in \Gamma$ , la información que contiene ha de repartirse a sus celdas más cercanas según cierto criterio. Este proceso de distribución de la evidencia de pertenencia a una agrupación sobre la rejilla bilateral es denominado *splatting*.

Sea  $\hat{\mathbf{p}} \in [0, 1]^K$  un vector cuya  $k$ -ésima componente contiene un valor (no necesariamente una probabilidad) que indica el grado de certeza de que el píxel  $\mathbf{p} \in \Omega \times [1, T]$  de un vídeo forme parte de la entidad  $k$ . Para cada celda  $\mathbf{v} \in \Gamma$ , una suma pondera-



**Figura 5.2:** Métodos de interpolación en la fase de *splatting*: vecino más cercano (izquierda), multilineal (centro), y adyacente (derecha).

da  $S(\mathbf{v}) \in \mathbb{R}^{K+1}$  de los píxeles levantados se calcula como

$$S(\mathbf{v}) = \sum_{\mathbf{p}} w(\mathbf{v}, \ell(\mathbf{p})) \cdot (\hat{\mathbf{p}}, 1), \quad (5.2)$$

acumulando en los nodos de la rejilla la información de su pertenencia a las  $K$  entidades, así como la cantidad (o fracción) de píxeles que contribuyen a ella. La función de peso  $w$  determina el rango e influencia que cada píxel levantado  $\ell(\mathbf{p})$  tiene sobre los vértices de  $\Gamma$ ; dependiendo de cómo esta se defina, la suma (5.2) puede repartir el valor de un píxel en varias celdas o concentrarlo en unas pocas. La posibilidad más simple es definir  $w(\mathbf{v}, \ell(\mathbf{p})) = 1$  si  $\mathbf{v}$  es la celda más cercana a  $\ell(\mathbf{p})$  y 0 en otro caso, lo que suma los valores inalterados de sus píxeles vecinos. Por otro lado, de manera más compleja, repartir una fracción de los valores de  $\hat{\mathbf{p}}$  a todas las celdas de alrededor de  $\ell(\mathbf{p})$  por medio de interpolación multilineal es más preciso, pero el número de ellas crece exponencialmente con la dimensión del espacio bilateral. Como tercera opción, [36] prefiere el uso de interpolación adyacente, ejemplificada junto a las otras dos en la Figura 5.2, que ignora aquellas celdas cuyo peso sería insignificante en la interpolación multilineal y distribuye los valores utilizando el peso

$$w(\mathbf{v}, \ell(\mathbf{p})) = \begin{cases} \prod_{i=1}^{d+3} (1 - |v_i - \ell_i(\mathbf{p})|) & \text{si } \mathbf{v} \in \mathcal{A}_{\ell(\mathbf{p})}, \\ 0 & \text{en otro caso,} \end{cases} \quad (5.3)$$

donde  $\mathcal{A}_{\ell(\mathbf{p})}$  es el conjunto de vértices que a la vez rodean a  $\ell(\mathbf{p})$  y son adyacentes a la celda que más se le acerca (incluyéndola), diferiendo de esta última en una única dimensión en el sentido del píxel levantado. La cantidad de celdas a las que la interpolación adyacente atribuye los valores de un píxel crece linealmente con respecto a las dimensiones de la rejilla: cuando el espacio bilateral es  $\mathbb{R}^6$ , hay siete de ellas, frente a las sesenta y cuatro de la interpolación multilineal. Como el número no es mucho mayor que la única celda del criterio del vecino más cercano, su complejidad computacional es reducida sin una pérdida grande de calidad. Esto hace de (5.3) una función de peso idónea cuando la rejilla bilateral es extensa.

### 5.1.2 Partición de la rejilla

Independientemente del tipo de interpolación utilizada en la ecuación (5.2), los vértices  $\mathbf{v} \in \Gamma$  y su suma asociada  $S(\mathbf{v})$  contienen toda la información sobre los píxeles del vídeo original y la evidencia de que pertenezcan a cada una de las distintas entidades

que aparecen en él. Encontrar una segmentación de este último, por tanto, es posible por medio de un etiquetado de las celdas de  $\Gamma$  en  $K$  divisiones. Para ello, se da a la rejilla una estructura de grafo  $G = (V, E)$ , con  $V = \Gamma$ . Otorgando a los nodos un sesgo de pertenencia a cada partición según las componentes de  $S(\mathbf{v})$ , y conectando celdas adyacentes por medio de aristas con pesos dependientes de su distancia, basta con obtener un corte mínimo de  $G$  para separar la rejilla en agrupaciones correspondientes a cada entidad.

En las secciones 3.2.3 y 3.2.4 se ha visto que los paseos aleatorios son una herramienta eficaz para cortar grafos cuando se dispone de información previa sobre sus nodos. No obstante, el sistema (3.66) no es aplicable en un contexto de densificación, ya que presupone que existen nodos marcados inmutables ( $\mathbf{x}_M$  en la ecuación). Este hecho no ocurre en la rejilla bilateral: el contenido de las sumas  $S(\mathbf{v})$  para  $\mathbf{v} \in \Gamma$  puede incluir información mezclada de píxeles marcados y no marcados, y el etiquetado inicial debe ser mutable para poder corregirse. En su lugar, [36] formula el corte como un problema de minimización de una función de energía dependiente del etiquetado de los nodos.

Como  $G$  tiene una configuración de retícula, bajo la hipótesis de que su etiquetado es regular, la probabilidad de que un nodo se etiquete de una manera concreta depende únicamente de la evidencia que el vértice dispone para ella, y de la de sus vecinos inmediatos por los que se conecta con una arista. Por esta propiedad, si  $\boldsymbol{\alpha} = (\alpha_{\mathbf{v}})_{\mathbf{v} \in \Gamma}$  son las etiquetas de los vértices vistas como variables aleatorias, estas forman un Markov Random Field (MRF) de primer orden [31, pp. 32–35], caracterizado por poder descomponer su función de densidad de probabilidad como

$$\Pr(\boldsymbol{\alpha}) = \prod_{\mathbf{v} \in \Gamma} \varphi_{\mathbf{v}}(\alpha_{\mathbf{v}}) \prod_{(\mathbf{u}, \mathbf{v}) \in E} \varphi_{\mathbf{uv}}(\alpha_{\mathbf{u}}, \alpha_{\mathbf{v}}), \quad (5.4)$$

siendo  $\varphi_{\mathbf{v}}$  y  $\varphi_{\mathbf{uv}}$  otras funciones de probabilidad que actúan como sus factores. El maximizador de (5.4) precisamente conduce a la forma deseada de etiquetar los nodos, al ser la más probable *a posteriori* entre todas las posibles dada la información disponible sobre el grafo. Definiendo

$$\lambda_u \theta_{\mathbf{v}}(\alpha_{\mathbf{v}}) = -\ln \varphi_{\mathbf{v}}(\alpha_{\mathbf{v}}) \quad \forall \mathbf{v} \in \Gamma, \quad (5.5)$$

y  $\lambda_s \theta_{\mathbf{uv}}$  de la misma manera con  $\varphi_{\mathbf{uv}}$  para todo  $(\mathbf{u}, \mathbf{v}) \in E$ , donde  $\lambda_u > 0$  y  $\lambda_s > 0$  son parámetros de escala para los términos que dependen respectivamente de los nodos (unarios) y de las aristas (de suavizado), la maximización de (5.4) equivale a la minimización de su logaritmo negado

$$E(\boldsymbol{\alpha}) = \lambda_u \sum_{\mathbf{v} \in \Gamma} \theta_{\mathbf{v}}(\alpha_{\mathbf{v}}) + \lambda_s \sum_{(\mathbf{u}, \mathbf{v}) \in E} \theta_{\mathbf{uv}}(\alpha_{\mathbf{u}}, \alpha_{\mathbf{v}}). \quad (5.6)$$

Los términos unarios modelan las desviaciones respecto al etiquetado preliminar: la función  $\theta_{\mathbf{v}}$  penaliza en mayor medida asignar a  $\mathbf{v} \in \Gamma$  una etiqueta marcada inicialmente como poco probable según el contenido de  $S(\mathbf{v})$ . Por ejemplo, en el caso de buscar una segmentación binaria de fondo y objetos, si  $S_F(\mathbf{v})$  es la componente de  $S(\mathbf{v})$  correspondiente a la evidencia del fondo y  $S_O(\mathbf{v})$  a la de los objetos, la función definida como

$$\theta_{\mathbf{v}}(\alpha_{\mathbf{v}}) = \begin{cases} S_O(\mathbf{v}) & \text{si } \alpha_{\mathbf{v}} \text{ es fondo,} \\ S_F(\mathbf{v}) & \text{si } \alpha_{\mathbf{v}} \text{ es objeto,} \end{cases} \quad (5.7)$$

## 5. DENSIFICACIÓN DE TRAYECTORIAS

---

cumple esa función. Los términos de suavizado, por su parte, tratan de favorecer que vértices vecinos tengan la misma etiqueta, estableciendo penalizaciones en caso contrario. Esta penalización depende del grado de afinidad entre los nodos, dada por los pesos de las aristas. Además, como las celdas de la rejilla reúnen cantidades distintas de píxeles, es necesario escalar la afinidad por dichos números para que asignar etiquetas diferentes a celdas vecinas sea aproximadamente equivalente a hacerlo sobre todos los píxeles que contienen [36]. Así pues, si  $S_M(\mathbf{v})$  es la última componente de  $S(\mathbf{v})$ , que recuenta las fracciones de píxeles que alberga  $\mathbf{v} \in \Gamma$ , la forma general de  $\theta_{\mathbf{u}\mathbf{v}}$  es

$$\theta_{\mathbf{u}\mathbf{v}}(\alpha_{\mathbf{u}}, \alpha_{\mathbf{v}}) = \begin{cases} S_M(\mathbf{u}) \cdot S_M(\mathbf{v}) \cdot g(\mathbf{u}, \mathbf{v}) & \text{si } \alpha_{\mathbf{u}} \neq \alpha_{\mathbf{v}}, \\ 0 & \text{en otro caso,} \end{cases} \quad (5.8)$$

donde  $g(\mathbf{u}, \mathbf{v})$  expresa la semejanza entre los dos vértices (usualmente a partir de su distancia). Así, la formulación de la energía (5.6) con las funciones (5.7) y (5.8) permite que los vértices puedan etiquetarse de una manera distinta a la que se inclinan inicialmente si sus vecinos no concuerdan con ella.

La energía (5.6) generalmente es difícil de minimizar, ya que no es necesariamente convexa y la cantidad de combinaciones de etiquetas suele ser desmesurada. Más precisamente, si  $\boldsymbol{\alpha}$  especifica más de dos etiquetas, su minimización ha sido demostrada como NP-compleja [4], y los algoritmos existentes que permiten optimizarla no garantizan encontrar un mínimo global cuando los términos de suavizado preservan discontinuidades, como ocurre con (5.8). Sin embargo, en el caso de buscar una segmentación binaria, el problema de su minimización equivale al corte mínimo de  $G$  con dos nodos ficticios añadidos (una *fuente* y un *sumidero*, pertenecientes respectivamente a los objetos y al fondo), cuyo óptimo global es alcanzable en tiempo polinómico en función del número de nodos obteniendo el *flujo* máximo que mana entre ellos [28]. El método de Boykov–Kolmogorov [5], basado en el acrecentamiento de caminos entre fuente y sumidero, ha demostrado ser eficiente para este propósito. Encontrar una segmentación que separa fondo y objetos, por tanto, es factible mediante el uso de este último algoritmo, mientras que para obtener una partición en más divisiones han de utilizarse métodos más complejos de corte múltiple [4, 11]. Alternativamente, aplicar Boykov–Kolmogorov reiteradamente para sobresegmentar los dos subgrafos que produce hasta llegar al número de particiones deseado es un método simple (aunque inexacto) de etiquetado múltiple.

