

**UNIVERSIDAD AUTONOMA DE MADRID**

**ESCUELA POLITECNICA SUPERIOR**



**Grado en Ingeniería Informática**

**TRABAJO FIN DE GRADO**

**Diseño de herramientas de apoyo para la detección de logotipos  
en secuencias de video**

**Efrén Martín García**  
**Tutor: Marcos Escudero Viñolo**  
**Ponente: Jesús Bescós Cano**

**JUNIO 2016**



# DISEÑO DE HERRAMIENTAS DE APOYO PARA LA DETECCIÓN DE LOGOTIPOS EN SECUENCIAS DE VIDEO



**AUTOR: Efrén Martín García**  
**TUTOR: Marcos Escudero Viñolo**  
**PONENTE: Jesús Bescós Cano**



**Video Processing and Understanding Lab**  
**Dpto. de Tecnología Electrónica y de las Comunicaciones**  
**Escuela Politécnica Superior**  
**Universidad Autónoma de Madrid**  
**Junio de 2016**

Trabajo parcialmente financiado por el Ministerio de Economía y Competitividad del Gobierno de España bajo el proyecto TEC2014-53176-R (HAVideo) (2015-2017)





# Resumen

Este Trabajo Fin de Grado se ha realizado usando herramientas y conceptos de visión por ordenador para poder desarrollar métodos analíticos que permitan procesar una secuencia de video y obtener distintos tipos de parámetros o datos que de forma independiente o encadenados puedan llevar a realizar detecciones de logos (precargados o no) en los distintos fotogramas de la secuencia a procesar.

El trabajo no se realiza sólo sobre un concepto dentro de la visión por ordenador y el procesado de imagen, sino que se intentan abarcar el máximo de herramientas y conceptos que pueden ser utilizados para detectar un logo, ya sean de color o forma.

El método comienza definiendo tres pasos de pre-procesado que, motivados por las heurísticas del diseño, determinan las áreas donde un logo es más susceptible de ser localizado. Específicamente, los métodos usados son estrategias basadas en técnicas estructurales, saliencia y color que vayan reduciendo las zonas donde se ejecutarán las tareas de detección. Además, una detección de regiones estáticas en la secuencia evita detecciones en éstas áreas.

En este proyecto, la detección de logotipos se logra mediante una serie de pasos, siendo el primero y más innovador el preprocesado, seguido del uso de segmentado de la imagen y *matching* de puntos de interés para alcanzar el reconocimiento correcto de un logotipo, que luego será revisado por varias técnicas incluyendo un módulo de perspectiva que detecta si el *match* está en la perspectiva general de la toma. Los logos se detectan midiendo el grado de similitud entre la plantilla transformada y el área candidata.

Los resultados experimentales en una serie de secuencias elegidas validan parcialmente el diseño y método para transmisiones futbolísticas. Aunque por otro lado, los resultados muestran las limitaciones y problemas del método al analizar secuencias de otros deportes.

Además, también se incluyen experimentos preliminares del uso de éste método en la generación de estadísticas enfocadas al análisis publicitario, dando resultados prometedores.

En términos generales, los resultados sugieren que el uso de técnicas de pre-procesado puede ayudar en la labor de detección automática de logotipos.

## Palabras clave

Visión por ordenador, detección, logos, logotipos, comparación, puntos de interés, color, homografía, distancia, saliencia,

# Abstract

This work describes an automatic method for the detection of brand logos in sport sequences. The work starts by studying the solutions existing in the state-of-the-art in the topic. From this study a set of conclusions is derived, and these are used to define the design of the proposed method.

The method starts by defining three pre-processing methods which—motivated by design-heuristics—determine the spatial areas on which a logo is prone to be placed. Specifically, the methods use colour, structural and saliency based strategies to constrain the areas on which the logo detection process takes place.

On the candidate areas—those prone to contain a logo—, a classical point-of-interest matching strategy is used to relate the candidate instances with a preload logo template. From these matches, an affine correction of the template is derived. Logos are detected by measuring the similarity between the transformed template and the candidate areas.

Experimental results on a set of candidate sequences partially validate the design and development of the method for soccer sequences. However, results also illustrate the method's drawbacks and limitations when analysing sequences of alternative sports. Furthermore, preliminary experiments on the use of the method for the generation of publicity statistics are also included, obtaining promising results. In overall, results suggest that the use of pre-processing techniques may help in the task of automatic logo detection.

# Keywords

Computer vision, detection, logos, logotypes, comparison, points of interest, colour, homography, distance, saliency,







# INDICE DE CONTENIDOS

<b>1</b>	<b>INTRODUCCIÓN</b>	<b>2</b>
1.1	MOTIVACIÓN	2
1.2	OBJETIVOS	2
1.3	ORGANIZACIÓN DE LA MEMORIA	3
<b>2</b>	<b>ESTADO DEL ARTE</b>	<b>5</b>
2.1	HERRAMIENTAS A UTILIZAR	7
2.1.1	PREPROCESADO DE LA IMAGEN	7
2.1.2	PUNTOS DE INTERÉS (POIS)	7
2.1.3	REGIONES DE INTERÉS	8
2.1.4	DESCRIPTORES DE FORMA	8
2.1.5	USO DE COLOR	9
2.1.6	ALMACENAMIENTO/SALIDA	9
2.1.7	DETECCIÓN DE <i>OUTLIERS</i>	10
2.2	RETOS	10
2.2.1	CAMBIOS DE ESCALA	10
2.2.2	ROTACIÓN	11
2.2.3	OTRAS TRANSFORMACIONES AFINES	11
2.2.4	TRANSFORMACIONES NO LINEALES	12
2.2.5	CAMBIOS DE ILUMINACIÓN	12
<b>3</b>	<b>DISEÑO</b>	<b>13</b>
3.1	ELECCIÓN DEL ENTORNO DE PROGRAMACIÓN	13
3.2	PREPROCESADO	13
3.2.1	DETECCIÓN DE ZONAS FIJAS	13
3.2.2	COMPONENTE DE SALIENCIA	13
3.2.3	USO DE FORMA Y PERSPECTIVA	15
3.2.4	COMPONENTE DE COLOR	16
3.2.5	UNIFICACIÓN PARA MÁSCARA DE DETECCIÓN	18
3.3	BÚSQUEDA DEL LOGO	18
3.3.1	BÚSQUEDA DE PUNTOS DE INTERÉS Y <i>MATCHES</i>	19
3.3.2	INFORMACIÓN ESPACIAL DE LOS <i>MATCHES</i>	19
3.3.3	HOMOGRAFÍA Y COMPROBACIÓN	19
3.4	INTERACCIÓN CON EL USUARIO	20
<b>4</b>	<b>DESARROLLO</b>	<b>22</b>
4.1	TRABAJO INICIAL DE INVESTIGACIÓN	22
4.1.1	FAMILIARIZACIÓN CON CONCEPTOS BÁSICOS DE CV	22
4.1.2	BÚSQUEDA Y ANÁLISIS DE DOCUMENTACIÓN DE TRABAJOS ANTERIORES	22
4.1.3	ELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN Y HERRAMIENTAS A USAR	22
4.2	GENERACIÓN DEL CÓDIGO	23
4.2.1	ALGORITMO DE DETECCIÓN DE ZONAS FIJAS	24
4.2.2	ALGORITMO DE DETECCIÓN DE PERSPECTIVA	24
4.2.3	ALGORITMO DE DETECCIÓN DE SALIENCIA	24
4.2.4	ALGORITMO DE DISTANCIA DE COLOR	25
4.2.5	MÁSCARA DEL PREPROCESADO	26
4.2.6	DETECCIÓN Y <i>MATCHING</i> DE POIS	26
4.2.7	VALIDACIÓN DE LA DETECCIÓN POR REGIÓN	27

4.2.8	MECANISMOS DE INTERACCIÓN CON EL USUARIO .....	28
4.2.8.1	Entrada de parámetros .....	28
4.2.8.2	Selección del logotipo por interfaz .....	29
4.2.8.3	Salida de datos por consola .....	29
4.2.8.4	Salida visual por interfaz gráfica .....	29
4.2.8.5	Volcado de datos a archivo .....	30
<b>5</b>	<b><u>INTEGRACIÓN, PRUEBAS Y RESULTADOS .....</u></b>	<b>31</b>
<b>5.1</b>	<b>PROCEDIMIENTO PARA PRUEBAS .....</b>	<b>31</b>
5.1.1	SECUENCIA 1 .....	31
5.1.2	SECUENCIA 2 .....	31
5.1.3	SECUENCIA 3 .....	32
5.1.4	SECUENCIA 4 .....	32
5.1.5	SECUENCIA 5 .....	32
5.1.6	SECUENCIA 6 .....	32
5.1.7	SECUENCIA 7 .....	32
5.1.8	SECUENCIA 8 (PRUEBA DETECCIÓN TEMPORAL) .....	33
<b>5.2</b>	<b>TABLA DE RESULTADOS .....</b>	<b>33</b>
<b>5.3</b>	<b>ANÁLISIS DE LOS RESULTADOS.....</b>	<b>35</b>
<b>6</b>	<b><u>CONCLUSIONES Y TRABAJO FUTURO .....</u></b>	<b>38</b>
<b>6.1</b>	<b>CONCLUSIONES .....</b>	<b>38</b>
<b>6.2</b>	<b>TRABAJO FUTURO .....</b>	<b>39</b>
6.2.1	MEJORA DEL MÓDULO DE PREPROCESADO .....	39
6.2.2	MEJORA DE LA ELIMINACIÓN DE FALSOS <i>MATCHES</i> .....	39
6.2.3	MEJORA DE LA DETECCIÓN DE REGIONES EN LA MÁSCARA DE POIS .....	39
6.2.4	ADAPTACIÓN A SECUENCIAS DISTINTAS A FÚTBOL.....	39
<b>6.3</b>	<b>CONCLUSIÓN PERSONAL .....</b>	<b>39</b>
	<b><u>REFERENCIAS.....</u></b>	<b>40</b>
	<b><u>GLOSARIO .....</u></b>	<b>41</b>
	<b><u>ANEXOS.....</u></b>	<b>I</b>
<b>A</b>	<b>MANUAL DE USO DEL PROGRAMA .....</b>	<b>I</b>

# INDICE DE FIGURAS Y TABLAS

TABLA 2-1: TABLA DEL ESTADO DEL ARTE.....	6
FIGURA 2-2: USO DE DESCRIPTORES TRIANGULARES CON POIS EN EL TRABAJO [4] EXTRAÍDA DE LA DOCUMENTACIÓN DEL MISMO. ....	8
FIGURA 2-3: USO DE COLOR PARA DETECTAR POR ROIS EL LÍMITE DEL CAMPO DE JUEGO EN [6]. ADAPTADA DE [6]. (A) FOTOGRAMA ORIGINAL (B) UMBRALIZACIÓN POR TONO (C) MÁSCARA DE CONTORNOS POR SOBEL[15] (D) POSIBLES LÍMITES DEL SUELO .....	9
FIGURA 2-4: USO DE MSER MULTI-RESOLUCIÓN[18] POR [3] EN EL LOGO DE ADIDAS™ . IMAGEN EXTRAÍDA DE LA DOCUMENTACIÓN DE [3]......	11
FIGURA 2-5: POSIBLES TRANSFORMACIONES AFINES . ADAPTADO DE [19]......	12
FIGURA 3-1: FLUJO DE DATOS DEL MÓDULO DE SALIENCIA A NIVEL DE PÍXEL.....	14
FIGURA 3-2: SALIDAS DE LAS DISTINTAS FUNCIONES DE SALIENCIA. A- NIV. DE PÍXEL MODO 2. B- NIV. DE PÍXEL MODO 1. C- SALIENCIA POR HISTOGRAMAS SIN UMBRALIZAR .....	15
FIGURA 3-3: IMAGEN SINTÉTICA. PROCESO DE SELECCIÓN DE REGIONES ADYACENTES DE COLOR	17
FIGURA 3-4: CAPTURA DEL MÓDULO DE COLOR. LAS IMÁGENES DE ARRIBA SON REGIONES CON DISTANCIAS BAJAS A CADA UNO DE LOS COLORES DEL LOGO. ABAJO SE ENCUENTRAN LAS REGIONES ADYACENTES Y LA IMAGEN ORIGINAL DE REFERENCIA.....	17
FIGURA 3-5: FLUJO DE DATOS EN EL MÓDULO DE DISTANCIA DE COLOR (FUNCIÓN DE DISTANCIA EUCLÍDEA). ....	18
FIGURA 3-6: MOCKUP DE LA VENTANA DE SELECCIÓN DEL LOGO DENTRO DE LA SECUENCIA. ....	20
FIGURA 3-7: MOCKUP DE INTERFAZ DE SALIDA DE DATOS DEL ANÁLISIS DE FOTOGRAMA.....	21
FIGURA 4-1: DIAGRAMA DE FLUJO DE DATOS GENERAL DEL SISTEMA. ....	24
FIGURA 4-2: DISTANCIAS DE COLOR A BLANCO (IZDA. DELTA-E, DCHA. EUCLIDEA) PARA LA DETECCIÓN DEL LOGO DE COCACOLA™ EN LA IMAGEN DE ABAJO. ....	25
FIGURA 4-3: IMAGEN SINTÉTICA PARA VISUALIZAR LA OBTENCIÓN DE LA MÁSCARA FINAL SEGÚN LA EXPLICACIÓN EN 4.2.5. SIENDO LAS IMÁGENES POR ORDEN LA MÁSCARA DE ELEMENTOS ESTÁTICOS, LA DE SALIENCIA, LA DE COLOR Y ABAJO LA FINAL DE PREPROCESADO. ....	26
FIGURA 4-4: MÁSCARA DE PUNTOS DE INTERÉS EN UNA DETECCIÓN DEL LOGO DE WILLIAM HILL. ....	27
FIGURA 4-5: EXPLICACIÓN SINTÉTICA DEL RECHAZO POR BAJA COBERTURA CON UN EJEMPLO DE HOMOGRAFÍA ERRÓNEA.....	28
FIGURA 4-6: CAPTURA DE LA PANTALLA DE CARGA DEL PROGRAMA QUE APARECE AL COMIENZO DE LA EJECUCIÓN. ....	28