### 5.1.3 Recuperación de las imágenes segmentadas

Una vez han sido etiquetadas las celdas de la rejilla bilateral, traducirlas de vuelta al espacio de imagen lleva a una clasificación completa de sus píxeles. Con ese fin, para cada píxel se identifica la totalidad de las celdas de la rejilla cercanas a su levantamiento, determinando por su distancia la influencia relativa que tienen sobre él. Conocida esta última, se efectúa una votación en la que tienen más peso los votos de las celdas más influyentes, que decide qué etiqueta asignar al píxel en cuestión. Este proceso, inverso del *splatting*, recibe el nombre de *slicing*.

Sea  $\hat{\boldsymbol{\alpha}} = (\hat{\alpha}_{\mathbf{v}})_{\mathbf{v} \in \Gamma}$  el minimizador de (5.6), que confiere a cada  $\mathbf{v} \in \Gamma$  su etiqueta óptima expresada como un número natural  $k \in \{1, \dots, K\}$ . Para cada píxel  $\mathbf{p} \in \Omega \times [1, T]$ ,

## 5.2. Sesgo por trayectorias en el espacio bilateral

---

la evidencia de que pertenezca a la agrupación  $k$  se recupera mediante la fórmula

$$\mathcal{M}_k(\mathbf{p}) = \sum_{\mathbf{v} \in \Gamma} w(\mathbf{v}, \ell(\mathbf{p})) \llbracket \hat{\alpha}_{\mathbf{v}} = k \rrbracket, \quad (5.9)$$

donde la notación  $\llbracket P \rrbracket$  se utiliza para denotar el valor 1 si la proposición  $P$  es cierta, y 0 en otro caso. La función  $w$ , al igual que en el *splatting*, evalúa la importancia de las celdas de la rejilla sobre el píxel levantado, dando un peso mayor a las que están más cerca de él. Su elección no necesariamente debe ser la misma que la utilizada en la construcción de la rejilla, aunque generalmente usar el mismo método de interpolación o uno que tiene en cuenta más celdas que él resulta en segmentaciones de mayor calidad [36]. En concreto, el uso tanto en el *splatting* como en el *slicing* de los pesos (5.3) de la interpolación adyacente es capaz de lograr resultados fieles a la realidad con un tiempo de ejecución reducido.

El conjunto de funciones  $\{\mathcal{M}_k: \Omega \times [1, T] \rightarrow [0, 1] \mid 1 \leq k \leq K\}$  definidas por (5.9) describe las probabilidades de que los puntos del video a segmentar formen parte de sus distintas entidades, cumpliendo que  $\mathcal{M}_1(\mathbf{p}) + \dots + \mathcal{M}_K(\mathbf{p}) = 1$  para todo píxel  $\mathbf{p}$ . Determinar a qué objeto (o fondo) pertenecen, por tanto, se reduce a escoger la etiqueta con la probabilidad más alta (o una de ellas, en caso de empate). Los conjuntos  $A_s = \{\mathbf{p} \in \Omega \times [1, T] \mid \mathcal{M}_s(\mathbf{p}) = \max_k \mathcal{M}_k(\mathbf{p})\}$  con  $s \in \{1, \dots, K\}$ , pues, son una segmentación del video. En particular, suponiendo que  $s = 1$  es la etiqueta del fondo y el resto son diferentes objetos, los conjuntos  $\mathcal{F} = A_1$  y  $\mathcal{O} = \bigcup_{s=2}^K A_s$  conforman una segmentación binaria que separa *background* y *foreground*. Como  $\mathcal{F} \cup \mathcal{O} = \Omega \times [1, T]$ , la segmentación resultante es densa, y el método cumple el objetivo de ser capaz de densificar información parcial que aportan píxeles dispersos.

El *slicing* de la rejilla favorece que se delimiten de manera precisa los bordes de objetos, ya que los píxeles que los sobrepasan (que, a pesar de ser cercanos en el espacio de imagen, pertenecen a otra agrupación) se levantan sobre puntos del espacio bilateral lejanos entre sí. Al establecerse condiciones de regularidad para la partición de la rejilla, las celdas a las que se adjudican los puntos de ambos lados del borde reciben etiquetas diferentes, lo que se traduce en su separación en el espacio de imagen. Sin embargo, aunque la solución es regular en el espacio bilateral, esta propiedad no se traslada a la segmentación final del video, y es habitual que en ella se formen artefactos pequeños en píxeles aislados, causando errores en su clasificación. Postprocesar el video aplicando a cada uno de sus fotogramas un filtro que asigne a cada píxel la etiqueta más común en un entorno de  $3 \times 3$  ayuda a enmendar dicho problema, usualmente eliminándolo por completo.

## 5.2 Sesgo por trayectorias en el espacio bilateral

El algoritmo de segmentación en el espacio bilateral detallado en la Sección 5.1 puede ser utilizado siempre y cuando se le proporcione información inicial sobre la clasificación de algunos de los píxeles de la secuencia de imágenes a segmentar. Es por este motivo que generalmente es empleado de manera interactiva para completar un etiquetado parcial procurado por un usuario, o para propagar una segmentación manual detallada de un único fotograma a lo largo del tiempo. No obstante, no es estrictamente necesario que dicha información proceda directamente de una persona, sino que también puede hacerlo de otro algoritmo. Bajo este planteamiento, la segmentación de

## 5. DENSIFICACIÓN DE TRAYECTORIAS

---

trayectorias resultante de los métodos del Capítulo 3 es adecuada para usarse como el sesgo de clasificación inicial que el algoritmo requiere.

En los métodos de segmentación semisupervisados, la interacción por parte de los usuarios es de carácter absoluto, y por tanto las semillas que estos facilitan se suponen inequívocas. Esta suposición es inválida cuando las semillas originan de la salida de otro algoritmo no supervisado: los resultados de este último pueden contener errores, que no deben proliferar en el proceso de densificación, y el grado de certeza de que un píxel no los tenga puede variar con su posición y fotograma. Así pues, al densificar trayectorias, es preferible basarse en la evidencia (cuantitativa) que se ha utilizado previamente para segmentarlas que en la propia clasificación resultante. En el caso de hacerlo con las obtenidas por el método de modelado de fondo de la Sección 3.1, los pesos de las trayectorias proporcionan esta evidencia. Asimismo, si se obtienen por el paseo aleatorio de semillas de la Sección 3.2, las probabilidades de alcanzar cada agrupación cumplen esa misma función. Específicamente, cuando se busca exclusivamente una segmentación binaria, Wehrwein y Szeliski [48] proponen representar la evidencia asociada a cada píxel  $\mathbf{p} \in \Omega \times [1, T]$  como un vector  $\hat{\mathbf{p}} = (\hat{p}_F, \hat{p}_O) \in [0, 1]^2$  tal que

$$\hat{p}_F = 2 \max(0, w - 0.5), \quad (5.10)$$

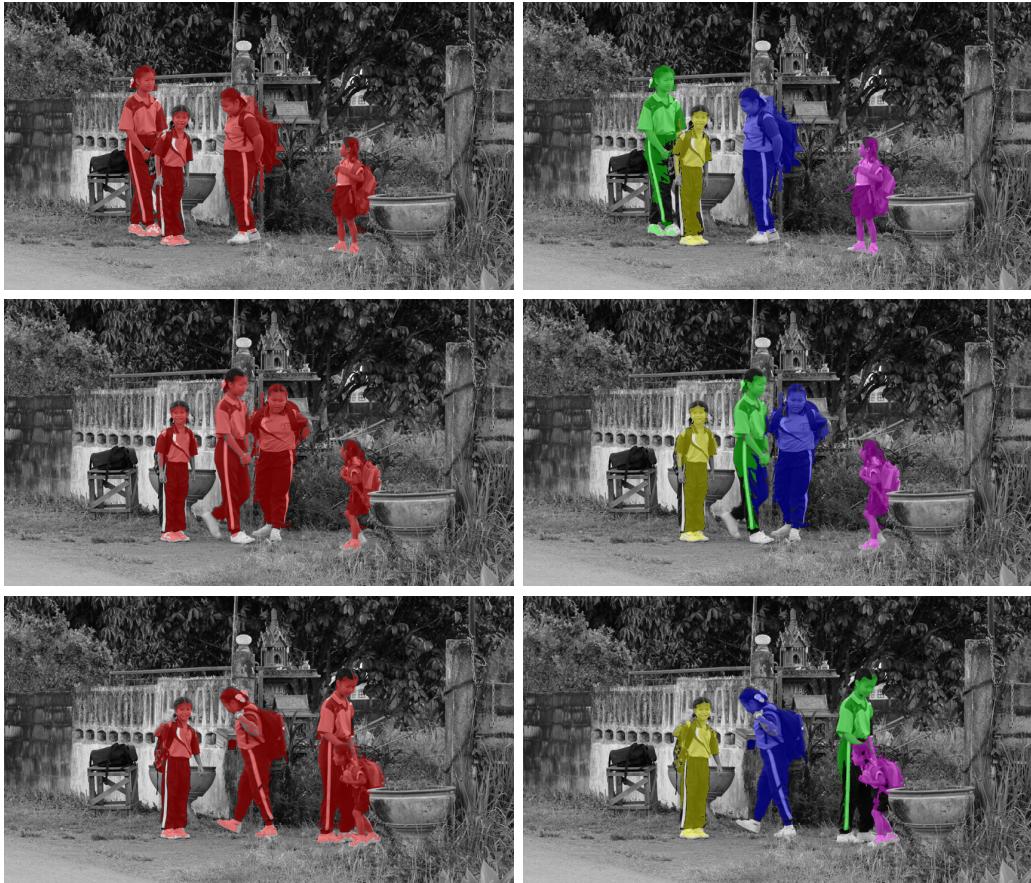
$$\hat{p}_O = 2 \max(0, 0.5 - w), \quad (5.11)$$

donde  $w \in [0, 1]$  es el peso (a favor del fondo) de la trayectoria que pasa por  $\mathbf{p}$ , y  $\hat{p}_F$  y  $\hat{p}_O$  son respectivamente la evidencia del fondo y de los objetos. Para generalizar esta definición a una segmentación de  $K$  entidades, es suficiente definir  $\hat{\mathbf{p}} \in [0, 1]^K$  como un vector de ceros excepto en la componente correspondiente a la entidad de probabilidad más alta, cuyo valor se obtiene por medio de (5.10), sustituyendo  $w$  por dicha probabilidad. Los valores recogidos en  $\hat{\mathbf{p}}$  son los que se reparten por medio de la fórmula (5.2) en la fase de *splatting* de la densificación.

A pesar de que el conjunto de trayectorias no cubre la totalidad de los puntos de un vídeo, todos los píxeles se levantan a su espacio bilateral aunque no contengan evidencia de ninguna agrupación. Aquellos por los que no pasa una trayectoria simplemente distribuyen un vector nulo durante el *splatting*, además de una unidad de *masa* que indica su presencia en las celdas en las que colaboran. Aun así, la falta de cobertura de trayectorias en algunas zonas (especialmente aquellas sin textura) puede impedir la recolección de evidencia en ellas, y dificultar su clasificación. Determinar la entidad a la que estas zonas pertenecen, pues, requiere de hacer suposiciones sobre el tipo de textura del fondo y de los objetos. De manera general, los objetos se caracterizan por tener un mayor interés visual, lo que usualmente comporta una abundancia de zonas con mucha textura; es por ello que es más verosímil que un área vasta sin textura sea parte del fondo. Bajo esta hipótesis, introducir un pequeño sesgo  $\hat{p}_F = 0.05$  a favor del fondo en píxeles que no tienen trayectorias vecinas en un determinado radio (por ejemplo, de 32 píxeles) ayuda a evitar la ambigüedad que esta situación provoca.

La rejilla bilateral creada a partir de la evidencia que aportan las trayectorias, vista como grafo, únicamente requiere que se tengan en consideración las celdas con una masa mayor que 0. De esta manera, a pesar de que el grafo se construye sobre un espacio de hasta seis dimensiones, su número de nodos es tratable computacionalmente. La partición de dicho grafo se consigue mediante la minimización de (5.6); específicamente, en la implementación del algoritmo presentada en este trabajo, se

## 5.2. Sesgo por trayectorias en el espacio bilateral



**Figura 5.3:** Densificación de la secuencia SCHOOLGIRLS de DAVIS. Izquierda: segmentación binaria de fondo (gris) y objetos (rojo) de los fotogramas 1, 31, y 61 de la secuencia. Derecha: segmentación de los mismos fotogramas distinguiendo los objetos entre sí.

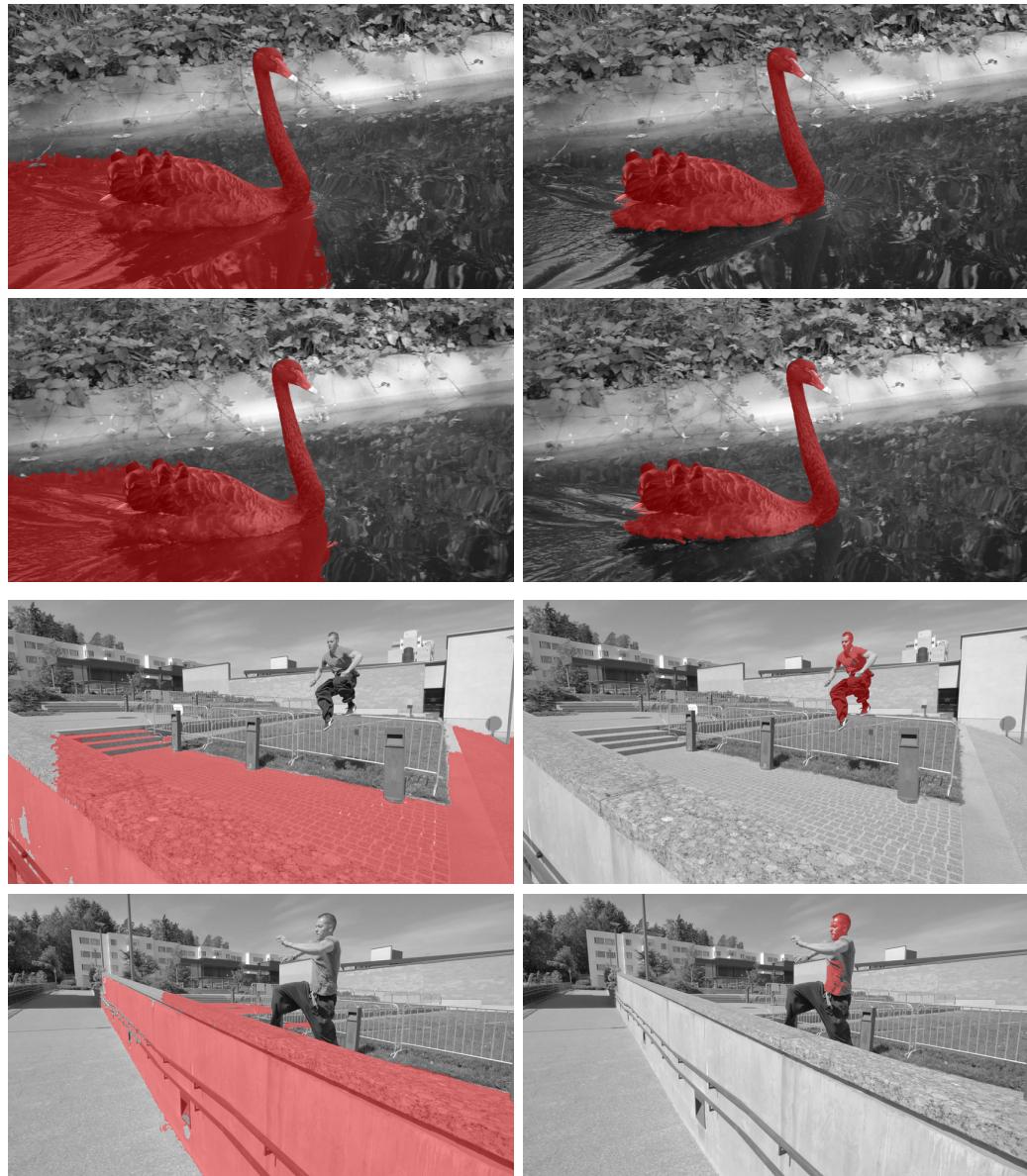
definen los términos de dicha energía según (5.7) y (5.8), utilizando

$$g(\mathbf{u}, \mathbf{v}) = e^{-\frac{1}{2} \|\mathbf{W}(\mathbf{u}-\mathbf{v})\|_2} \quad (5.12)$$

para transformar las distancias entre vértices en afinidades. La matriz diagonal  $\mathbf{W}$  escala cada dimensión del espacio bilateral, balanceando la influencia de la proximidad en espacio, tiempo, y color al diferenciar celdas en él. El uso de (5.7), exclusivo para segmentaciones binarias, se explica en el método de partición del grafo: en vídeos con varios objetos, en lugar de realizar el etiquetado por medio de algoritmos de corte múltiple, se opta por separar los objetos repetidamente uno a uno (contando el resto de imagen como fondo) a través de Boykov–Kolmogorov [5], lo que permite que el corte se efectúe en un tiempo polinomial. Esta última estrategia tiene el inconveniente de reducir su precisión con el número de objetos, y de otorgar ocasionalmente varias etiquetas a un mismo nodo, lo que ha de corregirse escogiendo una de ellas; aun así, los resultados que se recuperan después del *slicing*, ejemplificados en la Figura 5.3 (donde se utilizan las trayectorias de la Figura 3.4), son generalmente aceptables.

La calidad de la densificación final de trayectorias puede variar con el algoritmo utilizado para clasificarlas en primer lugar. Aunque la partición en el espacio bilateral

## 5. DENSIFICACIÓN DE TRAYECTORIAS



**Figura 5.4:** Segmentación de los fotogramas 1 y 21 de las secuencias BLACKSWAN (dos primeras filas) y PARKOUR (dos últimas filas) de DAVIS densificando las trayectorias de dos algoritmos distintos. Izquierda: densificación de las trayectorias de la Figura 3.2. Derecha: densificación de las trayectorias de la Figura 3.6.

asiste en corregir pequeños errores iniciales, estos pueden tener un impacto negativo importante si son demasiado extensos. La Figura 5.4 compara los resultados de densificar las trayectorias de BLACKSWAN y PARKOUR obtenidas con los métodos de las secciones 3.1 y 3.2 (figuras 3.2 y 3.6), secuencias que han demostrado ser problemáticas por su movimiento no rígido. La densificación de las trayectorias no supervisadas (izquierda) es deficiente, no distinguiendo agua de cisne en BLACKSWAN y descuidando completamente al hombre en PARKOUR. Por contra, la de aquellas adquiridas por el método interactivo (derecha) sí es capaz de llegar a resultados más ajustados a la reali-

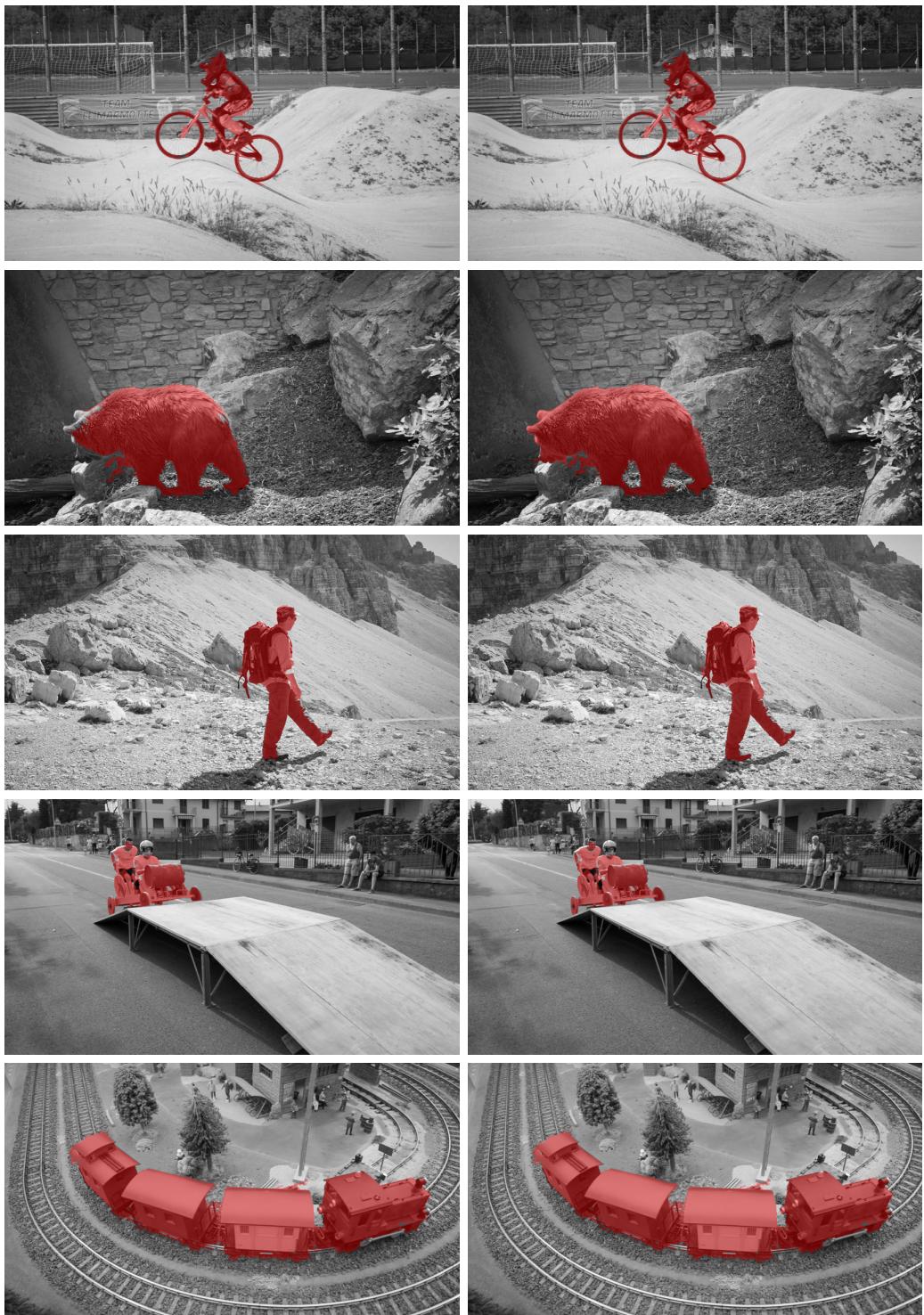
## 5.2. Sesgo por trayectorias en el espacio bilateral

---

dad, manifestándose su mayor limitación en la falta de cobertura de trayectorias sobre el hombre en PARKOUR. En vídeos más sencillos, donde los errores en las trayectorias no son tan pronunciados, el proceso de densificación tiende a enmendarlos acertadamente y dar resultados similarmente correctos para ambos algoritmos. En la Figura 5.5 se muestran ejemplos de este hecho, donde las diferencias entre métodos son poco perceptibles (notándose más en BEAR) a pesar de que las segmentaciones de trayectorias originales sean visiblemente distintas. De esta son destacables las rectificaciones en BMX-BUMPS y SOAPBOX respecto de la segmentación inicial no supervisada de la Figura 3.1, así como la forma en la que se densifican perfectamente las cubiertas de los vagones en TRAIN a pesar de no disponer de trayectorias sobre ellas. Finalmente, comparando estos resultados con los expuestos en la Figura 4.3, se constata que la delimitación de los bordes de objetos es más precisa que en el algoritmo NLCV, especialmente en vídeos con movimientos rápidos como BMX-BUMPS. La dependencia en unas trayectorias fiables, no obstante, limita el tipo de secuencias sobre las que la densificación es aplicable, haciendo de NLCV un método de segmentación densa más versátil.