FIGURA 4-7: CAPTURA DE LA PANTALLA DE LA PANTALLA DE SELECCIÓN DE LOGOTIPO ESTANDO SELECCIONADOS TRES DE LOS 4 PUNTOS NECESARIOS. ....	29
FIGURA 4-8: CAPTURA DE LA PANTALLA DE ANÁLISIS DE FOTOGRAMA DONDE SE PUEDEN VER POR ORDEN: EL FOTOGRAMA ORIGINAL, LA SALIDA DEL MÓDULO DE DETECCIÓN DE PERSPECTIVA, LA MÁSCARA DE SALIENCIA, LA MÁSCARA DE COLOR, LA MÁSCARA DE NUBES DE POIS Y FINALMENTE LAS DETECCIONES DE LOGOS. ....	30
FIGURA 4-9: SALIDA DE ARCHIVO DE UNA DETECCIÓN EN UN FOTOGRAMA ESTANDO ACTIVADA ESTA FUNCIÓN.....	30
FIGURA 5-1: TABLA DE LOS RESULTADOS DE LAS DISTINTAS PRUEBAS. EN LAS COLUMNAS TP Y FP APARECE EL NÚMERO TOTAL SEGUIDO DE LAS DISTANCIAS EN CURSIVA. ....	34
FIGURA 5-2: DIAGRAMA DE PRECISIÓN- <i>RECALL</i> EN FUNCIÓN DEL UMBRAL PARA LA SECUENCIA 3.34	
FIGURA 5-3: LÍNEA DE TIEMPO PARA DETECCIONES EN UNA SECUENCIA. ....	35
FIGURA 5-4: DISTRIBUCIÓN DE LAS DISTANCIAS DE LOS MATCHES (ROJO - FPs, VERDE - TPS). LOS DISTINTAS INTENSIDADES SE DEBEN A SUPERPOSICIÓN. ....	35
FIGURA 5-5: CAPTURA DE DETECCIÓN CON OCLUSIÓN .....	35
FIGURA 5-6: MÁSCARAS DE SALIENCIA, COLOR Y POIS DE UN FOTOGRAMA DE LA SECUENCIA 6 ..	36

# 1 Introducción

---

## 1.1 Motivación

Esta memoria de TFG se dedicará a explicar los pasos que se han seguido para realizar el análisis, investigación y aprendizaje de técnicas, incluyendo la investigación de trabajos y proyectos anteriores realizados sobre la detección de logotipos en imágenes y secuencias. Se describirá detalladamente el proceso de utilización de las técnicas aprendidas y elegidas para este ámbito y finalmente se detallarán aquí el diseño de pruebas que permitan evaluar la calidad de los resultados que podemos obtener con la máxima variedad de entradas al programa, realizando también un análisis de estos resultados.

El porqué de la elección de este tema y ámbito es una suma de causas. Por un lado, la utilidad de esta clase de detecciones automáticas es indudable en un mundo tan mediatizado como el de hoy, y con los deportes cobrando más y más importancia y alcanzando aficiones en cada vez más países. El negocio de la publicidad deportiva mueve millones de euros y al fin y al cabo, los estudios que realizamos en este trabajo están enfocados al análisis publicitario en transmisiones de vídeo de estadios y campus de distintos deportes.

Ya ha habido una intensa labor de investigación con buenos resultados, que podrán verse en más detalle en la sección 2 de esta memoria, e incluso ya existe algún software comercial o servicio <sup>[1]</sup>. Sin embargo, no existe aún una solución que puede ser aplicada en entornos genéricos. Los retos que debe solventar un algoritmo de detección de logotipos incluyen: oclusiones, cambios del punto de vista, deformaciones no rígidas, etc.

Uno de los problemas de los programas que se dedican al análisis de video es la duración del procesado de cada fotograma. La importancia de realizar un preprocesado de este para obtener las regiones donde realmente deberemos buscar el logo es una gran forma de reducir el costo del análisis por fotograma. Además, el preprocesado reduce el número de distractores evitando falsas detecciones en zonas donde se determina que no hay logotipos. Este trabajo destaca en ese apartado respecto a otros, haciendo una fuerte apuesta por la eficiencia del código

Además, las técnicas usadas para la detección de logos en una transmisión de un evento deportivo pueden ser adaptadas para otro tipo de detecciones en otro tipo de secuencias de vídeo o imágenes. Un logo es un ejemplo de imagen artificial con unas cualidades particulares que también pueden tener elementos como matrículas, señales, etc. Y por tanto, hay una infinidad de otras circunstancias en las que la detección automática de una marca o un elemento parecido puede resultar útil.

## 1.2 Objetivos

- Análisis del estado del arte actual y las relaciones de éste con la detección y reconocimiento automáticos de logotipos en imágenes o secuencias de vídeo buscando los mejores logros hasta el momento en éste ámbito así como las áreas con menos uso y por tanto, donde pueda haber un nicho de explotación.
- Evaluación los retos que se presentan en la detección de imágenes y logotipos para poder diseñar herramientas específicas que eviten o aminoren el efecto de éstos.
- Diseño de estrategias con la finalidad de reducir virtualmente el área de una imagen donde existen probabilidades altas de localizar logos o imágenes de marca utilizando información de apariencia conocida.

- Estimación analítica de la perspectiva de una imagen en vídeo mediante el análisis de rectas en ésta y la obtención de líneas de fuga.
- Diseño de herramientas que permitan obtener la información temporal de visualización de un determinado logo en una secuencia de vídeo.
- Creación de un código que permita realizar los apartados anteriores prestando especial atención en apartados como la eficiencia temporal de éste y que incluya una salida visual que permita al usuario visualizar las detecciones de los logotipos cargados en el programa en la propia secuencia.
- Evaluación del método creado a lo largo del periodo de este trabajo en términos de eficacia, robustez a retos mediante el uso de un banco de pruebas elegidas para ser representativas y variadas, incluyendo los retos evaluados en el segundo punto.
- Análisis cuantitativo de los resultados proporcionados por las pruebas realizadas e interpretación de éstos en forma gráfica y conceptual.

### **1.3 Organización de la memoria**

La memoria de este trabajo consta de las siguientes secciones:

- **Estado del arte**

Se desarrolla una organización de los trabajos más recientes e importantes en el panorama de investigación internacional relacionados con la detección y el reconocimiento de logos en imágenes, y más específicamente en transmisiones de vídeos deportivos.

Podremos encontrar una comparativa y contraste resumidos de las rutas y medios que unos y otros han seguido para lograr los mejores resultados posibles y cuáles han sido estos.

- **Diseño**

Se explica detenidamente el funcionamiento del programa y del código. Desde el funcionamiento de las distintas componentes de este a la explicación de cómo interaccionan entre sí y de cómo se complementan.

También se trata el tema de la interacción del usuario, incluyendo los conceptos de entrada y salida de datos. Este apartado incluye diagramas de diseño de software, mockups, etc.

- **Desarrollo**

Detalle paso a paso del proceso de desarrollo de nuestro código. Los resultados o errores que han aparecido y las decisiones que se han tomado para atajarlos e ir mejorando la eficacia, eficiencia y usabilidad del código.

Explicación de todas las funcionalidades y pasos que se siguen en el flujo de datos y se explicará a nivel técnico el funcionamiento de las funciones que se utilizan en este programa.

- **Integración, pruebas y resultados**

Se procede a explicar cómo todo el trabajo realizado se unifica e integra. Una vez tenemos un producto consolidado, se detalla el proceso de testeo del código, que la mayoría de las veces requiere ciertos retoques en el propio código para facilitar las pruebas. El diseño y la composición de las pruebas experimentales realizadas, los mencionados cambios y por supuesto la salida obtenida es detallada en esta sección.

Estos detalles incluyen un análisis de los resultados obtenidos en el que se discutirá la robustez del método propuesto a la máxima variedad posible de retos que sea posible analizar.

- **Conclusiones**

La última parte se reserva para un pequeño resumen del trabajo hecho y un análisis de éste.

Se debatirá sobre hasta qué punto se han alcanzado los objetivos previstos y si los resultados han sido los esperados

Esta parte incluirá tanto detalles generales sobre el desarrollo y el diseño como datos puntuales de carácter técnico sobre el código y el uso de unas u otras herramientas.

## 2 Estado del arte

---

El análisis de imágenes y video en busca de logotipos e imágenes de marca no es algo nuevo. A lo largo de los últimos 15 años, propiciado por un fuerte aumento del uso y mejora de las técnicas de visión artificial, se han venido realizando una amplia variedad de proyectos de investigación en este campo; entre los que se ha realizado una selección para su análisis en esta sección, ya que antes de realizar pruebas con código resulta conveniente conocer las técnicas y métodos que han usado otros anteriormente en el mismo campo.

Los trabajos realizados pueden dividirse de distintos modos. El más intuitivo es el del tipo de entrada, ya que algunos usan, como es el caso que concierne esta memoria, una entrada de secuencia de vídeo, mientras que otros se ejecutan sobre una sola imagen. La realidad es que estas dos opciones son mucho más cercanas de lo que puede parecer ya que un vídeo es una secuencia de imágenes, luego el proceso es a grandes rasgos el mismo salvo por que en una secuencia, elementos como el costo temporal del análisis por fotograma adquiere una importancia mucho mayor y la salida de los datos suele estar también expresada en función de una variable tiempo de la que los trabajos con imágenes estáticas carecen.

Por otro lado, en el apartado de la detección, existen varios enfoques posibles. Una búsqueda de logos genérica que devuelva o detecte las zonas de la imagen de entrada que sean consideradas por el programa como logotipos; una búsqueda genérica que cuente con una base de datos con la que comparar sus detecciones y que realice la detección sólo si detecta un logo que ya conozca o una búsqueda guiada que cuente con un solo logo y busque basándose en características extraídas de éste de forma más precisa.

Otra variable a tener en cuenta es el entorno de la imagen de entrada. En el caso de este trabajo se trata de un evento deportivo, pero la mayoría de los trabajos que sirven como referencia, salvo ejemplos como [6] y [9], tienen como entrada imágenes naturales totalmente arbitrarias. De esta diferencia puede depender gran parte del preprocesado del logo, ya que en una transmisión deportiva es posible realizar supuestos particulares sobre el encuadre y la perspectiva.

El apartado más característico de este trabajo es la parte de preprocesado que reduce, con técnicas menos costosas que la detección en sí, las regiones de la imagen sobre las que actuar. Este apartado es uno de los puntos menos explotados en los trabajos existentes sobre detección de logos existentes hasta la fecha y es algo en lo que se ha prestado especial atención durante el análisis del estado del arte en este campo.

En la tabla 2-1 se incluye un resumen de los trabajos estudiados y sus características principales, con la intención de realizar una comparativa visual fácil entre las distintas aproximaciones, así como de comparar sus puntos fuertes y resultados en general.



Trabajo (Ref.)	Descriptores del logo					Almacenamiento de datos y Salida del programa	Estimadores de la transformación (Sólo con PoIs)	Retos					Entrada de Video
	Preprocesado	PoIs	RoIs	Análisis de forma	Uso del color			Escala	Rotación	Tranf. afines	Deform. no lineales	Cambios iluminación	
[1]	x	✓SIFT	✓Mean Shift Segmentation	✓	(Para segmentación)	Almacenaje por mejor característica para definir el logo	?	✓	✓	✓	?	✓	x
[2]	x	✓SIFT	x	x	x	Salida de datos estadísticos del video	RANSAC[16]	✓	✓	✓	x	?	✓
[3]	x	✓SIFT (desc. Para MSER)	✓MSER[14] (+LAF)	(K-Means Tree)	x	Almacenamiento de template en K-Means tree de variaciones por familias de modelos	RANSAC[16]	✓	✓	✓	?	✓	x
[4][5]	x	✓Hess-Aff con desc. SIFT ((4)) SURF ((5))	x	x	Color-Sift features en los datos del triangulo	x[4] Visual codebook [5]	Los PoIs crean triángulos usados como "firmas" específicas de cada logo	✓	✓	✓	?	✓	x
[6]	✓ Detección del perímetro del campo	x	✓Detección del perim. Del campo y de separación vert. entre logos	Detección de contexto	Para detección del campo y match (Windowed hue Euclidean dist.)	La detección se produce cada n frames. Los intermedios usan la info anterior.	?	✓	✓?	✓	?	✓	✓
[7]	x	✓SIFT	x	x	x	Matching pyramid	?	✓Probs. a baja res.	✓	✓	?	?	x
[8]	✓ Con el uso de saliencia (SSS)	x	✓Spatial-Spectral Saliency	✓Partial Spatial context (~BOF)	Dentro de saliencia, color self-similarity (HSV)		?	✓	✓	?	x	✓	x
[9]	x	✓SIFT	x	x	Billboards det. With color hist.	Salida visual y estadística de detecc. en video (el usuario puede validar)	?	✓	✓	✓	? "Transfs. Geométricas"	?	✓
[10]	x	✓SIFT	x	NN de los PoIs	x		?	✓	✓	✓	?	?	✓
Propuesto	Por color y saliencia	✓SIFT	✓En preprocesado	Perspectiva	✓En preprocesado	Salida visual y de datos estadísticos	RANSAC[16] + fill	✓	✓	✓	x	✓	✓

Tabla 2-1: Tabla del estado del arte

## 2.1 Herramientas a utilizar

### 2.1.1 Preprocesado de la imagen

Realizar una búsqueda de logos sobre una imagen completa es un proceso costoso. En un vídeo estándar en HD existen casi un millón de píxeles (921600), conteniendo cada uno de ellos una tripla de valores numéricos; en el caso de la definición HD1080 la cifra asciende al doble. Para evitar este alto coste computacional, el preprocesado es posible usar una fase de preprocesado de la imagen, que mediante técnicas menos costosas aunque más genéricas que la comparación y detección real de logotipos, puede permitir reducir notablemente el área de la imagen donde existen posibilidades de encontrar los objetivos de la búsqueda.