Las imágenes de las figuras 5.3, 5.4, y 5.5 resultan de la ejecución de una implementación propia en C++ del algoritmo de segmentación semisupervisado de [36], adaptado al cometido de densificación acorde a [48]. Se han utilizado los parámetros  $(s_s, s_t, s_r^1, s_r^2, s_r^3) = (35, 10, 7.3, 8.5, 8.5)$  como escalas en el levantamiento (5.1),  $\lambda_u = 100$  y  $\lambda_s = 0.001$  en la energía (5.6), y  $W = \text{diag}(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{13}{10}, \frac{3}{2}, \frac{3}{2})$  en la aplicación (5.12), considerando zonas sin textura aquellas sin trayectorias vecinas en un radio de 32 píxeles y atribuyendo un sesgo a favor del fondo a sus píxeles con un espaciado de 8 de ellos en horizontal y vertical. El corte del grafo se ha realizado por medio de la biblioteca `maxflow v3.04` [5], una implementación del algoritmo de Boykov–Kolmogorov proporcionada públicamente por sus autores, mientras que la manipulación de imágenes se ha efectuado con las herramientas de OpenCV [6].

## 5. DENSIFICACIÓN DE TRAYECTORIAS



**Figura 5.5:** Segmentación resultante de la densificación de trayectorias en varias secuencias de DAVIS. De arriba a abajo: BMX-BUMPS, BEAR, HIKE, SOAPBOX, TRAIN. A la izquierda, el vigesimoprimer fotograma de las secuencias, utilizando las trayectorias de la Figura 3.1. A la derecha, el mismo fotograma usando las trayectorias de la Figura 3.5.

CAPÍTULO



## EVALUACIÓN DE LOS ALGORITMOS

Los métodos de segmentación de objetos en vídeo que se han explorado en los capítulos anteriores utilizan estrategias distintas, que comportan resultados más o menos precisos según las características de las secuencias de imágenes sobre las que se aplican. Dada la naturaleza visual de las imágenes, para determinar si un algoritmo segmenta bien un objeto, es suficiente analizar su salida de manera cualitativa: si se ajusta a la región donde se encuentra el objeto real, la segmentación es correcta; si no, esta presenta errores. Del mismo modo, una comparación cualitativa es usualmente la más adecuada para contrastar la calidad de varios algoritmos aplicados a un vídeo concreto, ya que de ellos lo más importante es la apariencia de sus soluciones. Sin embargo, para estudiar rigurosamente las características generales de los diferentes algoritmos, se vuelve necesario definir medidas que valoren numéricamente el ajuste de sus resultados a la realidad. Junto con una base de datos de referencia como DAVIS [39], un análisis cuantitativo a partir de ellas permite examinar fácilmente el comportamiento de los métodos en distintos tipos de vídeos, dando sobre ellos una visión global y menos sesgada por la visión humana.

Este capítulo tiene por objetivo establecer una serie de criterios para determinar de manera cuantitativa la calidad de métodos de segmentación densa de objetos. Con ellos se espera caracterizar las propiedades de los algoritmos de los capítulos 4 y 5, así como identificar los vídeos sobre los que dan resultados más fiables y precisos, en relación a las clasificaciones reales conocidas de todos sus píxeles. No se pretende, pues, evaluar métodos de segmentación de trayectorias sin densificar, cuyo propósito principal es el seguimiento de puntos de interés que se encuentran sobre los objetos de un vídeo, y no la delimitación exacta de estos últimos.

En la Sección 6.1 se proponen tres medidas para la evaluación de segmentaciones en secuencias de imágenes, en las que se tienen en cuenta características como la precisión en la clasificación de píxeles y en la delimitación de bordes, o la coherencia de la clasificación de objetos a lo largo del tiempo. Finalmente, en la Sección 6.2, se utilizan dichas medidas sobre los algoritmos de los capítulos anteriores: se aplican en secuencias para las que se han dado ejemplos de resultados cualitativos, se discuten

## 6. EVALUACIÓN DE LOS ALGORITMOS

---

las puntuaciones que estas reciben para cada método, y se comparan los algoritmos según los atributos de las secuencias en las que muestran mejores y peores resultados.

### 6.1 Medidas

Juzgar la calidad de una segmentación debe hacerse considerándola en el contexto del propósito final de la aplicación que la requiere. Cuando la segmentación de vídeo se utiliza principalmente para identificar y aislar objetos de interés como parte de un proceso más grande, como en la eliminación o sustitución del fondo en una videoconferencia, interesa minimizar la cantidad de píxeles mal etiquetados. Por otro lado, en aplicaciones de edición de vídeo, la precisión en la delimitación de contornos y su estabilidad temporal son propiedades más importantes, ya que es laborioso conseguirlas de manera manual. Asimismo, esta estabilidad temporal también es primordial para el seguimiento de objetos en cámaras de seguridad, que exige que las segmentaciones sean coherentes a lo largo del tiempo. Son estas tres medidas las que se describen en esta sección, estableciendo diferentes relaciones entre las salidas de un algoritmo y sus valores de *verdad terreno* (*ground truth*).

#### 6.1.1 Similitud de objetos

Sea una secuencia de imágenes  $\mathcal{V}: \Omega \times [1, T] \rightarrow \mathbb{R}^d$  que ha sido segmentada en  $K$  conjuntos disjuntos  $\mathcal{F}, \mathcal{O}_1, \dots, \mathcal{O}_{K-1}$  correspondientes al fondo y a los  $K - 1$  objetos que aparecen en el vídeo, cuya unión constituye todo  $\Omega \times [1, T]$ . Dada la segmentación real  $G_k$  del  $k$ -ésimo objeto (su verdad terreno), se define el ajuste de  $\mathcal{O}_k$  a  $G_k$  como su *índice de Jaccard* [41]

$$\mathfrak{J}_k = \frac{|\mathcal{O}_k \cap G_k|}{|\mathcal{O}_k \cup G_k|}, \quad (6.1)$$

que determina la proporción de píxeles del objeto que han sido bien clasificados respecto al tamaño del conjunto que resulta de agregarles los falsos positivos (píxeles etiquetados como objeto cuando no lo son) y los falsos negativos (píxeles no etiquetados como objeto cuando sí lo son).

La medida (6.1) tiene el beneficio de ser invariante a la escala de las imágenes, dando una puntuación entre 0 y 1 según la semejanza de la segmentación con la verdad terreno. Un valor reducido sugiere que hay muchos píxeles mal etiquetados, mientras que uno cercano a 1 se da cuando la segmentación del objeto presenta pocos errores. Esta medida de similitud, pues, sirve para evaluar individualmente el nivel de detalle por el que se distingue cada objeto del resto de los elementos de la escena. Obteniendo el índice de Jaccard para todas las entidades (excluyendo el fondo, de menor interés, que deriva inmediatamente del resto), es posible comparar segmentaciones por su exactitud en la clasificación de las regiones del interior de los objetos. En el caso  $K = 2$ , (6.1) es suficiente para describir la calidad de los resultados sobre el vídeo; si hay múltiples objetos en él, por contra, el cálculo de la media aritmética

$$\mathfrak{J} = \frac{1}{K-1} \sum_{i=1}^{K-1} \mathfrak{J}_k \quad (6.2)$$

es necesario para tenerlos en consideración [42].

### 6.1.2 Exactitud de contornos

Además de por medio de los conjuntos  $\mathcal{F}, \mathcal{O}_1, \dots, \mathcal{O}_{K-1}$ , la segmentación de un vídeo  $\mathcal{V}$  también puede expresarse por una serie de mapas de contornos cerrados que delimitan la extensión espacial de los objetos a lo largo del tiempo. Esta interpretación no aporta la misma información que la totalidad de los conjuntos densos (especialmente en el caso de que los objetos presenten huecos en su interior), pero es útil para conocer cómo se encuadra la segmentación a los objetos que separa.

Sean  $c(\mathcal{O}_k) : \Omega \times [1, T] \rightarrow \{0, 1\}$  vídeos binarios para  $k \in \{1, \dots, K\}$  que otorgan el valor 1 a los puntos que constituyen los bordes del objeto  $k$  detectados por la segmentación  $\mathcal{O}_k$ , y el valor 0 a los demás. Sean también  $c(G_k)$ , con  $k \in \{1, \dots, K\}$ , definidos igualmente para la segmentación real que viene dada por la verdad terreno  $G_k$ . El ajuste de  $c(\mathcal{O}_k)$  a  $c(G_k)$  se puede determinar observando el número de píxeles de contorno que coinciden en ambas imágenes (toman valor 1 en ellas) en relación a los que son dispares, dándoles un margen de error para compensar la estrechez de las fronteras. Dilatando los fotogramas de  $c(\mathcal{O}_k)$  y  $c(G_k)$  un porcentaje de su tamaño para incrementar el área que abarcan los contornos, y denotando por  $C_{\mathcal{O}}^k, \bar{C}_{\mathcal{O}}^k \subseteq \Omega \times [1, T]$  (respectivamente,  $C_G^k$  y  $\bar{C}_G^k$ ) a los subconjuntos de puntos de sus dominios que son parte de un borde en las imágenes dilatada (indicada por la barra) y sin dilatar, los cocientes

$$P_c^k = \frac{|C_{\mathcal{O}}^k \cap \bar{C}_G^k|}{|C_{\mathcal{O}}^k|} \quad \text{y} \quad R_c^k = \frac{|C_G^k \cap \bar{C}_{\mathcal{O}}^k|}{|C_G^k|} \quad (6.3)$$

son medidas de la precisión y de la sensibilidad de la delimitación del  $k$ -ésimo objeto [41]. En concreto,  $P_c^k$  mide la proporción de píxeles de la segmentación que se han clasificado correctamente, y  $R_c^k$  mide la relación de píxeles bien etiquetados entre el total que deberían haberlo sido acorde a la verdad terreno.

Tanto  $P_c^k$  como  $R_c^k$  ofrecen valores entre 0 y 1 que resultan deseables de maximizar, y conjuntamente son buenos indicadores de que un objeto está bien contorneado. Como ambas medidas son relevantes para determinar este hecho, con el objetivo de ponderar la distancia a una delimitación perfecta con  $P_c^k = R_c^k = 1$ , estas pueden combinarse en

$$\mathfrak{F}_k = \frac{2P_c^k R_c^k}{P_c^k + R_c^k}, \quad (6.4)$$

llamada la F-medida del contorno, que también toma valores entre 0 y 1. Haciendo su media aritmética para todos los objetos del vídeo al igual que en (6.2), el valor  $\mathfrak{F}$  que se obtiene evalúa la calidad de la segmentación completa.

### 6.1.3 Estabilidad temporal

La estabilidad temporal de una segmentación es un aspecto que describe cómo evoluciona la forma de los objetos identificados en un vídeo a medida que avanzan los fotogramas. Intuitivamente, un movimiento aceptable en un objeto es aquel que lo transforma de forma fluida, regular, y precisa entre imágenes, sin crear una sensación visual de fluctuación en la clasificación de sus bordes en el espacio. Consecuentemente, una medida de estabilidad debe reflejar esta idea penalizando perturbaciones no deseadas en los cambios de fotograma. Específicamente, en esta sección se proporciona una medida para la inestabilidad temporal, que crece con dichas perturbaciones.