El preprocesado, por tanto, es una forma de obtener regiones de interés (de las que se habla en la sección 2.1.3) de manera que éstas sufran los procesos de detección ignorando el resto de la imagen.

El principal ventaja del preprocesado es por tanto la reducción del coste computacional del análisis del fotograma, aunque en sí también conlleve coste. Además de esto, el preprocesado puede ayudar a evitar falsas detecciones si está bien hecho, aunque si es defectuoso también puede evitar que se analicen zonas en las que hay un logo presente y por tanto evitar la detección de este.

El ejemplo más claro de preprocesado en los trabajos analizados es el de [6], que limita su rango de análisis al entorno de su detección del campo.

### 2.1.2 Puntos de Interés (PoIs)

La detección de puntos de interés es una técnica muy usada en el reconocimiento de figuras y patrones en el análisis de imagen; ésta consiste en localizar singularidades como esquinas o transiciones entre dos superficies distintas. Prácticamente todos los trabajos anteriores que se han tenido en cuenta hacían uso de una forma u otro de este conjunto de técnicas. Entre las cuales, como se puede apreciar en la tabla 2-1, destaca particularmente una aproximación denominada Scale-Invariant Feature Transform [12] ya sea en el uso de su detector o de sus descriptores. Esta técnica hace uso de la diferencia de gaussianas de distintas escalas para la detección y usa un descriptor de 16 bins obtenido por el análisis de un vecindario de 16x16 alrededor de cada punto. El trabajo [5] en cambio es el único en esta lista que usa Speeded up Robust Features[13], que destaca por su rapidez respecto a SIFT.

Una vez los puntos han sido detectados y analizados, existen herramientas que mediante uso de motores estadísticos permiten, a partir de dos conjuntos de puntos en dos imágenes, enlazar las dos de modo que mediante una transformación proporcionada por el método denominada homografía, los puntos de una encajen con los de la otra, y esa será la detección buscada.

Existen otros modos de usar los PoIs detectados, como es el caso de [4] que crea firmas espaciales triangulares con ellos (figura 2-2).

El uso de PoIs es muy frecuente, como se ha visto en la detección de logos por las características particulares de los logos en términos de fuertes contrastes entre regiones y el tener patrones y esquinas muy marcadas. El principal inconveniente del uso de éste tipo de técnicas de detección es el alto coste de buscar y describir los puntos en toda la imagen y, sobre todo, de realizar los *matchings* de los puntos.



Figura 2-2: Uso de descriptores triangulares con PoIs en el trabajo [4] extraída de la documentación del mismo.

### 2.1.3 Regiones de interés

Una RoI es cualquier zona de una imagen donde queramos centrar nuestra atención. Dicho, al recortar una imagen ortogonalmente obtenemos una RoI de esta.

Cuando se habla del uso de RoIs en un trabajo de detección de logotipos, al igual que ocurre con los PoIs, se está hablando de la detección automática de estas regiones. En todos estos trabajos analizados, al fin y al cabo, lo que deseamos lograr es detectar automáticamente las regiones de interés que contienen los logotipos buscados.

Una de las técnicas más comunes para extracción automática de RoIs es MSER[14], que consiste en la detección de zonas de la imagen que son estables respecto de luminancia en un rango dado. Existen también técnicas que devuelven regiones de interés según un tipo específico de análisis, como el de saliencia, que devuelve zonas consideradas más *llamativas* al ojo humano.

Los trabajos analizados hacen menos uso de técnicas genéricas de detección de RoIs; salvo casos como [3] que usa MSER para lidiar con los problemas derivados de las uniones de zonas en condiciones de baja resolución o [6] que la aplica a un preprocesado para la detección del campo; que de puntos, ya que un logotipo se caracteriza mucho más por sus esquinas y la estructura de sus detalles que por las características regionales que lo forman.

### 2.1.4 Descriptores de forma

El uso de descriptores de forma permite obtener datos mediante el análisis de patrones geométricos; ya sea en porciones de la imagen, para la búsqueda de regiones que puedan hacer *match* con las formas buscadas; o en la imagen completa, para obtener algún tipo de información sobre esta en su totalidad. Por ejemplo, el uso de detecciones de ciertos patrones de forma puede llevar en una secuencia de video la obtención del contexto de una toma en particular de esta. De esta forma, analizando datos de las formas geométricas de una imagen se logra saber por ejemplo si un fotograma dado es un primer plano o una toma general y este dato puede llevar a la decisión sobre cómo actuar para lograr la finalidad buscada en base a estos datos.

En [3] y, de forma similar en [10] se usan datos de forma y perspectiva para la obtención de una base de datos de logos más versátil. Usando información de forma de las imágenes analizadas, se almacenan *templates* donde los logos han sido sometidos a un subconjunto de transformaciones geométricas (es decir, varias versiones de un mismo logo). El objetivo de este procedimiento es permitir la detección del logo aunque este aparezca deformado de acuerdo a alguna de estas transformaciones. La contraparte es que los procesos de comparación son más complejos y de mayor carga computacional.

### 2.1.5 Uso de color

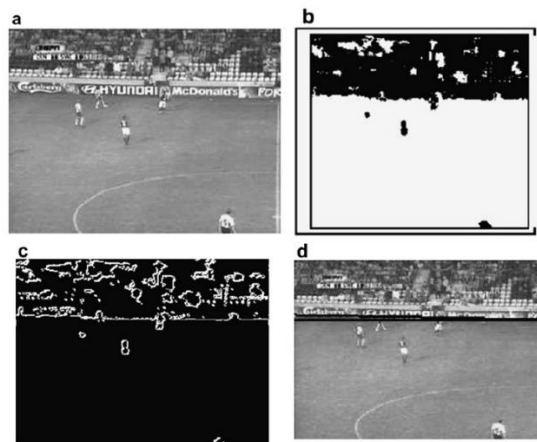
Las imágenes, tanto fijas como los fotogramas de un vídeo, son matrices de números que representan la luminosidad de cada punto. Una imagen en color; que es lo más común y la entrada de todos los trabajos analizados, así como la que tendrá el nuestro; es una matriz con tres dimensiones, siendo la tercera los canales de la imagen (por lo general tres, dependiendo del espacio de color utilizado). Esto implica que si sólo analizamos formas, frecuencias o regiones homogéneas, estamos usando de alguna manera sólo un tercio de la información de la que disponemos.

El uso del color es por tanto muy conveniente y útil al permitir hacer uso de más datos que pueden caracterizar una región; especialmente en la detección de logotipos, ya que frecuentemente, el color de estos es una característica realmente diferenciadora dentro de la imagen general.

Como se ha dicho, el color en una imagen simplemente implica que se tienen varios canales de datos en vez de uno sólo como ocurre con la escala de grises, por lo tanto puede ser añadido a otros pasos, como es el caso de [1] y de [6], que lo usan para detectar regiones de interés (el terreno de juego en el segundo caso, como se muestra en la figura 2-3), o de [5] para los descriptores de puntos de interés detectados.

El coste de realizar los análisis a color es mayor que sobre escala de grises, pero si una operación se quiere hacer en grises sobre una imagen a color, el paso de un modo a otro también tiene un costo.

Por otro lado, usar datos de color en el proceso de detección o de reconocimiento implica necesariamente añadir la restricción de que la entrada de datos al código debe ser en color.



**Figura 2-3: Uso de color para detectar por RoIs el límite del campo de juego en [6]. Adaptada de [6]. (a) Fotograma original (b) Umbralización por tono (c) Máscara de contornos por Sobel[15] (d) Posibles límites del suelo**

### 2.1.6 Almacenamiento/salida

Dependiendo del mecanismo y la filosofía del programa, este almacenará o accederá a más o menos datos guardados, necesitando en el caso de algunos de trabajos analizados bases de datos bastante extensas. Esto es una necesidad cuando la finalidad del software es la detección no de un logotipo en particular sino de uno dentro de un conjunto de logos precargados. Además, evidentemente la salida del programa va a depender en gran medida de las características de este y el uso que queramos darle. Siendo el entorno de este trabajo la visión artificial, en la mayoría de los casos, aunque no en todos, encontramos salidas gráficas, generalmente mediante interfaz (ej. [9]) aunque también mediante volcado a

archivo (ej. [3]) para futuras ejecuciones. Es también común encontrar salidas de datos estadísticos (ej. [2], [9]) ya que la utilidad de muchos de estos programas es precisamente el análisis estadístico de vídeo.

La salida gráfica resulta muy útil en un programa que realice un análisis a tiempo real, por tanto, es prácticamente imprescindible en códigos cuya entrada sea una imagen fija. En los programas encargados de analizar una secuencia de vídeo completa, resultará más ventajoso la salida de datos en bruto o procesados que aporten información genérica sobre las detecciones a lo largo del tiempo en el vídeo; de este modo, el tiempo de procesado no resulta una restricción y no es necesario que haya un operador comprobando la salida durante la ejecución.

La visualización de resultados por pantalla es además bastante costosa, y en el caso de una secuencia, si esta ha de realizarse en cada fotograma, el coste de tiempo puede representar un porcentaje indeseablemente alto en el tiempo de procesado de cada fotograma. Aunque durante el desarrollo del programa es indudablemente útil para detección y corrección de errores.

### **2.1.7 Detección de outliers**

Cualquier proceso de detección automático devuelve unos resultados basados en una serie de análisis. Estos resultados pueden ser correctos o no serlo, y de este ratio depende la calidad del algoritmo que se esté usando. Por lo general, los resultados son evaluables de alguna manera, es decir, se pueden representar de manera numérica o parametrizable y analizar estos valores. Mediante este análisis, hay formas de comprobar si un resultado está fuera de los rangos comunes en el resto de resultados devueltos o de los resultados que se han marcado como correctos de forma manual, es decir, es un resultado aislado o un *outlier*.

La detección de este tipo de salidas resulta útil para desecharlas directamente o aplicar más técnicas sobre ellas para ser capaz de decidir con mayor exactitud si es una salida correcta o no. Los trabajos [2] y [3] usan RANSAC[16], algoritmo matemático iterativo que consiste en detectar un patrón dentro de unos datos y permite determinar si un dato en particular concuerda o no con ese patrón.

A pesar de que por la documentación aportada no se pueda saber con certeza, es razonable pensar que todos los métodos que hacen uso de los PoIs mencionados en la sección 2.1.2 utilicen RANSAC o un algoritmo similar para poder obtener la homografía de la plantilla de búsqueda con la detección que devuelva su sistema.

## **2.2 Retos**

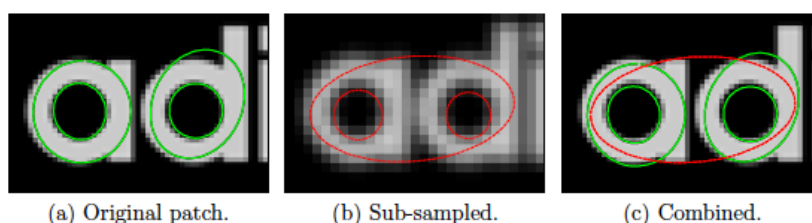
Denominamos retos a cada una de las posibles dificultades que pueden presentar las entradas que introducimos a un código dado. En el campo de la visión artificial y la detección automática de logotipos en imágenes naturales o secuencias de vídeo, se han tomado como referencia algunos de los retos más comunes y problemáticos a los que todos estos trabajos deben enfrentarse.

### **2.2.1 Cambios de escala**

En el ámbito del reconocimiento de logos, es complicado encontrar una aparición del logo objetivo que tenga exactamente el mismo tamaño en píxeles que la imagen plantilla que se usa para la búsqueda. Si así fuese bastaría con realizar un barrido de la imagen precargado sobre el fotograma.

Cualquier aproximación que haga uso de puntos de interés ha de ser robusta a cambios de escala, ya que la detección y obtención de homografía por PoIs lo es, así como también lo es a otro tipo de transformaciones afines. El cambio de escala es un caso particularmente sencillo de transformación afín.

Si bien el problema de un cambio de escala como tal no supone un problema, a medida que la escala llega a ciertos límites mínimos de tamaño, se pierden datos de la imagen. Esto tiene su explicación en el hecho de que, al ser la imagen una matriz de píxeles, una matriz de 10x30 por ejemplo, va a contener siempre menos información espacial que una de 100x300, cien veces menos. Esta pérdida de información en imágenes naturales se presenta como un suavizado, haciendo que desaparezcan esquinas y que áreas que deberían estar separadas se unan. Aunque en la documentación de la mayoría de los trabajos, salvo el caso de [7], no se hace referencia a este efecto, es fácil suponer que salvo que tengan una herramienta para combatir este efecto, todos se enfrentarán a fallos de detección en estas condiciones. El único caso en que hay un método de combatirlo es en [3], que usando el detector de MSER modificado para actuar en varias resoluciones[18] junto con un descriptor llamado LAF[19] es capaz de detectar las áreas que se unen al reducir el tamaño de la imagen. En la figura 2-4 se ve un ejemplo de aplicación de este algoritmo.



**Figura 2-4: Uso de MSER multi-resolución[18] por [3] en el logo de Adidas™ . Imagen extraída de la documentación de [3].**

## 2.2.2 Rotación

La rotación de una imagen es transformación euclideana que se encuentra de forma regular en las imágenes naturales. Es difícil lograr en una imagen natural que en este caso un logo se encuadre de forma totalmente paralela al plano de la imagen.

De manera similar a lo explicado en el apartado 2.2.1, la rotación es una transformación afín muy sencilla a la que también es robusta cualquier técnica que utilice detección de puntos de interés.

## 2.2.3 Otras transformaciones afines

Además de cambios de escala y rotación, una imagen puede sufrir otras transformaciones afines como inclinación o reflexión. De nuevo y como en los dos puntos anteriores, aunque en algunos casos falten datos en la documentación que lo confirmen, todos los trabajos que usen detección por puntos de interés deben resistir estas transformaciones en los logotipos y ser capaces de localizarlos correctamente salvo que en algún paso posterior a la aplicación de SIFT[12] o SURF[13] se descarte el *match*.

Con la figura 2-5 se pueden comprender mejor las transformaciones afines.

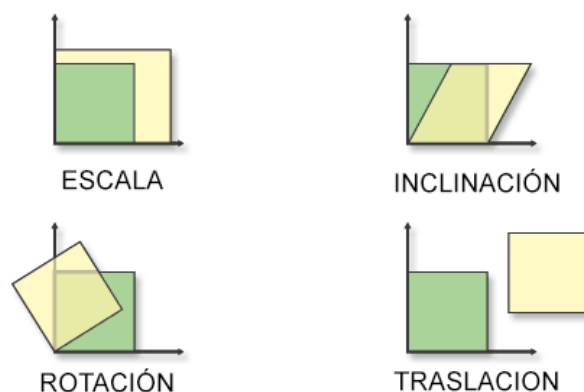


Figura 2-5: Posibles transformaciones afines . Adaptado de [19].

### 2.2.4 Transformaciones no lineales

A diferencia de las transformaciones afines, que son regulares y se pueden describir mediante una matriz, una transformación no lineal es una distorsión irregular de la imagen. Con una transformación de este tipo la relación espacial entre los puntos de interés se pierde, es decir, no todos los puntos del logo se transforman igual de una imagen a otra. Por ello, no es posible realizar una detección en estos casos. Métodos como el de [4] y [5], que usan triángulos de PoIs como identificadores en vez de todos los puntos a la vez pueden añadir cierta robustez a estas transformaciones pero en ningún caso, usando PoIs, se puede ser totalmente robusto a ellas. En realidad ningún descriptor geométrico puede ya que estas transformaciones cambian la estructura geométrica del logo de forma única. Por esto, ninguno de los trabajos analizados defiende ser robusto a transformaciones no lineales.

### 2.2.5 Cambios de iluminación

Un cambio de iluminación es un cambio en el valor (o los valores en el caso de una imagen a color) de una región de píxeles de forma uniforme. Es decir, que la zona del logo aparezca en la imagen más clara o más oscura en general. En ambos casos, el problema no es tanto el cambio total sino una disminución en el contraste del logo si se da el caso de que haya zonas blancas o negras que saturan, haciendo que la distancia con su entorno disminuya. Esto puede desencadenar problemas a la hora de detectar puntos de interés. Aun así, esta situación sólo se produce en cambios de iluminación muy extremos y salvo que en algún paso de preprocesado se hayan establecido restricciones muy fuertes sobre el color del logo a buscar por las que un cambio de iluminación lo mueva fuera del umbral, siempre que los puntos de interés se detecten bien, un cambio de iluminación tampoco debería afectar a una detección basada en PoIs. Ninguno de los trabajos especifica en su descripción o sus pruebas haber tenido problemas con este reto.

Este trabajo, como se menciona con anterioridad, consta de varias etapas en las cuales se extraen datos sobre la imagen que serán combinados con otros o usados para obtener estos otros de manera que poco a poco podamos ir incrementando la confianza sobre si un determinado logotipo o imagen de marca está presente en un fotograma dado o no, y sobre la localización de este. En esta sección se tratará sobre el diseño de las tres componentes de este proceso.

## 3 Diseño

---

### 3.1 Elección del entorno de programación

A pesar de que herramientas como MATLAB podrían ofrecer mayor facilidad de programación, este proyecto se realiza en código C/C++ no orientado a objetos. Se tomó esta decisión principalmente en aras de mejorar la eficiencia temporal de la detección por fotograma, ya que MATLAB, debido a su complejidad y a su entorno de compilación. La razón de no usar clases para este software está basada en el hecho de que es un flujo de datos que se va sometiendo a varios procesados basados en métodos matemáticos. Cada salida es una matriz o un vector de datos, no una estructura determinada.

Al usar C/C++ para usos de visión artificial, la mejor opción que existe es usar la librería de funciones disponible para este lenguaje OpenCV[24], que además tiene la ventaja añadida de ser software de código abierto.

### 3.2 Preprocesado

Es el conjunto de operaciones que se usan para localizar en el fotograma de la secuencia de vídeo la zona o zonas donde es susceptible encontrar el logotipo dado con mayor certeza. La ventaja de la obtención de estas regiones es reducir el análisis posterior y como consecuencia, el costo total del proceso. Además, permite evitar falsos positivos con detecciones en regiones donde no puede haber logotipos. Concentra la mayor parte del código y de la complejidad de este programa, así como la mayoría del gasto temporal que este programa usa en su análisis, ya que es capaz de reducir el tiempo de análisis de la propia detección.

Este preprocesado usa varias técnicas conocidas dentro de la teoría de análisis de video y visión por ordenador como la saliencia, el análisis por valores de color y el análisis de formas y patrones geométricos que compondrán una máscara para cada fotograma donde están activas las zonas donde se realizará la búsqueda del logo.

#### 3.2.1 Detección de zonas fijas

Además, hay una componente de preprocesado que se aplica sobre la secuencia completa, encargada de detectar las regiones estáticas del vídeo (elementos como el marcador y mosca del canal de televisión), y cuya salida será restada de la máscara de preprocesado de cada fotograma para evitar detecciones en estas zonas.

Este proceso se basa en la obtención de un número dado de fotogramas a lo largo de toda la secuencia y localizar las zonas con menor diferencia entre todos ellos.

#### 3.2.2 Componente de saliencia

La saliencia es un concepto muy usado en la visión por ordenador, aunque no hay una definición técnica de ésta y se han presentado una gran cantidad de métodos para obtenerla. El objetivo de un detector de saliencia es localizar las zonas más salientes de una imagen, es decir, las partes de esta que son más llamativas al ojo humano y en las que generalmente, antes se fijará una persona.

Por lo general, un logotipo publicitario en un panel deportivo tiende a intentar ser llamativo y a destacar en una toma general. Por ello, el diseño de este código incluye una componente de detección de saliencia. Particularmente se usa una saliencia a nivel de píxel que será explicada más adelante, aunque hubo otros intentos anteriores que se descartaron.

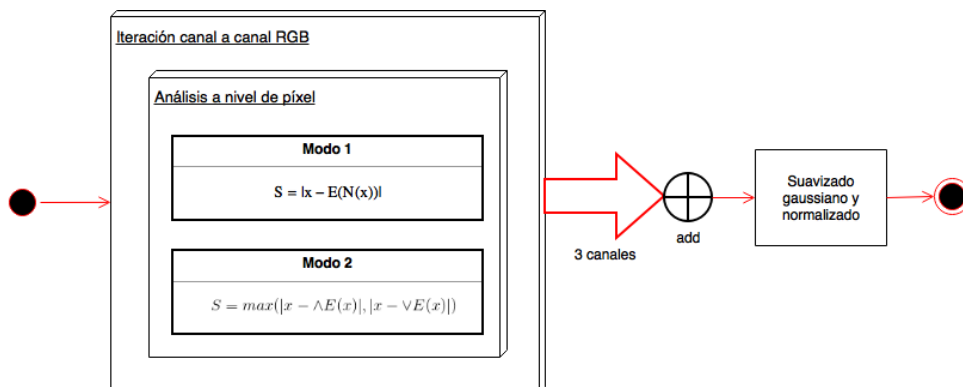


En el código se incluyen varias funciones de saliencia que han sido analizadas a lo largo del desarrollo. Esta primera idea consistía en analizar el fotograma con una ventana deslizante, acumulando en la región de esa ventana en la salida la distancia que había entre su histograma y el de una ventana concéntrica pero mayor. Este diseño está basado en el principio de que si una zona es muy distinta a la zona que la rodea, esa zona se considera saliente.

La otra función de saliencia diseñada para el preprocesado es conceptualmente similar pero sin usar histogramas y sin necesidad de que el valor de un píxel estuviese formado por los valores de un número indeterminado de ventanas a las que pertenece. Este diseño es mucho más intuitivo y directo: calcular la salida de cada píxel comparando únicamente el valor de éste con el de su entorno. Este nuevo código tiene dos modos de funcionamiento; la comparación puede hacerse con la media del entorno, o, el modo más “intenso” (es decir, la salida será más contrastada), con los extremos del entorno, obteniendo la distancia máxima entre el píxel y el máximo o el mínimo de su entorno. Este segundo modo es el que corremos, aunque es un valor configurable.

En el código se realiza el mapa de saliencia sobre los tres canales de color (en este caso usando el espacio RGB, aunque con otros espacios debería funcionar correctamente también) y el resultado final es la norma de los tres, obtenida mediante suma de las matrices y una normalización final. Finalmente se aplica un suavizado gaussiano al resultado para hacer una limpieza de la salida, reduciendo fronteras bruscas y ruido.

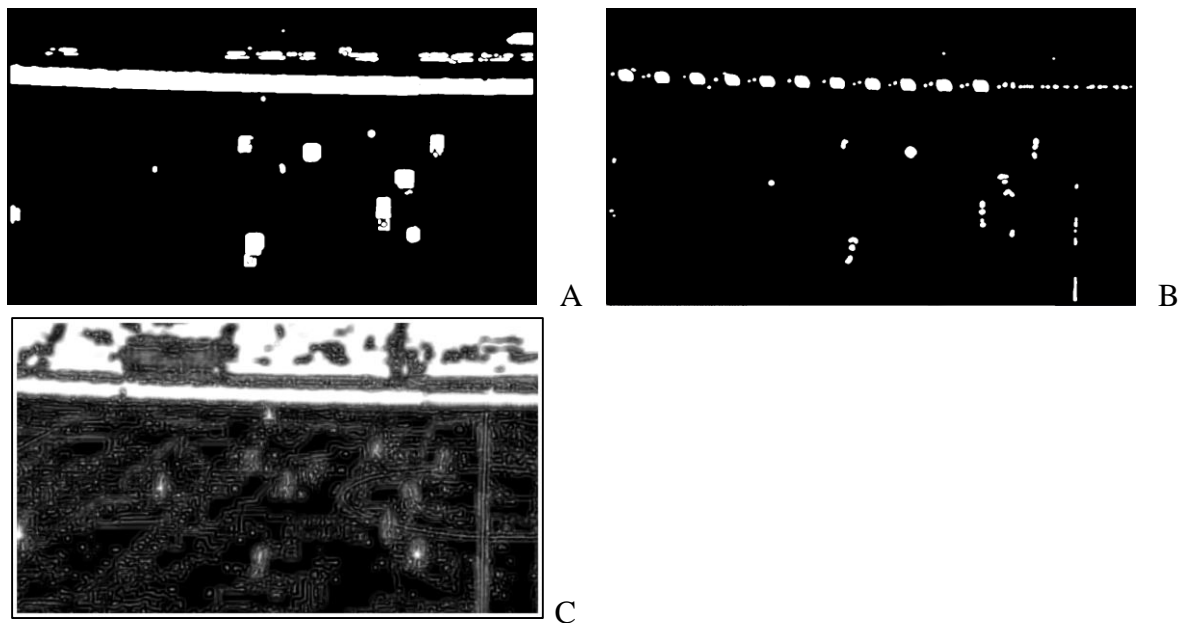
La figura 3-1 muestra el orden de las operaciones que sufren los datos para alcanzar el mapa de saliencia.



**Figura 3-1: Flujo de datos del módulo de saliencia a nivel de píxel**

Finalmente se simplifica la salida umbralizándola para obtener simplemente zonas positivas y negativas con el método de binarización de Otsu[20]

Se pueden comparar los tres métodos de obtención de saliencia en la figura 3-2.



**Figura 3-2: Salidas de las distintas funciones de saliencia. A- Niv. De píxel modo 2. B- Niv. De píxel modo 1. C- Saliencia por histogramas sin umbralizar**

### 3.2.3 Uso de forma y perspectiva

El objetivo de este proyecto es la localización en la imagen de unos patrones muy definidos y específicos, y es que un logo es al fin y al cabo una imagen generada por ordenador, con las ventajas que esto conlleva. Una imagen generada por ordenador, a diferencia de una imagen natural, es una imagen “perfecta” de algún modo, ya que no tiene deformaciones, ni ruido ni problemas de desenfoque (salvo que haya sido introducido a propósito). Un logo, comúnmente está formado por una serie de figuras geométricas muy bien definidas. Además, el entorno de búsqueda, una secuencia deportiva, tiene unas características muy particulares, y la detección del ángulo de la toma puede ser muy beneficiosa en el proceso de detección del logo, ya que con ella es posible obtener datos de la transformación que ha sufrido el logo en el fotograma.

Por el párrafo anterior, se barajó la idea de usar descriptores de forma y color en el código. Esta opción fue descartada porque este tipo de descriptores son útiles para formas geométricas muy sencillas, un logotipo está fuera del rango de funcionamiento de estos métodos. La herramienta a usar en el caso de que se la opción de los descriptores de forma fuese útil con logotipos es la implementación de los momentos de Hu [21] en OpenCV[24]. En el caso del color, será usado pero no mediante descriptores sino a nivel de píxel. Este uso se detalla en la sección 3.2.4.

El uso de formas sí que puede resultar de utilidad para encontrar perspectivas en la imagen. Sabiendo el ángulo de un fotograma, podemos usar ese dato para saber cuál es la transformación que sufre nuestra plantilla y además, como los logos tienen muchas fronteras y contornos horizontales muy marcados, es razonable suponer que un logo debe tener una cantidad de segmentos con la misma inclinación que la perspectiva de la toma, considerando que estos logos están situados en paneles paralelos a los límites del campo, que son la referencia para la detección de la perspectiva.

Por lo citado en el párrafo anterior, una vez se realiza la detección de perspectiva en el fotograma, se almacenan las rectas detectadas en este proceso que son acordes a la perspectiva o perspectivas principales de la imagen (puede haber más de una en tomas de

esquina). Estas rectas almacenadas serán usadas tras el preprocesado en la parte de detección para descartar regiones, de manera que si una región del preprocesado no contiene rectas paralelas al plano de perspectiva, se considera que es una falsa detección y que en ella no será posible encontrar un logo.