## 6. EVALUACIÓN DE LOS ALGORITMOS

---

La segmentación en el  $t$ -ésimo fotograma de un vídeo  $\mathcal{V}: \Omega \times [1, T] \rightarrow \mathbb{R}^d$  puede distinguir una serie de polígonos que delimitan los objetos, cuyo número es igual o superior a la cantidad  $K$  de estos últimos. Si la secuencia es estable, estos polígonos tienen correspondencias en el siguiente fotograma, cuya forma se les asemeja. Formalmente, el parecido entre sus puntos se determina mediante un Shape Context Descriptor (SCD), especificado en un histograma que, dado un punto central, captura la distribución espacial absoluta de los puntos de su alrededor [39]. Sea  $h: \Omega \rightarrow \mathbb{N}^B$  la aplicación que, para cada punto, cuenta el número de píxeles cercanos que pertenecen a su mismo polígono, divididos en  $B$  contenedores distribuidos uniformemente en un círculo a su alrededor de forma log-polar. La disimilitud  $d: \Omega \times \Omega \rightarrow [0, +\infty)$  de la forma de dos puntos  $\mathbf{x}, \mathbf{y} \in \Omega$  se define como

$$d(\mathbf{x}, \mathbf{y}) = \sum_{b=1}^B \frac{(h_b(\mathbf{x}) - h_b(\mathbf{y}))^2}{h_b(\mathbf{x}) + h_b(\mathbf{y})}, \quad (6.5)$$

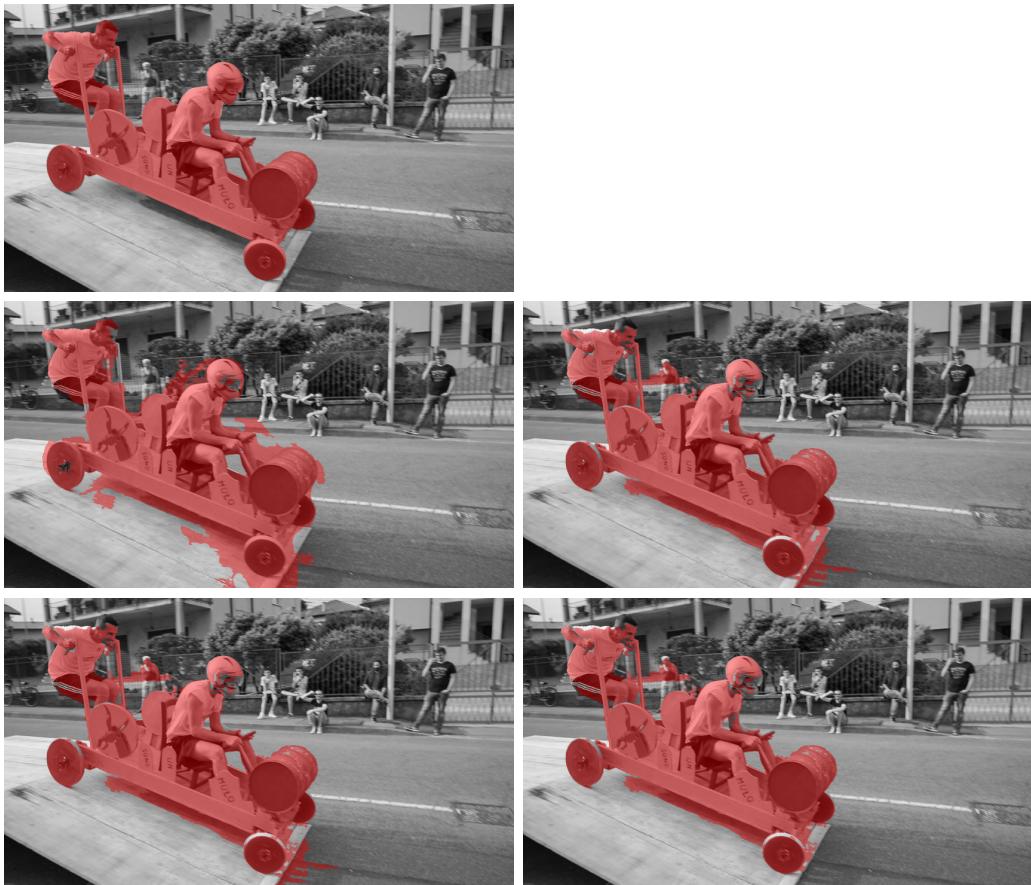
donde  $h_b(\mathbf{x})$  es la  $b$ -ésima componente de  $h(\mathbf{x})$ , e igualmente para  $h_b(\mathbf{y})$  [52].

Definida la aplicación (6.5), determinar el parecido de las formas de los polígonos entre dos fotogramas consecutivos puede plantearse como un problema de Dynamic Time Warping (DTW). Así, se busca emparejar los píxeles entre imágenes, de manera que todos ellos tengan una correspondencia y se minimice la suma de sus distancias a la vez que se preserva el orden de los puntos dentro de sus polígonos. La solución a este problema es fácilmente computable utilizando un algoritmo de programación dinámica [52], y las distancias óptimas entre correspondencias miden su fluctuación al avanzar de fotograma. Por tanto, la media  $\mathfrak{T}$  de estos valores de disimilitud para todo par de fotogramas sucesivos es una medida de la inestabilidad temporal en la totalidad del vídeo: un valor pequeño indica que las transformaciones de las segmentaciones de los objetos son generalmente suaves, mientras que uno grande advierte que estas muestran perturbaciones.

Ha de notarse que la medida  $\mathfrak{T}$ , al contrario que  $\mathfrak{J}$  y  $\mathfrak{F}$ , no está acotada superiormente y sus valores deseables son los próximos a cero. Además, es sensible a occlusiones, que pueden alterar los contornos de los objetos y hacer que se malinterpreten como transformaciones inestables. Es por ello que no existe un valor de referencia para determinar exactamente si una segmentación cualquiera es estable o no, y consecuentemente su uso se ve limitado a comparar únicamente segmentaciones de una misma secuencia.

## 6.2 Resultados

Las medidas presentadas en la Sección 6.1 ofrecen diferentes tipos de información sobre la calidad de segmentaciones que permiten evaluarlas cuantitativamente desde varios puntos de vista. La similitud de objetos  $\mathfrak{J}$  y la exactitud de contornos  $\mathfrak{F}$  están ligeramente correlacionadas, ya que una segmentación ajustada a un objeto suele delimitar bien sus contornos; aun así, las situaciones en las que sus valores pueden diferir son suficientemente numerosas como para que su uso conjunto sea significativo [39]. Por otro lado, tanto  $\mathfrak{J}$  como  $\mathfrak{F}$  son independientes de  $\mathfrak{T}$ , de manera que esta última las complementa adecuadamente. En esta sección, pues, se utilizan estas tres medidas para discutir y comparar los algoritmos de segmentación densa vistos en los capítulos anteriores. Específicamente, se usa  $\mathfrak{F}$  con una dilatación del 0.8 % del número de píxeles



**Figura 6.1:** Segmentación de un fotograma de SOAPBOX utilizando los algoritmos evaluados en esta sección. Primera fila: verdad terreno. Segunda fila: NLCV (izquierda), densificación de trayectorias no supervisada (derecha). Tercera fila: densificación interactiva anotando un fotograma (izquierda), anotando periódicamente (derecha).

de los polígonos en (6.3), y  $\mathfrak{T}$  toma  $B = 60$  en los SCDs de (6.5): cinco contenedores según el logaritmo del radio del círculo correspondiente, y doce según su ángulo.

Puesto que la calidad de la segmentación de un vídeo puede variar con cada fotograma, además de los valores medios de las medidas anteriores también interesan datos como la cantidad de fotogramas con una segmentación aceptable, y la progresión de la calidad en el tiempo. Con esto en consideración, dada una medida  $\mathfrak{M} \in \{\mathfrak{J}, \mathfrak{F}\}$ , se define su *sensibilidad (recall)* como la fracción de fotogramas que puntúa por encima de un umbral de calidad (por defecto, 0.5). Asimismo, su *deterioro (decay)* se conceptualiza como su pérdida de rendimiento con el paso del tiempo: si  $\{Q_1, \dots, Q_4\}$  es una partición de los fotogramas de un vídeo en cuartos ordenados cronológicamente, dicho deterioro se calcula como la resta  $\mathfrak{M}|_{Q_1} - \mathfrak{M}|_{Q_4}$ , donde la notación  $\mathfrak{M}|_{Q_1}$  restringe la medida  $\mathfrak{M}$  a la porción de vídeo  $Q_1$  (e igualmente para  $Q_4$ ). En  $\mathfrak{T}$  estos últimos datos carecen de significación, por lo que para ella únicamente se define su media en todo el vídeo.

Es natural que los métodos de segmentación interactivos muestren mejores resultados que los no supervisados, por lo que únicamente tiene sentido equiparar las segmentaciones de algoritmos que se encuentran en la misma categoría. Así, por un

## 6. EVALUACIÓN DE LOS ALGORITMOS

---

**Cuadro 6.1:** Evaluación de las segmentaciones obtenidas por NLCV. Las columnas **M**, **R**, y **D** contienen la media, sensibilidad, y deterioro de la correspondiente medida. Una flecha ↑ indica que un valor alto es mejor; una flecha ↓, lo contrario.

Secuencia	$\mathfrak{J}$			$\mathfrak{F}$			$\mathfrak{T}$
	<b>M</b> ↑	<b>R</b> ↑	<b>D</b> ↓	<b>M</b> ↑	<b>R</b> ↑	<b>D</b> ↓	<b>M</b> ↓
BEAR	0.908	1.000	-0.010	0.809	0.975	0.055	0.251
BLACKSWAN	0.676	1.000	-0.074	0.584	0.958	-0.030	0.437
BMX-BUMPS	0.286	0.386	0.466	0.409	0.466	0.439	1.193
HIKE	0.862	1.000	0.041	0.882	1.000	0.096	0.347
PARKOUR	0.546	0.622	-0.039	0.568	0.714	0.071	1.180
SCHOOLGIRLS	0.515	0.551	-0.138	0.560	0.756	-0.094	0.894
SOAPBOX	0.508	0.608	0.536	0.541	0.546	0.249	0.982
TRAIN	0.752	0.962	-0.085	0.677	0.974	0.027	0.766

**Cuadro 6.2:** Evaluación de las segmentaciones obtenidas por la densificación de trayectorias no supervisadas. Ver Cuadro 6.1 para descripción de las columnas.

Secuencia	$\mathfrak{J}$			$\mathfrak{F}$			$\mathfrak{T}$
	<b>M</b> ↑	<b>R</b> ↑	<b>D</b> ↓	<b>M</b> ↑	<b>R</b> ↑	<b>D</b> ↓	<b>M</b> ↓
BEAR	0.870	1.000	0.071	0.771	1.000	0.148	0.116
BLACKSWAN	0.371	0.000	-0.050	0.622	1.000	-0.034	0.074
BMX-BUMPS	0.357	0.386	0.750	0.467	0.477	0.890	0.620
HIKE	0.921	1.000	0.030	0.952	1.000	0.026	0.134
PARKOUR	0.145	0.010	-0.259	0.286	0.133	-0.216	0.397
SCHOOLGIRLS	0.565	0.833	-0.144	0.621	1.000	-0.075	0.393
SOAPBOX	0.810	1.000	0.054	0.784	1.000	0.157	0.259
TRAIN	0.883	1.000	0.003	0.811	1.000	0.030	0.097

lado, interesa comparar las medidas de NLCV con las de la densificación de trayectorias no supervisadas (clasificadas por modelado de fondo); por el otro, estudiar la evolución de la calidad del algoritmo interactivo cuando crece el número de anotaciones sirve para conocer su eficiencia respecto de una segmentación manual. En los cuadros 6.1 y 6.2 se agrupan las puntuaciones de los dos métodos no supervisados, mientras que en los cuadros 6.3 y 6.4 se encuentran las del método interactivo con diferentes anotaciones (una única en el Cuadro 6.3, y una cada aproximadamente treinta fotogramas en el Cuadro 6.4). En ellos se recogen las calificaciones de las secuencias segmentadas en los capítulos 4 y 5 (vistas en las figuras 4.3, 4.4, y 5.3 a 5.5), evaluadas tomando como referencia su verdad terreno facilitada en DAVIS, exemplificada en la Figura 6.1 junto con los resultados de los algoritmos a contrastar. Todo el contenido de los cuadros se ha obtenido mediante el código de evaluación que los autores de la base de datos proporcionan junto a ella [39], sin distinguir múltiples objetos entre sí. En las figuras 6.2 a 6.8 al final del capítulo se muestran también resultados cualitativos de todos los algoritmos para cada secuencia evaluada, con tal de facilitar su comparación.

**Cuadro 6.3:** Evaluación de las segmentaciones obtenidas de forma interactiva anotando un único fotograma de sus secuencias. Ver Cuadro 6.1 para descripción de las columnas.

Secuencia	$\mathfrak{J}$			$\mathfrak{F}$			$\mathfrak{T}$
	$M \uparrow$	$R \uparrow$	$D \downarrow$	$M \uparrow$	$R \uparrow$	$D \downarrow$	$M \downarrow$
BEAR	0.949	1.000	0.000	0.952	1.000	-0.030	0.100
BLACKSWAN	0.931	1.000	-0.003	0.954	1.000	-0.011	0.065
BMX-BUMPS	0.515	0.614	0.436	0.642	0.739	0.592	0.868
HIKE	0.914	1.000	-0.032	0.949	1.000	-0.072	0.133
PARKOUR	0.156	0.051	0.231	0.230	0.071	0.303	1.107
SCHOOLGIRLS	0.597	0.513	-0.464	0.631	0.718	-0.412	0.642
SOAPBOX	0.815	1.000	0.048	0.794	1.000	0.136	0.258
TRAIN	0.885	1.000	0.000	0.819	1.000	0.011	0.095

**Cuadro 6.4:** Evaluación de las segmentaciones obtenidas de forma interactiva con anotaciones periódicas. Ver Cuadro 6.1 para descripción de las columnas.

Secuencia	$\mathfrak{J}$			$\mathfrak{F}$			$\mathfrak{T}$
	$M \uparrow$	$R \uparrow$	$D \downarrow$	$M \uparrow$	$R \uparrow$	$D \downarrow$	$M \downarrow$
BEAR	0.951	1.000	-0.001	0.958	1.000	-0.033	0.104
BLACKSWAN	0.931	1.000	-0.005	0.954	1.000	-0.011	0.067
BMX-BUMPS	0.544	0.682	0.343	0.689	0.807	0.397	0.930
HIKE	0.932	1.000	0.001	0.968	1.000	-0.033	0.126
PARKOUR	0.357	0.051	0.127	0.477	0.357	0.098	0.682
SCHOOLGIRLS	0.814	1.000	0.026	0.807	1.000	0.005	0.520
SOAPBOX	0.827	1.000	0.013	0.801	1.000	0.115	0.259
TRAIN	0.885	1.000	0.000	0.819	1.000	0.011	0.095

Lo primero que se observa analizando las puntuaciones de los algoritmos no supervisados es que las segmentaciones que proceden de la densificación de trayectorias son mucho más estables en el tiempo que las de NLCV. Este hecho tiene explicación en que las trayectorias aportan coherencia temporal a las segmentaciones, forzando que sigan fluidamente el movimiento de los objetos después de la densificación. Por contra, la base de NLCV está en la división de los vídeos en regiones, que se obtienen independientemente en cada fotograma, y cuya conexión utiliza información espacio-temporal limitada, lo que puede causar fluctuación en los bordes de los objetos. Por estos mismos motivos, y porque la densificación en el espacio bilateral preserva bien dichos bordes, los valores de  $\mathfrak{F}$  son generalmente más elevados cuando se utilizan trayectorias. Concretamente, en los cuadros 6.1 y 6.2 se aprecia que su sensibilidad es perfecta en todas las secuencias excepto dos, mientras que NLCV es irregular en ese aspecto. En cuanto al índice de Jaccard, tanto su media como sensibilidad varían entre secuencias: el modelado geométrico de fondo es ligeramente más preciso cuando los fondos de las secuencias son rígidos y planares (como en SCHOOLGIRLS, SOAPBOX o TRAIN), mientras que en escenas con movimientos más complejos (como BLACKSWAN

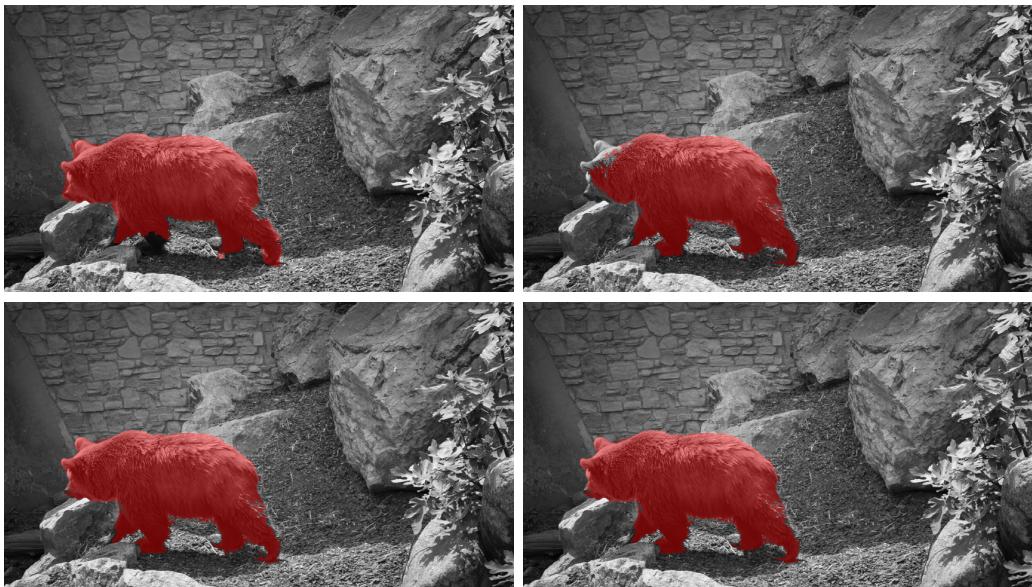
## 6. EVALUACIÓN DE LOS ALGORITMOS

---

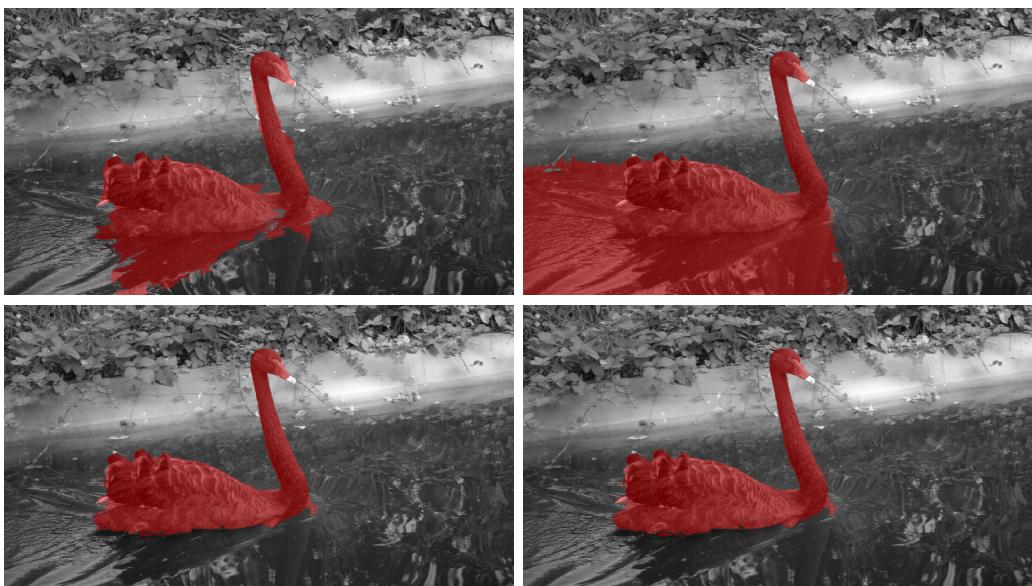
o PARKOUR), NLCV es capaz de identificar los objetos con menos error. Finalmente, son destacables los deterioros anómalos de BMX-BUMPS (en ambos métodos) y de SOAPBOX (solo en NLCV): el primero se debe a una oclusión en el tramo final del vídeo que deja al ciclista fuera del encuadre, imposibilitando su seguimiento; el segundo, a la falta de movimiento dominante después de un cambio de perspectiva, que impide la inicialización de votos en NLCV, pero no la continuación de las trayectorias. Así, el uso de trayectorias es preferible cuando la secuencia a segmentar cumple las hipótesis para su modelado de fondo, mientras que NLCV puede considerarse como un método más equilibrado y versátil, capaz de segmentar vídeos complejos.

En la variante interactiva de densificación de trayectorias, el aumento de la cantidad de anotaciones afecta especialmente a las secuencias que presentan desplazamientos rápidos y oclusiones. Por ejemplo, en SCHOOLGIRLS, donde las niñas se entrecruzan varias veces, la media y sensibilidad de  $\bar{\gamma}$  y  $\bar{\delta}$  crecen sustancialmente y su deterioro se acerca a 0 cuando se introducen anotaciones adicionales antes y después de las oclusiones. Asimismo, en PARKOUR, donde el giro de cámara y rápido movimiento del deportista hace perder muchas trayectorias, volver a etiquetar los reemplazos de las pérdidas ayuda a darles continuidad, aunque su densificación sigue siendo deficiente. En el resto de secuencias, la mejora que comporta el mayor número de semillas es marginal en todas las medidas. Consecuentemente, este algoritmo interactivo propaga de forma fiable la información que se le proporciona incluso cuando el nivel de interacción es pequeño, y puede llegar a segmentar objetos durante largos períodos de tiempo sin perder calidad. De este modo, la principal razón para requerir añadir anotaciones constantemente es la desaparición de trayectorias; aun así, si este hecho se da muy frecuentemente, es previsible que los resultados sigan sin llegar a un mínimo de calidad, y es aconsejable utilizar otro método que no dependa de ellas.

Los resultados cuantitativos de los cuadros 6.1 a 6.4 suponen que la segmentación evaluada es binaria y solo pretende separar el fondo del resto de objetos. Mientras que la versión original de DAVIS no está diseñada para evaluar segmentaciones de múltiples objetos, en una de sus ampliaciones [42] sí se contempla la posibilidad de que se den. Como solamente una de las ocho secuencias evaluadas (SCHOOLGIRLS) contiene varios objetos, no se han incluido sus medidas cuando estos se toman por separado. Pese a esto, se puede asegurar que su calificación no puede ser superior a la del caso binario, puesto que el algoritmo interactivo de densificación de trayectorias (cuadros 6.3 y 6.4), que es el único de los presentados capaz de distinguir objetos, clasifica los objetos uno a uno de manera independiente entre sí.