El módulo de perspectiva, además de tener como salida el conjunto de rectas detectadas en las perspectivas de la imagen, también devuelve una representación visual de las perspectivas detectadas para poder enseñarla mediante la interfaz gráfica del programa.

### 3.2.4 Componente de color

Es fácil llegar a la conclusión de que una de las características que mejor definen a la zona en la que se encuentra un logo dentro del fotograma es el color de este, ya que suele ser muy característico dentro de la imagen completa. En este diseño se ha optado por usar la plantilla precargada para realizar la búsqueda del logo en el fotograma; otra aproximación sería realizar un preprocesado independiente y usar el logo cargado a la hora de realizar el *matching* con las detecciones.

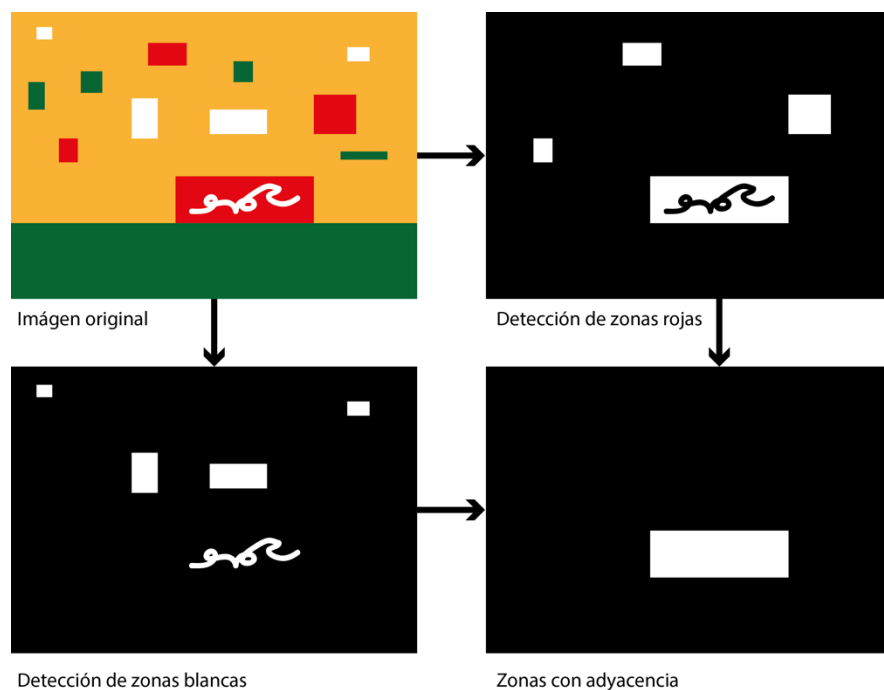
De la misma forma que con la saliencia, la salida esperada de la función de color es un mapa de puntuaciones, siendo más altas las zonas más cercanas al color que buscamos. Luego, el objetivo es obtener la distancia entre los puntos de la imagen y el color del logo, que posteriormente será invertida para convertir distancias en puntuaciones.

Se ha incluido también en el código una adaptación para matrices OpenCV de un código fuente en C++ de la función de distancia de color CIE Delta-E del año 2000 [28].

Finalmente se opta por el uso de una distancia euclídea válida para cualquier espacio de color ya que la aplicación de Delta-E es menos eficiente y el resultado no es mejor.

El módulo de color, por tanto, detecta mediante análisis de histograma la plantilla del logo que está siendo buscado las modas cromáticas de éste, es decir, los picos de color, y obtiene las regiones de la imagen con menor distancia a cada uno de estos colores. Una vez obtenidas estas regiones, se procede a descartar todas las que estén aisladas, es decir, que no tienen ningún pixel en cuyo vecindario se sitúe una región de detectada en los picos de color.

Para explicar la motivación de este proceso, utilicemos un ejemplo de logo a buscar bien reconocido como es el de CocaCola™, cuyos colores son el rojo y el blanco; llegados a este punto contamos con dos máscaras, una con las regiones del fotograma cercanas al rojo y otra con las cercanas al blanco. Las regiones donde el logo esté presente son la unión de una región cercana al rojo que sea adyacente (en el caso del ejemplo de CocaCola™, una que contenga) a otra región cercana al blanco. Luego este paso consiste en la selección para la máscara de salida de las regiones de cada uno de los picos de color que sean adyacentes a otras de otros colores. La figura 3-3 explica este proceso y en la figura 3-4 se puede ver el ejemplo mencionado de CocaCola™:

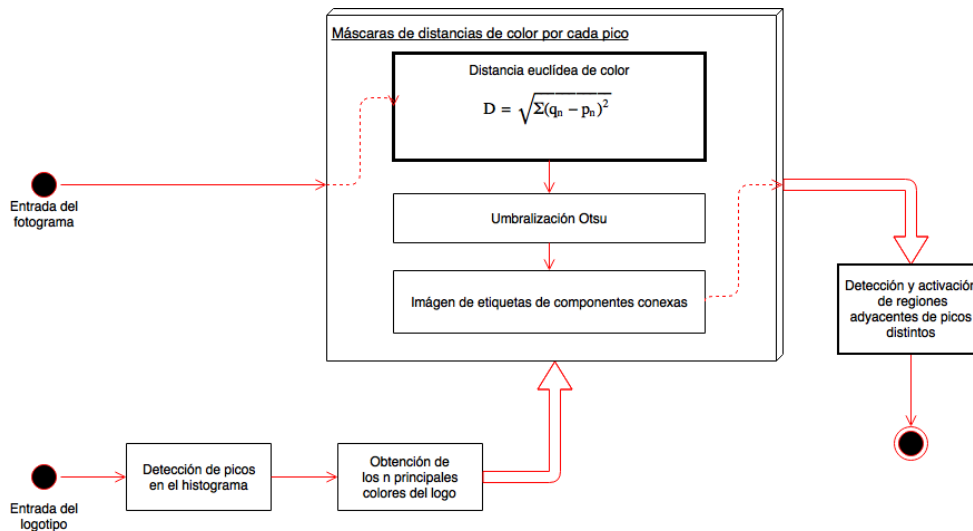


**Figura 3-3: Imagen Sintética. Proceso de selección de regiones adyacentes de color**



**Figura 3-4: Captura del módulo de color. Las imágenes de arriba son regiones con distancias bajas a cada uno de los colores del logo. Abajo se encuentran las regiones adyacentes y la imagen original de referencia.**

El sistema de distancia de color completo está representado de forma esquemática en la figura 3-5.



**Figura 3-5: Flujo de datos en el módulo de distancia de color (función de distancia euclídea).**

### 3.2.5 Unificación para máscara de detección

Una vez obtenidas las máscaras binarias de saliencia y color, es necesario unificar los datos para tener una sola máscara en la que realizar las detecciones. Existen dos aproximaciones posibles para llegar a esta única máscara. Hacer las detecciones de color y saliencia más “exigentes”, es decir, las salidas de estas tienen menor cantidad de positivos, y sumar ambas, o hacer el proceso contrario, hacer que las salidas de cada una de ellas por separado puedan tener varias regiones que no van a ser de utilidad, pero al unificarlas, se obtendrá la máscara final a partir de las regiones que son comunes en ambas.

La opción elegida es la segunda ya que un logo debería ser saliente y por supuesto, aparecer en el proceso de detección de su propio color; de forma que se eliminarán las regiones que no se activen en las dos máscaras a la vez.

Además, a esta máscara final obtenida se le restarán las regiones estáticas detectadas en el preprocesado general de la secuencia de vídeo descrito en la sección 3.2.1 y en mayor detalle, en la sección 4.2.1.

En este momento, el programa ya cuenta con un mapa que contiene la información de dónde ejecutar la detección del logo.

## 3.3 Búsqueda del logo

A partir del fotograma, el template del logo y la máscara obtenida en la sección 3.2, es posible comenzar el proceso de detección del logo mediante el uso de PoIs. Cabe recordar aquí que el hecho de no realizar esta búsqueda por toda la imagen sino usando una máscara obtenida mediante el preprocesado mejora de forma notable la eficiencia del código, a la par que reduce el número de distractores, minimizando la posibilidad de producir falsos positivos.

### 3.3.1 Búsqueda de puntos de interés y *matches*

Antes de la ejecución del bucle que realiza el análisis de cada fotograma de la secuencia, junto con la detección de elementos estáticos en el vídeo, se hace una detección de puntos de interés en la imagen plantilla del logotipo a buscar, de manera que sea posible acceder a estos puntos durante toda la ejecución del programa.

Dentro del bucle, una vez obtenida la máscara de preprocesado, se realiza la búsqueda de puntos de interés usando el mismo método que se ha usado para el logotipo, y por supuesto, usando la máscara obtenida de modo que sólo se detecten puntos en las regiones donde se ha determinado que puede haber logotipos.

Una vez obtenidos los PoIs de las dos partes, se realiza una función de *matching* que devuelve una puntuación y una escala a cada punto del fotograma relacionado con un punto de la plantilla. Aun así, gran parte de estas conexiones siguen siendo falsas detecciones, además, existe la posibilidad de que un mismo logotipo esté presente varias veces en un fotograma (de hecho, es muy común), por lo que el siguiente paso será, a raíz de los *matches* obtenidos, crear una máscara que nos dé información espacial de éstos. En la sección 3.3.2 se explica en detalle este proceso.

### 3.3.2 Información espacial de los *matches*

A partir del fotograma, se crea una imagen vacía del tamaño de éste y a ella se van añadiendo círculos centrados en cada uno de los puntos de interés cuyos radios son proporcionales a la escala de éstos y cuya intensidad depende de la distancia del *match* (la calidad de la detección). El resultado obtenido es una nube de círculos en la que las regiones con intensidades más altas son en las que se acumulan un mayor número de PoIs, y de mayor calidad ya que la intensidad va en función de la frecuencia. Esta máscara proporciona por ello información espacial sobre la situación de los puntos detectados en la imagen, de manera que cada nube conexa de puntos acumulados se trata como una región. Para los pasos siguientes, se procede a almacenar cada una de las regiones aisladas que aparecen en esa máscara, interpretándolas como una detección del logo, que en pasos posteriores será comprobada.

### 3.3.3 Homografía y comprobación

El próximo paso es, a partir de los puntos de interés de cada una de las regiones obtenidas en el punto 3.3.2, tratar de obtener la homografía de la plantilla para ésta región.

Para cada detección, se aplica la transformación afín entre ésta y la plantilla, situando la segunda en la imagen del fotograma.

En este punto, la primera comprobación es el factor de cobertura de la plantilla transformada sobre su rectángulo exterior; si éste es muy bajo significa que la transformación implica una deformación muy alta y por tanto se descarta el *match*. Si ha superado esta comprobación, se busca la MSE (Mean Square Difference, o distancia cuadrática media) entre el *template* transformado y la región equivalente en la imagen. Aceptar o no esta detección depende de un umbral de distancia, de modo que la detección sólo se considere correcta si la distancia no supera el valor dado.

La salida de la localización es el *bounding box* (rectángulo exterior) del área de la plantilla tras la transformación de la homografía.

### 3.4 Interacción con el usuario

De forma relativamente independiente a todo el proceso interno, el usuario tiene la posibilidad de interactuar de varias formas con el software. Por un lado, éste puede introducir mediante parámetro la ruta del archivo que se va a utilizar como entrada de vídeo y la que usará como plantilla del logotipo a buscar, además del umbral que se usará en las detecciones (3.3.3), aunque por otro lado, también se permite al usuario seleccionar el logotipo en el propio video de modo que pueda desplazarse por este y hacer una selección por ratón del polígono que rodea a una vista del logotipo en una fotograma (Figura 3-6). Esta selección en el propio vídeo es ventajosa ya que no hay variación entre los colores del logotipo en el video y los del logo a buscar (en este caso, los mismos). La principal desventaja en cambio, es que la calidad de la imagen del logo en el fotograma puede ser mala, en términos de ruido, distorsión por movimiento e incluso transformaciones no afines. Estos problemas se arrastran a lo largo de todo el proceso pudiendo afectar en gran medida los resultados finales.

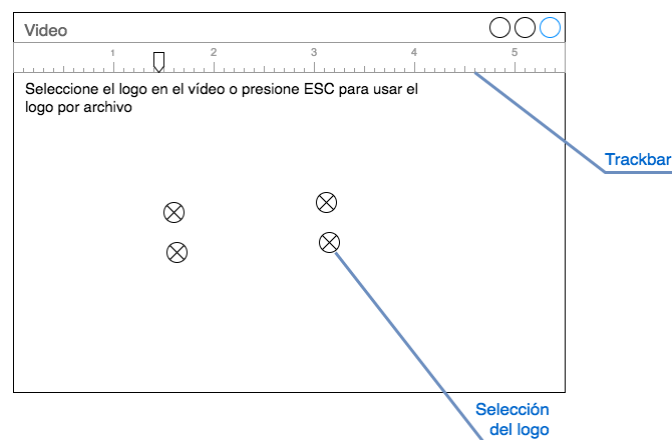
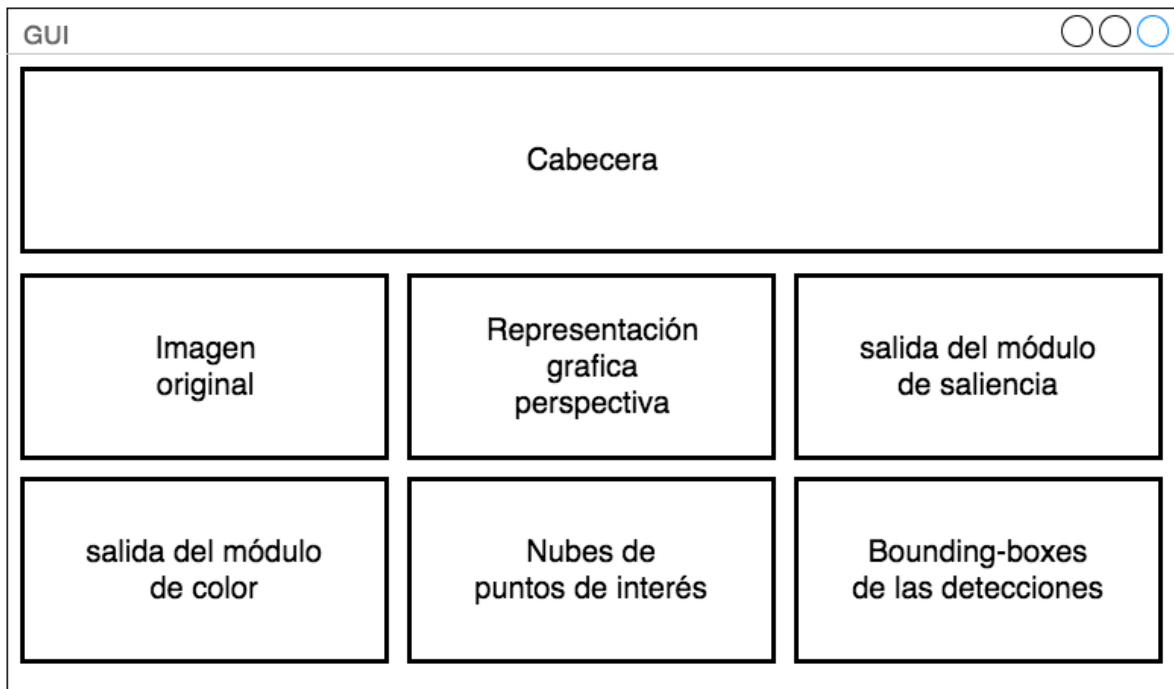


Figura 3-6: Mockup de la ventana de selección del logo dentro de la secuencia.

Por otro lado, aparte de interactuar con la entrada, el usuario también verá una salida por pantalla.

La salida gráfica, por razones de programación, es simplemente un mosaico informativo compuesto por varias matrices obtenidas en distintas etapas del flujo de los datos. Aunque es muy sencillo modificar en el código cuáles son las imágenes que se presentan en esta interfaz, actualmente se muestran la entrada de vídeo, la salida gráfica del módulo de detección de perspectiva (secc. 3.2.3), la máscara binaria de la detección de regiones salientes (3.2.2), la máscara binaria de detección de color (3.2.4), la matriz de puntos de interés que se usará para la obtención de las regiones (3.3.2) y las detecciones finales en forma de *bounding boxes* situadas sobre la imagen de entrada (3.3.3). Además, y como añadido únicamente visual, hay una cabecera de la interfaz que incluye el logotipo de este proyecto junto con el de la universidad, la escuela y el departamento en el que se ha desarrollado el trabajo. En la figura 3-7 se puede ver la organización de esta interfaz.

También como añadido estético, durante el preprocesado general del vídeo y de la plantilla, antes de iniciar el bucle de análisis de cada fotograma, se muestra una pequeña ventana también con los logotipos anteriores y una línea informando al usuario del estado de dicho proceso.



**Figura 3-7: Mockup de interfaz de salida de datos del análisis de fotograma.**



## 4 Desarrollo

---

### 4.1 Trabajo inicial de investigación

#### 4.1.1 Familiarización con conceptos básicos de CV

Ya que la visión por ordenador es un campo tan particular, es necesario familiarizarse con las bases que fundamentan este tipo de programación. En este punto, es necesario asimilar que una imagen es simplemente una matriz bidimensional de números que representan la luminancia de cada punto de ésta. Que las imágenes en color se basan en el mismo concepto con la diferencia de tener tres imágenes en paralelo representando cada uno de los canales. Mediante algo de formación académica anterior a este proyecto y también investigación durante el mismo, ha sido posible utilizar técnicas como detección de PoIs y RoIs, saliencia, bordes de la imagen, filtrados, etc. Imprescindibles para la realización de este trabajo.

#### 4.1.2 Búsqueda y análisis de documentación de trabajos anteriores

El primer paso que se debe realizar antes de iniciar cualquier tipo de proyecto es realizar una investigación a fondo sobre el campo en el que se desarrollará dicho proyecto, familiarizándose con las posibles técnicas y los logros que ya se han alcanzado en dicho ámbito. La documentación existente sobre este campo de estudio es bastante abundante, y la facilidad de acceso a documentación científica hoy en día es un gran aliciente para realizar una investigación a fondo.

Por todo esto, no ha sido difícil encontrar inspiración para la realización de este proyecto, si bien es cierto que no todos los trabajos presentaban unos resultados demasiado favorables para el caso que se presenta aquí, esto también puede ser considerado para en el futuro descartar posibles técnicas en la detección.

Mediante la búsqueda en fuentes de artículos de investigación como IEEE [22] o Springer[23], se han analizado técnicas de detección de logos y marcas, tanto en imágenes fijas, como en transmisiones de vídeo, también deportivas en algunos casos, de laboratorios de investigación de universidades y empresas privadas de todos los continentes. Pudiendo concluir que entre los años 2005 y 2013 se ha trabajado mucho en este campo, haciendo interesantes avances y utilizando una amplia variedad de técnicas.

A raíz de la investigación se ha determinado que una técnica que es prácticamente imposible dejar de utilizar es la detección y comparación de puntos de interés, ya que ésta se aplica en prácticamente todos los trabajos de detección automática.

Esta investigación también ha sido útil para comprobar que en trabajos anteriores no es común realizar un preprocesado de la imagen reduciendo de manera intensa el campo de búsqueda y por tanto aumentando la velocidad de análisis.

#### 4.1.3 Elección del lenguaje de programación y herramientas a usar

El mundo de la visión artificial es un caso muy especial dentro de la programación, muy relacionado con operaciones matemáticas de grandes cantidades de datos dentro de matrices, ya que una imagen digital es al fin y al cabo una matriz de números (o ternas de números en el caso de una imagen a color).

Una herramienta de uso muy común en el procesado de imagen, por lo mencionado anteriormente, es MATLAB; un lenguaje sencillo de usar, robusto e intuitivo y con una gran carga de operaciones matemáticas y operaciones específicas sobre imágenes, entre las cuales se incluyen algunas utilidades de interfaz gráfica realmente fáciles de usar. La razón por la

que se descartó el uso de MATLAB es que, al ser análisis de video; lo que conlleva analizar, considerando que no se descartan fotogramas, entre 25 y 30 imágenes cada segundo; la eficiencia del software cobra un papel muy importante, y MATLAB, en parte debido precisamente a esa robustez y complejidad interna, no destaca por esa cualidad.

La mejor alternativa a la variedad y calidad que ofrece MATLAB en análisis de imagen es la librería pública OpenCV[24]. Esta librería lleva años de desarrollo de la comunidad y es realmente potente aunque a la par, también algo más difícil de usar. El aspecto más positivo de OpenCV[24] es que está disponible para su uso en los lenguajes de programación C/C++ y Python (también hay algunas implementaciones en Java).

Descartado MATLAB, existe la opción del uso de Python, pero tras un análisis y un pequeño comienzo de desarrollo en esta plataforma, se puede llegar a la conclusión de que hay disponibles muchos menos recursos en comparación con C/C++ y finalmente se optó por el uso de éste, que a la par es el que mejor eficiencia ofrece.

## **4.2 Generación del código**

El código del programa se divide en dos apartados considerando su ejecución: uno es el código que se ejecuta una sola vez para el vídeo y otro es el que se ejecuta para la detección en cada fotograma dentro de un bucle. En cada una de las subsecciones de este apartado se precisa en qué punto se ejecuta cada función.

Al comienzo del programa se analizan los parámetros de entrada, y en caso de no existir, se utilizan valores por defecto. Una vez se cuenta con los datos de entrada se abren los archivos de vídeo y plantilla y se realizan sobre ambos los análisis previos necesarios; detección de zonas estáticas en el vídeo y análisis de puntos de interés para el logo.

Si está activado, antes del análisis de los datos de entrada, el programa permite enseñar una ventana al usuario donde éste, mediante cuatro clics sobre la imagen pueda seleccionar el logotipo a buscar en el propio vídeo en vez de usar una imagen precargada de archivo. Si se realiza la selección, una bandera se activará y alterará el comportamiento del código siempre que se tenga que usar la plantilla para usar la selección sobre la imagen en vez de la imagen de archivo.

Tras esto, comienza un bucle que va obteniendo fotogramas de la captura de vídeo de forma parametrizable, ya que el usuario puede establecer un límite superior de fotogramas y un salto, de manera que, en vez de analizar cada fotograma, el software analice por ejemplo uno de cada cinco.

Dentro del código también hay una bandera que permite activar o desactivar el volcado de los datos a archivo.

En la figura 4-1 se detalla todo el flujo de datos desde la entrada de archivos a la salida de los datos procesados.

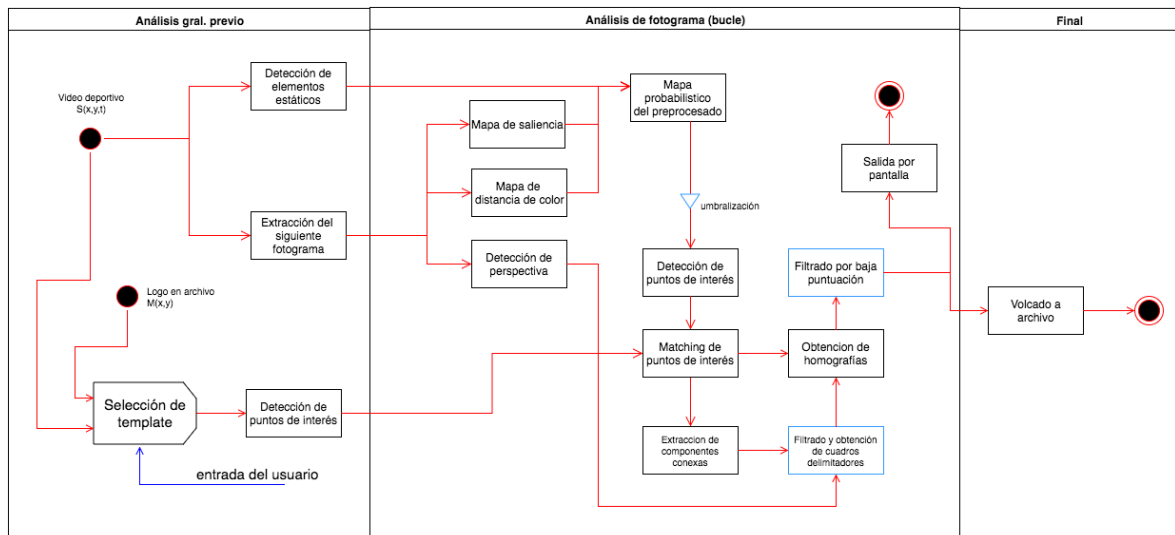


Figura 4-1: Diagrama de flujo de datos general del sistema.

#### 4.2.1 Algoritmo de detección de zonas fijas

Durante el procesado previo del vídeo, antes de comenzar el bucle de la detección de logos, se crea una máscara donde las zonas que están activas son aquellas que permanecen constantes en el vídeo. Es necesario localizar estas regiones para no tenerlas en cuenta en la detección ya que van a acumular muchos puntos de interés.

El algoritmo usado consiste en un bucle que procesa la *accumulated frame difference*, es decir, las diferencias acumuladas entre los fotogramas que se están procesando. En las zonas que son estables y no cambian a lo largo del tiempo, la diferencia entre unos fotogramas y otros es cercana a cero. Al acumular estas diferencias, la posibilidad de error se reduce cuantos más fotogramas se consideren. En este diseño se toma la decisión de usar 20 fotogramas equidistantes entre el comienzo y el final de la secuencia.

#### 4.2.2 Algoritmo de detección de perspectiva

En este modo, el primer paso es sacar la imagen de bordes del fotograma. Esta se consigue mediante el método de Canny[29]. A partir de este resultado se aplica el detector de rectas de Hough[30], y se analizan todas las rectas encontradas para obtener sus pendientes.

Una vez con los datos de pendiente de todas las rectas encontradas, se recorren en bucle para obtener las pendientes más frecuentes. Estas pendientes serán almacenadas como los ángulos de la perspectiva de cada fotograma.

Este modo devuelve una imagen sintetizada que representa todos los ángulos obtenidos (ver figura 4-5) y un vector con las localizaciones de las rectas cuya pendiente es alguna de las modas, que serán usadas posteriormente para la comprobación de las detecciones de logos.

#### 4.2.3 Algoritmo de detección de saliencia

El algoritmo de saliencia usado en el código ha cambiado a lo largo del desarrollo, siendo finalmente usado uno programado específicamente que detecta la saliencia a nivel de píxel.

El proceso se basa primero en obtener el valor del píxel y compararlo con el máximo y el mínimo valor de un entorno de 8-conectividad; la salida es el máximo de esas dos distancias. Este algoritmo también se puede configurar para que la salida sea simplemente la distancia entre el valor del píxel y el valor medio del entorno. En esta segunda opción, la salida es menos contrastada.

Una vez obtenido el mapa de saliencia, se aplica un umbralizado a la imagen mediante el método de Otsu[20].

#### 4.2.4 Algoritmo de distancia de color

El primer paso para la obtención de las regiones cercanas a los colores del logo que se está detectando es precisamente obtener los valores de estos colores. Esto se realiza obteniendo los picos en el histograma de la plantilla en escala de grises y a partir de éstos, se divide el histograma completo en secciones con centros cada uno de los picos.

Con los intervalos de estas secciones se usan máscaras sobre las que se obtiene el valor medio en la imagen de logo a color. Estas medias son los valores de colores a buscar en el fotograma.