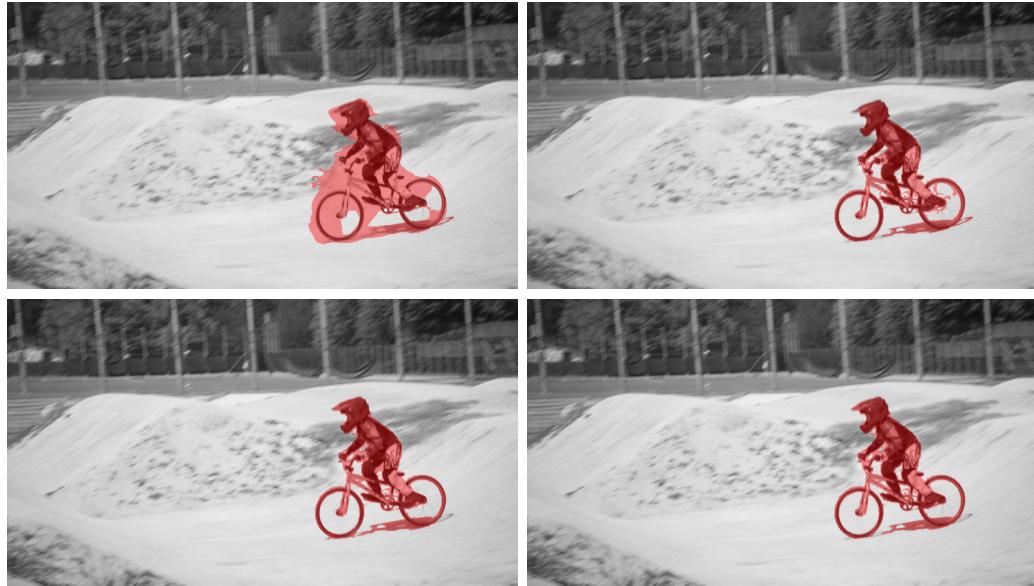
**Figura 6.2:** Segmentación de un fotograma de BEAR con los métodos evaluados. Primera fila: NLCV (izquierda), densificación de trayectorias no supervisada (derecha). Segunda fila: densificación de trayectorias interactiva con anotaciones limitadas (izquierda), con anotaciones detalladas (derecha).



**Figura 6.3:** Segmentación de un fotograma de BLACKSWAN con los métodos evaluados. Imágenes posicionadas igual que en la figura 6.2.

## 6. EVALUACIÓN DE LOS ALGORITMOS

---

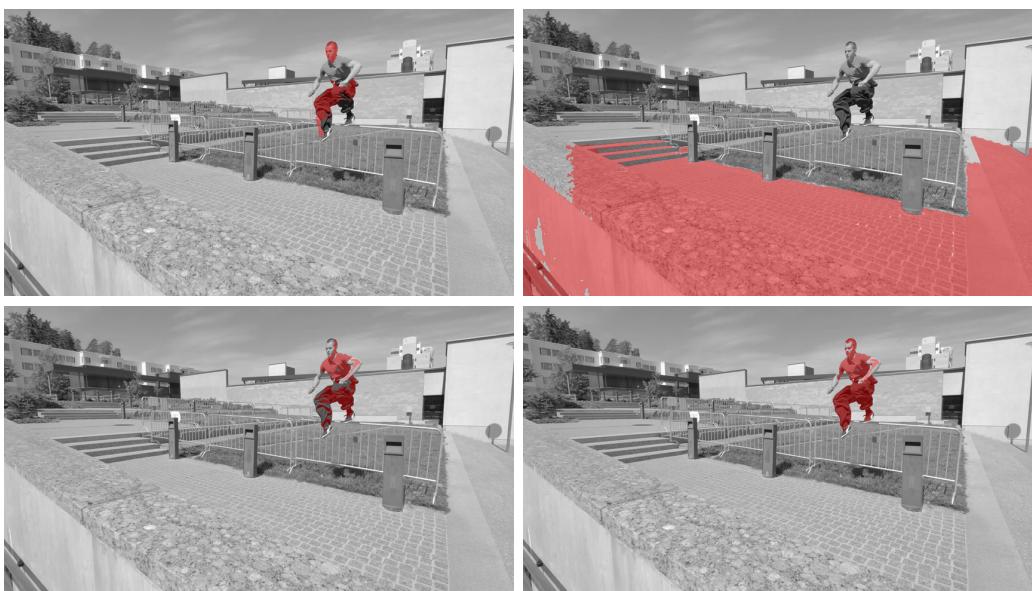


**Figura 6.4:** Segmentación de un fotograma de BMX-BUMPS con los métodos evaluados. Imágenes posicionadas igual que en la figura 6.2.

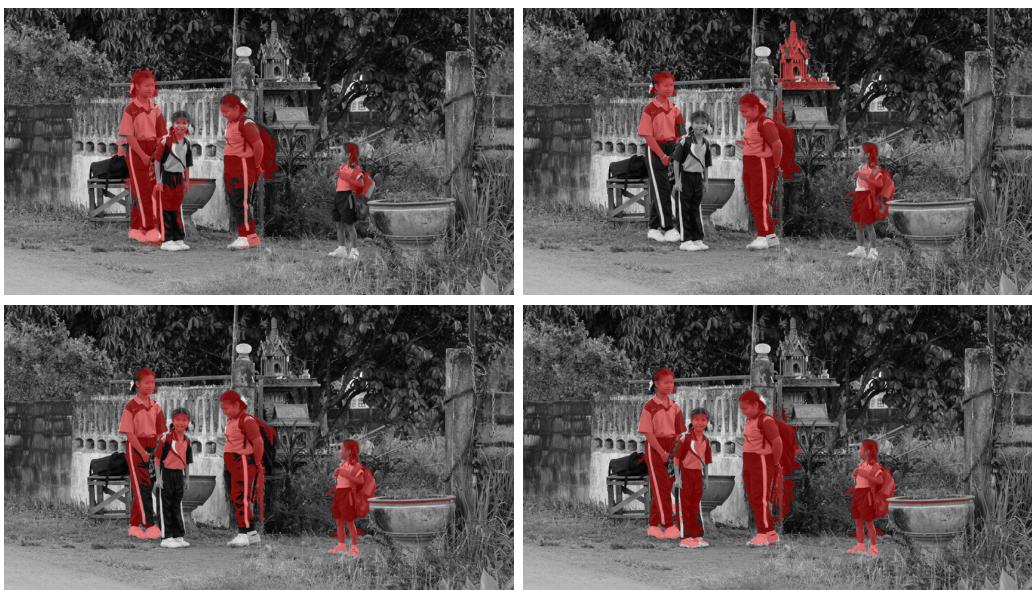


**Figura 6.5:** Segmentación de un fotograma de HIKE con los métodos evaluados. Imágenes posicionadas igual que en la figura 6.2.

## 6.2. Resultados



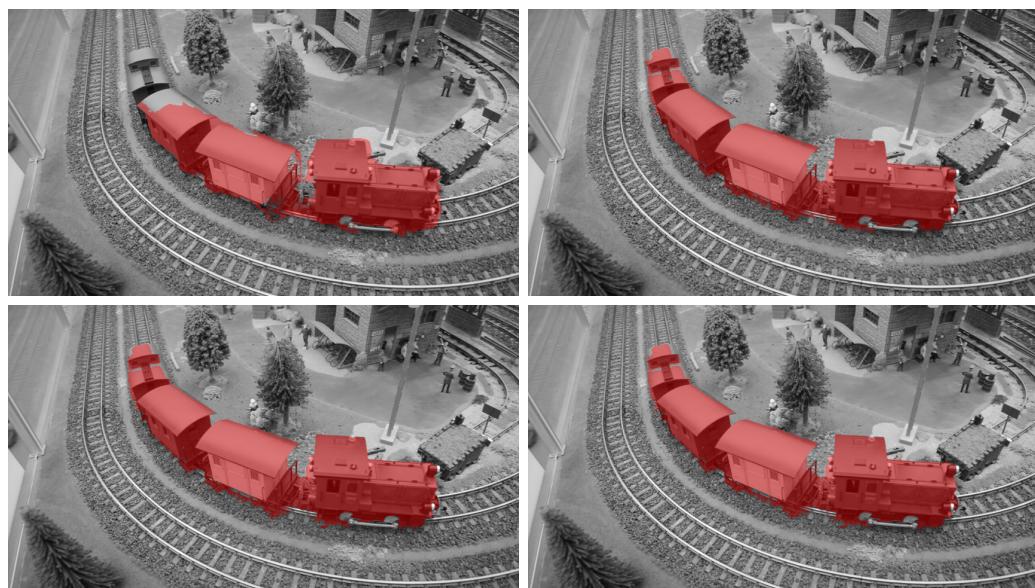
**Figura 6.6:** Segmentación de un fotograma de PARKOUR con los métodos evaluados. Imágenes posicionadas igual que en la figura 6.2.



**Figura 6.7:** Segmentación de un fotograma de SCHOOLGIRLS con los métodos evaluados. Imágenes posicionadas igual que en la figura 6.2.

## 6. EVALUACIÓN DE LOS ALGORITMOS

---



**Figura 6.8:** Segmentación de un fotograma de TRAIN con los métodos evaluados. Imágenes posicionadas igual que en la figura 6.2.

CAPÍTULO



# 7

## CONCLUSIONES Y TRABAJO FUTURO

El objetivo de este trabajo ha sido estudiar diversas técnicas de seguimiento y segmentación de objetos en vídeo. Capaces de identificar y discernir las partes constituyentes de las imágenes de una secuencia, los algoritmos que se han presentado han sido planteados con diferentes intereses en consideración, abarcando desde la separación automática de entidades hasta su clasificación asistida por la interacción con usuarios. La estimación del movimiento entre fotogramas, tanto a corto plazo como en largos intervalos de tiempo, ha erigido la base sobre la que estos algoritmos se han fundado; específicamente, el rastreo de trayectorias de puntos ha servido el doble propósito de hacer un seguimiento de los puntos clave de los vídeos, y de actuar como un peldaño para producir segmentaciones densas. Además, el color y la textura han sido otros aspectos instrumentales para lograr una coherencia visual en las segmentaciones, complementando al movimiento e incluso corrigiéndolo ante imprecisiones.

Para fundamentar los métodos de segmentación exhibidos en el trabajo, se ha visto cómo se interpreta una escena real al proyectarse sobre una imagen. En el Capítulo 2 se ha explicado el proceso de captura de la luz a través de una cámara perspectiva, en el que se han relacionado las coordenadas de los puntos del plano de imagen con las posiciones reales que reproducen. Además, se ha definido la noción de vídeo como una secuencia de imágenes coherente en el tiempo, y se ha estudiado cómo se percibe el movimiento en él cuando avanzan sus fotogramas; para ello, se han comprendido maneras de estimar dicho desplazamiento en forma de campos de flujo óptico, explorando al mismo tiempo los rasgos de las imágenes cuyo rastreo es más fiable. Finalmente, se ha extendido la idea de flujo óptico en el tiempo introduciendo el concepto de trayectorias de puntos, y se ha proporcionado un método capaz de seguir una proporción grande de puntos distribuidos por todo el espacio de imagen, asentando la base para varios métodos de segmentación de objetos.

El Capítulo 3 se ha centrado en la clasificación de las trayectorias obtenidas en el capítulo anterior, con el objetivo de conocer las entidades rastreadas por cada una de ellas. En una primera estrategia no supervisada, se ha planteado un modelado paramétrico del movimiento del fondo a partir de correspondencias de puntos facilitadas

## 7. CONCLUSIONES Y TRABAJO FUTURO

---

por las trayectorias. Desde ahí, se ha dado una medida de ajuste al modelo, detectando como objeto a las desviaciones de su movimiento esperado. La separación resultante entre fondo y objetos ha manifestado ser adecuada en secuencias con fondos rígidos y planares. Por otro lado, se ha expuesto un segundo método de carácter interactivo, basado en la propagación de semillas entre trayectorias. Estas se han configurado como un grafo no dirigido, a cuyas aristas se ha otorgado una medida de afinidad, convirtiendo su segmentación en un problema de corte mínimo normalizado. Se ha demostrado cómo llegar a una solución óptima del problema, que se ha relacionado con el comportamiento de un paseo aleatorio sobre el grafo. Para acabar, con este último se ha esparcido la información de interacción, encontrando una fórmula que evita su simulación, y conduciendo a segmentaciones realistas incluso con pocas semillas.