Existen dos algoritmos de distancia de color en el código de este proyecto. Uno de ellos comienza con una conversión del fotograma de BGR al espacio de color CIE  $L^*a^*b$ [25], una vez realizada la conversión, se ha creado un código para usar la función de distancia de color Delta-E de CIE2000[26] adaptada a las matrices que usa OpenCV[24]. Esta función resulta bastante costosa computacionalmente, por lo que en la versión final se utiliza la alternativa de distancia euclídea, que píxel a píxel calcula la distancia entre los tres valores en BGR de la imagen con los valores del color a buscar. Ambas salidas se aprecian en la figura 4-2.



**Figura 4-2: Distancias de color a blanco (izda. Delta-E, dcha. Euclídea) para la detección del logo de CocaCola™ en la imagen de abajo.**

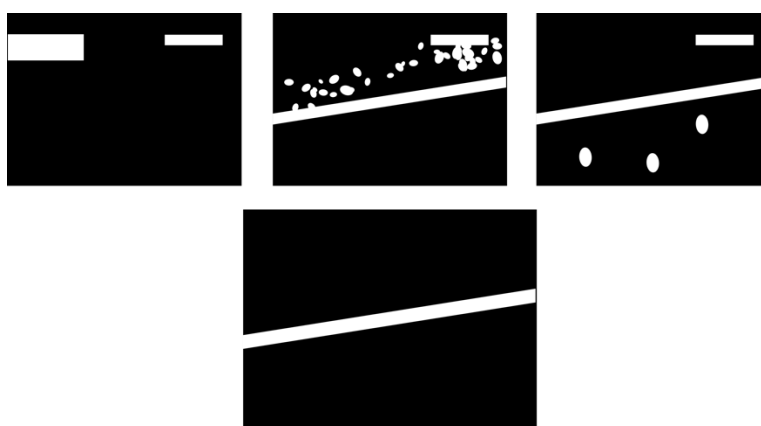
Una vez realizada la búsqueda de los colores, se cuenta con un mapa de distancias por cada uno de los colores del logo, que han sido ya umbralizados por el método de Otsu[20]. Es necesario ahora obtener sólo las regiones que contienen todos los colores del logo. Para ello, se realiza un mapa de etiquetas para todas las regiones de todos los mapas, empezando por las etiquetas del primero y añadiendo las regiones de los siguientes etiquetadas a partir de la mayor etiqueta usada hasta el momento.

Con la información de en qué rangos están las etiquetas, se analiza píxel a píxel buscando si alguno tiene una región adyacente que pertenezca a un rango distinto de etiquetas, es decir,

a un color distinto del logo, respecto del de su propia etiqueta. Si es así, ambas regiones adyacentes se activan en la máscara final.

#### 4.2.5 Máscara del preprocesado

En este punto del código, ya contamos con tres máscaras binarias con los datos de saliencia, color y elementos estáticos. La detección de puntos de interés para obtener logos se aplica sobre las zonas salientes, cercanas en color y no estáticas, por lo que primero se realiza una operación de multiplicación lógica (AND) entre las máscaras de color y saliencia, de modo que sólo permanezcan activas las zonas activas en ambas máscaras. Al resultado se le aplica una resta lógica de la máscara de elementos estáticos, de modo que aunque esas regiones fuesen salientes y cercanas en color, si son estáticas estarán desactivadas en la salida. Este proceso se puede ver fácilmente con el ejemplo de la figura 4-3



**Figura 4-3: Imagen sintética para visualizar la obtención de la máscara final según la explicación en 4.2.5. siendo las imágenes por orden la máscara de elementos estáticos, la de saliencia, la de color y abajo la final de preprocesado.**

#### 4.2.6 Detección y *matching* de Pols

Es necesario contar con puntos de interés tanto en el fotograma como en la plantilla. Ya que la plantilla permanece inalterable a lo largo de la ejecución, una vez se obtiene (ya sea mediante carga de archivo o mediante selección en el propio vídeo), se analiza en busca de puntos de interés, que son descritos mediante el descriptor SIFT[12], cuya implementación está disponible en OpenCV[24], que permitirá obtener *matches* más adelante.

Ya dentro del bucle de análisis de la secuencia, es necesario realizar el mismo proceso en la imagen del vídeo donde se pretende detectar el logo. La búsqueda de PoIs no se realiza como ya se ha mencionado sobre el fotograma completo sino sobre las zonas donde el preprocesado ha concluido que es probable encontrar logotipos, usando la máscara descrita en la sección 4.2.5.

Una vez se cuenta con los dos conjuntos de PoIs, este código utiliza el método de enlace de PoIs Fast Library for Approximate Nearest Neighbors [27] (FlannMatcher en OpenCV[24]). Este devuelve pares de puntos del logo y del fotograma junto con una escala de cada punto en el fotograma y una distancia inversamente proporcional a la calidad de la detección de cada correspondencia encontrada.

Con todas esas correspondencias, se obtiene una máscara donde los puntos del fotograma que son etiquetados como *matches* por [27] se representan con un círculo centrado en el punto, de radio la escala de detección y de luminosidad la distancia de la correspondencia. Ésta máscara se binariza por el método de Otsu[20] y a partir de esta binarización se obtienen las componentes conexas de la máscara, es decir, las regiones que tienen puntos acumulados. Cada una de estas regiones va a ser considerada una posible detección del logotipo.

En la figura 4-4 se pueden apreciar las nubes de puntos de interés en esta máscara antes del umbralizado y la detección de regiones conexas.



**Figura 4-4: Máscara de puntos de interés en una detección del logo de William Hill.**

#### 4.2.7 Validación de la detección por región

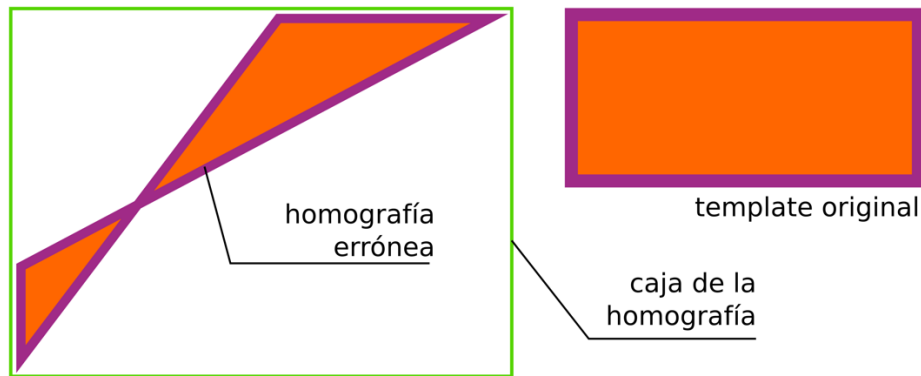
Como primer método de evitar falsas detecciones, de estas regiones se descartan aquellas que no contengan en su interior ninguna recta del ángulo de la perspectiva de la imagen, obtenidas en el algoritmo descrito en la sección 4.2.2.

De las regiones restantes, se obtiene a partir de los puntos de interés pertenecientes exclusivamente a cada región y sus equivalentes en la plantilla por el *match*, la homografía que define la transformación afín entre la plantilla y la región detectada mediante RANSAC[16]; siempre que existan POIs suficientes y que éstos sean coherentes.

Con el valor de la matriz de homografía, se aplica la transformación descrita en ésta a la plantilla del logo, de modo que se obtiene una imagen donde la plantilla aparece teóricamente en la zona del fotograma donde se ha detectado y colocado en la misma perspectiva que en la detección.

Llegados a este punto, se realizan dos pasos para determinar si la homografía es correcta. El primero es una comprobación de la compacidad del rectángulo que contiene a la imagen transformada. Si ésta representa una parte menor del 30% del área del *bounding box* se descarta la homografía puesto que se considera que la transformación no es coherente. Este método se ilustra en la figura 4-5.

Además, se comprueba también si el rectángulo ocupa más del 90% de la imagen o menos del 0.5%, en cuyo caso también se descarta, puesto que se consideran fallos en la extracción de las componentes conexas que definen las regiones susceptibles de contener logotipos.



**Figura 4-5: Explicación sintética del rechazo por baja cobertura con un ejemplo de homografía errónea.**

El segundo paso es la obtención de la distancia media entre los puntos del logotipo situado en el fotograma y los puntos equivalentes en el propio fotograma. Cuanto más baja sea la distancia, mayor es la certeza de que ese logo realmente encaja en esa región. El valor de distancia a partir del cual se descartan *matchings* se deja como parámetro de entrada del código.

Se realizará un estudio sobre el umbral de esta distancia en la sección 5.3.

#### 4.2.8 Mecanismos de interacción con el usuario



**Figura 4-6: Captura de la pantalla de carga del programa que aparece al comienzo de la ejecución.**

Además de la ventana meramente informativa de la figura 4-6, que aparece el iniciar el programa durante el preprocesado del código, el usuario tiene cinco maneras de interactuar con este código:

##### 4.2.8.1 Entrada de parámetros

Este código admite entrada de argumentos por la llamada desde un terminal. Los argumentos no son necesarios y si no se encuentran se utilizan valores precargados, si un argumento no

está presente no se procederá a buscar el siguiente; dicho de otro modo, los argumentos tienen un orden y para introducir el argumento 5, por ejemplo, es necesario haber introducido los cuatro anteriores.

#### 4.2.8.2 Selección del logotipo por interfaz

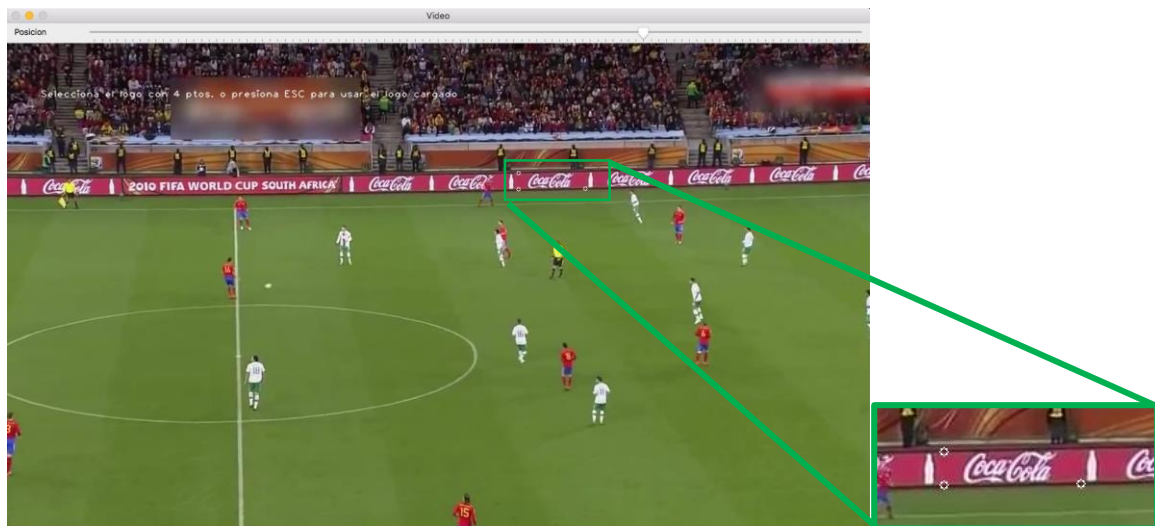
Este software permite activar por código la selección del logotipo a buscar (lo que se ha denominado plantilla en esta memoria).

Si esta opción está activada, tras ejecutar el programa aparecerá una ventana del sistema que mostrará el primer fotograma de la secuencia y el mensaje "Selecciona el logo con 4 ptos. o presiona ESC para usar el logo cargado". Cada clic en la imagen representa un punto y al llegar a cuatro se crea una máscara sobre esa imagen que será usada como logo.

Además, se ha añadido una barra de desplazamiento a la interfaz para poder seleccionar el logo en cualquier momento del vídeo.

Si se presiona la tecla escape, el programa usará como plantilla el archivo precargado.

Esta interfaz aparece en la figura 4-7.



**Figura 4-7: Captura de la pantalla de la pantalla de selección de logotipo estando seleccionados tres de los 4 puntos necesarios.**

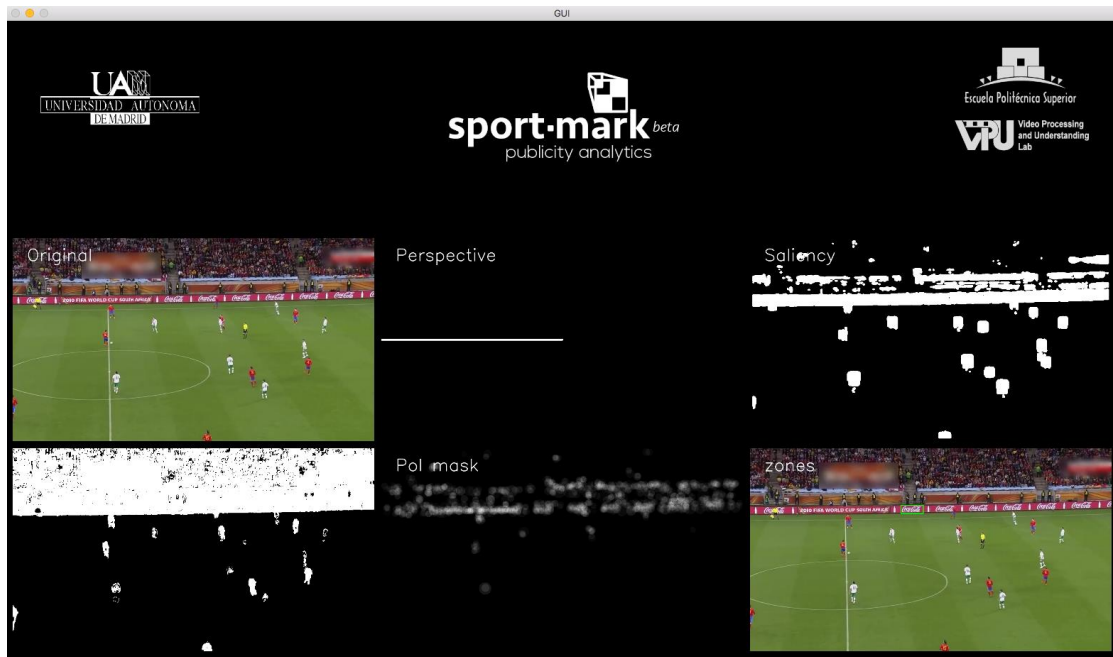
#### 4.2.8.3 Salida de datos por consola

En la versión final, la principal función de la consola es demostrar que el código se está ejecutando. Hay algunos mensajes para el usuario meramente informativos sin valor técnico que se muestran por éste medio.