En el Capítulo 4 se han dejado de lado las trayectorias, explorando un algoritmo de segmentación densa que sustituye sus vínculos estrictos por conexiones probabilísticas no locales entre regiones. Formulando el método como un sistema de votación, los votos iniciales se han definido a partir de la prominencia del movimiento en cada punto, repartida entre superpíxeles de apariencia similar. Describiendo las características distintivas de cada región, y asociándolas entre sí según su parecido, se ha dejado que estas se influencien en un proceso iterativo de decisión por consenso llevado a cabo como un paseo aleatorio, capaz de estabilizar los votos iniciales. Considerando los votos como la certeza de pertenencia a un objeto, la segmentación que deriva de ellos ha demostrado poder tratar con vídeos complejos, problemáticos al usar trayectorias.

En el Capítulo 5 se ha recuperado el enfoque en las trayectorias, buscando una manera de llegar con ellas a segmentaciones densas. Para ello, se ha adaptado un algoritmo semisupervisado basado en el levantamiento de puntos en el espacio bilateral de vídeo y el subsiguiente corte de la rejilla que este forma, sustituyendo anotaciones de usuario por la información parcial de trayectorias preclasificadas. El método modificado ha permitido completar las estrategias del Capítulo 3, corrigiendo sus errores y produciendo segmentaciones completas que preservan los bordes de los objetos.

Finalmente, en el Capítulo 6 se han establecido medidas cuantitativas para la comparación de los algoritmos de segmentación densa descritos en los capítulos anteriores, y se han valorado con ellas diversas secuencias de DAVIS. Los aspectos considerados han sido la precisión por píxel de los objetos, la exactitud en la delimitación de los contornos, y la estabilidad temporal de las segmentaciones, todos ellos en relación a valores de verdad terreno. Todas las estrategias que se han analizado han resultado ser adecuadas para cumplir diferentes finalidades y tratar tipos de vídeos variados, comportando que puedan coexistir y ser utilizadas en distintas situaciones.

Este trabajo ha expuesto solo una parte de las posibilidades que abre la segmentación de objetos en vídeo, donde el foco se ha centrado en secuencias completas con principio y final conocidos. Como trabajo futuro, puede ser prometedor tomar una dirección hacia el desarrollo de estrategias capaces de segmentar vídeos en *streaming* a tiempo real, donde el uso de trayectorias no se ha extendido todavía a pesar de los beneficios que su memoria entraña. Por otro lado, es especialmente interesante el progreso en el ámbito de la clasificación de múltiples objetos, con y sin supervisión, que ha despegado en los últimos años gracias al rápido avance del *machine learning*. Finalmente, mejoras de *hardware* como unidades de procesamiento gráfico más potentes y el uso de múltiples cámaras pueden permitir el desarrollo de algoritmos de segmentación más rápidos y precisos, aplicables a vídeos de alta calidad y duración.

## BIBLIOGRAFÍA

- [1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “SLIC superpixels,” EPFL, Tech. Rep. 149300, 06 2010. 57, 58
- [2] K. Aftab and R. Hartley, “Convergence of iteratively re-weighted least squares to robust M-estimators,” in *2015 IEEE Winter Conference on Applications of Computer Vision*, 02 2015, pp. 480–487. 29, 30
- [3] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski, “A database and evaluation methodology for optical flow,” *International Journal of Computer Vision*, vol. 92, pp. 1–31, 01 2007. 16
- [4] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001. 70
- [5] Y. Boykov and V. Kolmogorov, “An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004. 36, 70, 73, 75
- [6] G. Bradski, “The OpenCV Library,” *Dr. Dobb's Journal of Software Tools*, 2000. 51, 62, 75
- [7] T. Brox and J. Malik, “Large displacement optical flow: descriptor matching in variational motion estimation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 500–513, 2011. [Online]. Available: <http://lmb.informatik.uni-freiburg.de/Publications/2011/Bro11a> 16, 18
- [8] A. Bruhn, J. Weickert, and C. Schnörr, “Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods,” *International Journal of Computer Vision*, vol. 61, pp. 211–231, 02 2005. 15
- [9] S. Brutzer, B. Höferlin, and G. Heidemann, “Evaluation of background subtraction techniques for video surveillance,” in *CVPR 2011*. IEEE, 2011, pp. 1937–1944. 2
- [10] J. Chen, S. Paris, and F. Durand, “Real-time edge-aware image processing with the bilateral grid,” *ACM Trans. Graph.*, vol. 26, p. 103, 07 2007. 66
- [11] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov, “Fast approximate energy minimization with label costs,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2173–2180. 70

---

## BIBLIOGRAFÍA

---

- [12] A. Faktor and M. Irani, “Video segmentation by non-local consensus voting,” *BMVC 2014 - Proceedings of the British Machine Vision Conference 2014*, 01 2014. 1, 2, 53, 56, 57, 61
- [13] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, p. 381–395, Jun. 1981. 27
- [14] J. Gibson and O. Marques, “Optical flow and trajectory estimation methods,” in *SpringerBriefs in Computer Science*, 2016. 10
- [15] L. Grady, “Random walks for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, 2006. 36, 47, 48
- [16] G. Guennebaud, B. Jacob *et al.*, “Eigen v3,” <http://eigen.tuxfamily.org>, 2010. 51, 62
- [17] R. I. Hartley, “In defense of the eight-point algorithm,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580–593, 1997. 27
- [18] M. G. Helander, T. K. Landauer, and P. V. Prabhu, *Handbook of Human-Computer Interaction*, 2nd ed. North-Holland, 08 1997. 10
- [19] B. Horn and B. Schunck, “Determining optical flow,” *Artificial Intelligence*, vol. 17, pp. 185–203, 08 1981. 14
- [20] Jianbo Shi and Carlo Tomasi, “Good features to track,” in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600. 16
- [21] Jianbo Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000. 36, 39, 40, 42, 43
- [22] Jinhui Pan, Chuang Gu, and Ming-Ting Sun, “An MPEG-4 virtual video conferencing system with robust video object segmentation,” in *Proceedings of Workshop and Exhibition on MPEG-4 (Cat. No.01EX511)*, 02 2001, pp. 45–48. 2
- [23] K. Kanatani, Y. Sugaya, and Y. Kanazawa, *Guide to 3D Vision Computation*, ser. Advances in Computer Vision and Pattern Recognition. Springer Nature, 01 2016. 25
- [24] M. Keuper, B. Andres, and T. Brox, “Motion trajectory segmentation via minimum cost multicuts,” in *IEEE International Conference on Computer Vision (ICCV)*, 2015. [Online]. Available: <http://lmb.informatik.uni-freiburg.de/Publications/2015/KB15b> 1, 2, 36, 39, 43
- [25] A. Khoreva, A. Rohrbach, and B. Schiele, “Video object segmentation with language referring expressions,” in *Computer Vision – ACCV 2018*, C. Jawahar, H. Li, G. Mori, and K. Schindler, Eds. Cham.: Springer International Publishing, 05 2019, pp. 123–141. 2

- [26] C. Kim and J.-N. Hwang, “Fast and automatic video object segmentation and tracking for content-based applications,” *IEEE transactions on circuits and systems for video technology*, vol. 12, no. 2, pp. 122–129, 2002. 2
- [27] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, “Video panoptic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9859–9868. 2
- [28] V. Kolmogorov and R. Zabih, “What energy functions can be minimized via graph cuts?” *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, pp. 147–159, 03 2004. 70
- [29] R. Krishna, “Computer vision: Foundations and applications,” Dec. 2017. [Online]. Available: [http://vision.stanford.edu/teaching/cs131\\_fall1718/files/cs131-class-notes.pdf](http://vision.stanford.edu/teaching/cs131_fall1718/files/cs131-class-notes.pdf) 10
- [30] T. Kroeger, R. Timofte, D. Dai, and L. V. Gool, “Fast optical flow using dense inverse search,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 16
- [31] O. Lezoray and L. Grady, *Image Processing and Analysis with Graphs: Theory and Practice*, ser. Digital Imaging and Computer Vision. CRC Press, 2012, <http://greyc.stlo.unicaen.fr/lezoray/IPAG/>. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-00813324> 69
- [32] J. Li, P. Yuan, D. Gu, and Y. Tian, “Hierarchical deep cosegmentation of primary objects in aerial videos,” *IEEE MultiMedia*, vol. 26, no. 3, p. 9–18, Jul. 2019. 2
- [33] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proceedings of the International Joint Conference and Artificial Intelligence*, 04 1981, pp. 674–679. 12
- [34] K.-K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, “Video object segmentation without temporal information,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018. 1
- [35] M. Meilă and J. Shi, “A random walks view of spectral segmentation,” in *AISTATS*, Jan. 2001. 36, 43, 45
- [36] N. Märki, F. Perazzi, O. Wang, and A. Sorkine-Hornung, “Bilateral space video segmentation,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 743–751. 1, 2, 65, 67, 68, 69, 70, 71, 75
- [37] N. Nagaraja, F. Schmidt, and T. Brox, “Video segmentation with just a few strokes,” in *IEEE International Conference on Computer Vision (ICCV)*, 2015. [Online]. Available: <http://lmb.informatik.uni-freiburg.de/Publications/2015/NSB15> 1, 2, 36, 47, 48, 49
- [38] P. Ochs, J. Malik, and T. Brox, “Segmentation of moving objects by long term video analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1187 – 1200, Jun. 2014, preprint. [Online]. Available: <http://lmb.informatik.uni-freiburg.de/Publications/2014/OB14b> 1, 2, 36, 38, 39

- [39] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. V. Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 724–732. 2, 20, 35, 48, 62, 77, 80, 82
- [40] K. Plataniotis and A. Venetsanopoulos, *Color Image Processing and Applications*, ser. Digital Signal Processing. Springer Verlag. Berlin, 01 2000. 8, 9, 10
- [41] J. Pont-Tuset and F. Marques, "Supervised evaluation of image segmentation and object proposal techniques," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 38, no. 7, pp. 1465–1478, 2016. 78, 79
- [42] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 DAVIS challenge on video object segmentation," *arXiv:1704.00675*, 2017. 78, 84
- [43] C. Schnörr, "Determining optical flow for irregular domains by minimizing quadratic functionals of a certain class," *Int. J. Comput. Vision*, vol. 6, no. 1, p. 25–38, Apr. 1991. 14
- [44] P. Sturm, "Pinhole camera model," in *Computer Vision: A Reference Guide*, K. Ikeuchi, Ed. Boston, MA: Springer US, 2014, pp. 610–613. 6
- [45] N. Sundaram, T. Brox, and K. Keutzer, "Dense point trajectories by GPU-accelerated large displacement optical flow," in *European Conference on Computer Vision (ECCV)*, ser. Lecture Notes in Computer Science. Springer, Sep. 2010. [Online]. Available: <http://lmb.informatik.uni-freiburg.de/Publications/2010/Bro10e> 18, 19, 31
- [46] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010. 11
- [47] C. Tomasi and T. Kanade, "Detection and tracking of point features," *International Journal of Computer Vision*, Tech. Rep., 1991. 17
- [48] S. Wehrwein and R. Szeliski, "Video segmentation with background motion models," in *BMVC*, 2017. 1, 2, 24, 25, 31, 33, 36, 67, 72, 75
- [49] R. Yao, G. Lin, S. Xia, J. Zhao, and Y. Zhou, "Video object segmentation and tracking: A survey," *ACM Transactions on Intelligent Systems and Technology*, vol. 11, no. 4, May 2020. 2
- [50] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L1 optical flow," in *Joint pattern recognition symposium*, vol. 4713, 09 2007, pp. 214–223. 15
- [51] J. Zaragoza, T.-J. Chin, M. Brown, and D. Suter, "As-projective-as-possible image stitching with moving DLT," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 06 2013, pp. 2339–2346. 26
- [52] Z. Zhang, P. Tang, and R. Duan, "Dynamic time warping under pointwise shape context," *Information Sciences*, vol. 315, pp. 88–101, 09 2015. 80