#### 4.2.8.4 Salida visual por interfaz gráfica

Este código está preparado para mostrar varias salidas gráficas en una sola ventana. Según la configuración actual, la ventana muestra la siguiente información por cada fotograma analizado de la secuencia de vídeo (figura 4-8):





**Figura 4-8:** Captura de la pantalla de análisis de fotograma donde se pueden ver por orden: El fotograma original, la salida del módulo de detección de perspectiva, la máscara de saliencia, la máscara de color, la máscara de nubes de POIs y finalmente las detecciones de logos.

No hay ningún tipo de interacción disponible a tiempo real para que el usuario actúe sobre esta ventana salvo la opción de terminar manualmente el programa.

#### 4.2.8.5 Volcado de datos a archivo

También mediante un *flag* activado por código, es posible hacer que el programa durante su ejecución guarde en un archivo, cuyo nombre se crea en función del nombre de los archivos de entrada, los datos de los *bounding boxes* de las detecciones y el valor de la distancia (descrita en la sección 4.2.7) de cada una de ellas.

Adicionalmente, es posible activar la salida de las imágenes con las detecciones marcadas en los fotogramas en verde (Figura 4.9).



**Figura 4-9:** Salida de archivo de una detección en un fotograma estando activada esta función.

## 5 Integración, pruebas y resultados

Se han elegido varias secuencias de vídeo para evaluar la bondad del comportamiento del código resultante de este trabajo.

A lo largo de la codificación y el desarrollo, las pruebas se han realizado exclusivamente con la entrada al algoritmo de transmisiones de partidos de fútbol. Estas transmisiones tienen características muy particulares con respecto a otros deportes, ya que cuentan con planos muy abiertos y la mayor parte del tiempo, los planos son al centro del campo, por lo que los logos de los paneles del perímetro del campo están poco deformados.

Estas pruebas tratan de evaluar principalmente dos factores:

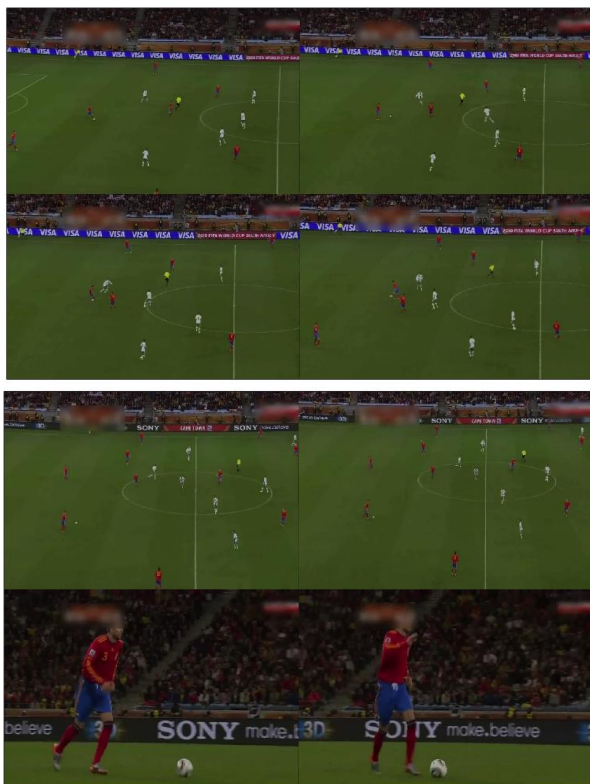
- Precisión y *recall* de la detección (En función del umbral de detección utilizado)
- Visibilidad temporal del logo en una secuencia completa

### 5.1 Procedimiento para pruebas

El primer paso es una selección de secuencias para realizar pruebas con ellas. El cometido de esta sección es poner el código a prueba, por lo que carece de sentido usar imágenes que se han usado durante su desarrollo.

Con cada secuencia utilizada se realizan anotaciones manuales de los logos a detectar que realmente aparecen y se corre el código de detección automática de manera que almacene en un archivo los datos y capturas de las detecciones, incluyendo la distancia de cada detección, dato que permitirá concluir cuál es el umbral más favorable para descartar falsos positivos.

Aquí se especifican las secuencias utilizadas y los retos específicos que suponen para el código.



#### 5.1.1 Secuencia 1

(Fútbol) Encuentro España-Portugal Euro 2014. Resolución XGA. Logo a buscar: SONY™

Elegido por contener primeros planos del logotipo y oclusiones de éste por un jugador.

#### 5.1.2 Secuencia 2

(Fútbol) Encuentro España-Portugal Euro 2014. Resolución XGA. Logo a buscar: Visa™

Ejemplo sencillo de logo único y bien definido. El reto más claro aquí es la repetición del logo y la posible unión de regiones



### 5.1.3 Secuencia 3

(Fútbol) Encuentro Inglaterra-Brasil. 6 de febrero de 2013. Resolución WXGA. Logo a buscar: williamhill.com

El principal reto de esta secuencia es que, a diferencia del resto de secuencias de fútbol utilizadas, la publicidad perimetral es animada.



### 5.1.4 Secuencia 4

(Badminton) Lin Dan vs Lee Chong Wei 2010 Asian Games MS Final. Resolución WXGA. Logo a buscar: China Telecom™

Elegido por el reto de que en esta secuencia, el color no permite distinguir los distintos logos que todos han sido adaptados al azul y blanco.



### 5.1.5 Secuencia 5

(Hockey Hielo) Encuentro Linköping HC vs Växjö Lakers de la SHL. 17/03/2016. Resolución SVGA. Logo a buscar: Ricoh™

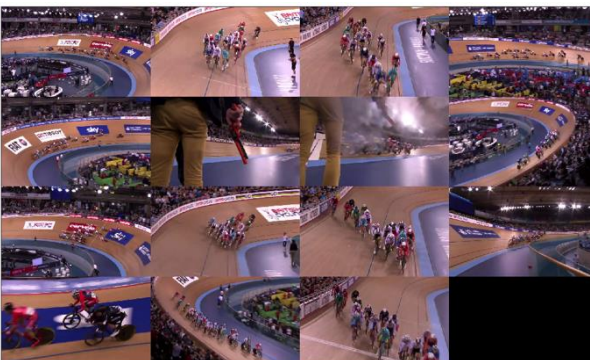
Elegido por intenso movimiento de cámara, planos cercanos, abundancia de zonas claras y definición baja.



### 5.1.6 Secuencia 6

(Waterpolo) Encuentro Serbia-Montenegro Euro 2016. Resolución WXGA. Logo a buscar: Arena™

Elegido por la variabilidad de tonos (fuerte saliencia) y cambios de perspectiva.



### 5.1.7 Secuencia 7

(Ciclismo de pista) Carrera de eliminación 4/3/2016. Mundial UCI Londres. Resolución WXGA. Logo a buscar: Cervelo™

Muy rápido movimiento de cámara y soporte no planar del logo.

### 5.1.8 Secuencia 8 (prueba detección temporal)



(Fútbol) Encuentro España-Portugal Euro 2014. Calidad HD720. Logo a buscar: Hyundai™

Video de larga duración donde se analizarán en que fotogramas hay detecciones y en qué periodos de tiempo está el logo visible.

## 5.2 Tabla de resultados

Para esta tabla, se considera que un solape de más del 50% entre la región real del logo y la detección automática es una detección positiva, si el solape es menor, ésta es errónea y cada bloque anotado existente sin solape superior al 50% con una detección se considera un bloque perdido.

Las pruebas 1, 2, 3 y 8 son secuencias de partidos de fútbol. Éste es el caso para el que el algoritmo ha sido diseñado y el que se ha usado para comprobar el correcto funcionamiento de éste a lo largo del desarrollo. Se espera que los resultados en éstas sean mejores que para el resto de secuencias.

Por el contrario, las secuencias 4, 5, 6 y 7 son vídeos de otros deportes que han sido utilizados en este proceso de testeo en aras de comprobar la bondad del algoritmo para situaciones distintas al fútbol. En la sección 5.1 se explican en detalle los retos que suponen cada una de las secuencias.

En la tabla 5-1, los valores que aparecen tienen el siguiente significado:

- TP: *True Positive*  
Aquellas detecciones que se consideran correctas.
- FP: *False Positive*  
Falsos positivos, detecciones que no corresponden con una aparición del logo a buscar.
- FN: *False negative*  
Apariciones del logo que se está buscando que no han producido una detección en el sistema.
- Precisión:  
Cociente entre el número de detecciones correctas entre el número total.

$$P = \frac{TP}{TP + FP}$$

- Recall  
Cociente entre el número de detecciones correctas del logo y el número de logos que aparecen.

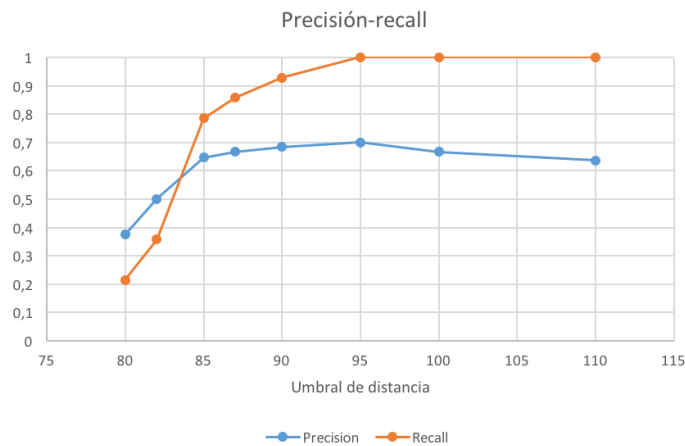
$$R = \frac{TP}{TP + FN}$$

El contenido de las casillas de detecciones son las distancias de estas introducidas por su recuento.

Secuencia (Frames)	TP (anotaciones detectadas)	FP (detecciones no anotadas)	FN (anotaciones no detectadas)	Precisión x100	Recall x100
1(4)	4 – 52, 52, 47, 59	3 – 99, 97, 46	1	57%	80%
2(4)	15 - 44, 42, 40, 46, 42, 47, 47, 45, 48, 43, 45, 48, 43, 45, 53	0	24	100%	38%
3(16)	14 -82, 82, 83, 88, 86, 90, 77, 76, 84, 81, 84, 80, 82, 77	8 - 106, 96, 57, 83, 70, 77, 70, 67	0	64%	100%
4(15)	0	4 – 45, 54, 46, 123	10	0%	0%
5(15)	0	5 - 123, 113, 118, 109, 78	5	0%	0%
6(10)	0	4 – 148, 151, 138, 98	13	0%	0%
7(15)	0	6 – 110, 121, 89, 121, 122, 130	14	0%	0%

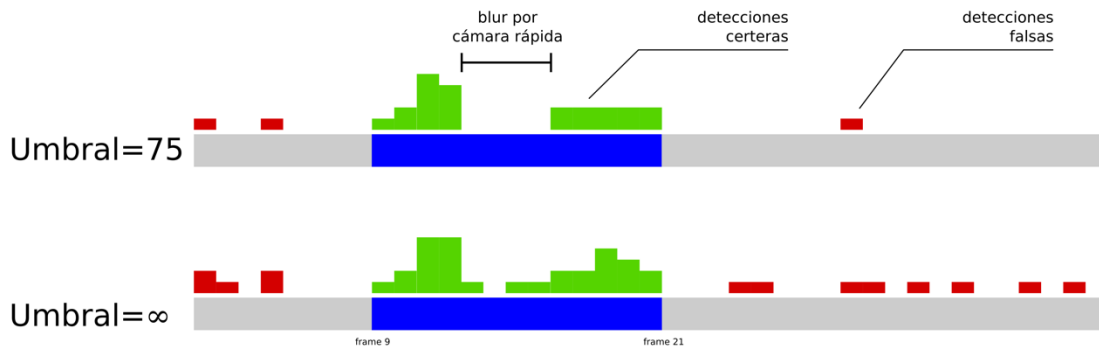
**Figura 5-1: Tabla de los resultados de las distintas pruebas. En las columnas TP y FP aparece el número total seguido de las distancias en cursiva.**

La tabla 5-1 muestra también las distancias de todas las detecciones, permitiendo observar la diferencia entre las distancias de las anotaciones erróneas y las de las anotaciones correctas.



**Figura 5-2: Diagrama de precisión-recall en función del umbral para la secuencia 3.**

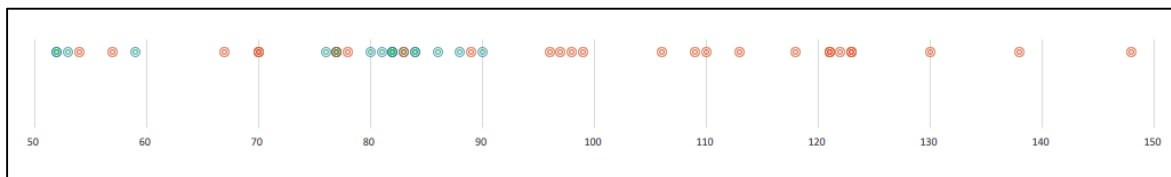
En la figura 5-2 se aprecia como la aplicación de un umbral para las distancias de las homografías mejora la precisión del sistema reduciendo falsas detecciones. En esta entrada en particular, el umbral óptimo se sitúa en 95.



**Figura 5-3: Línea de tiempo para detecciones en una secuencia.**

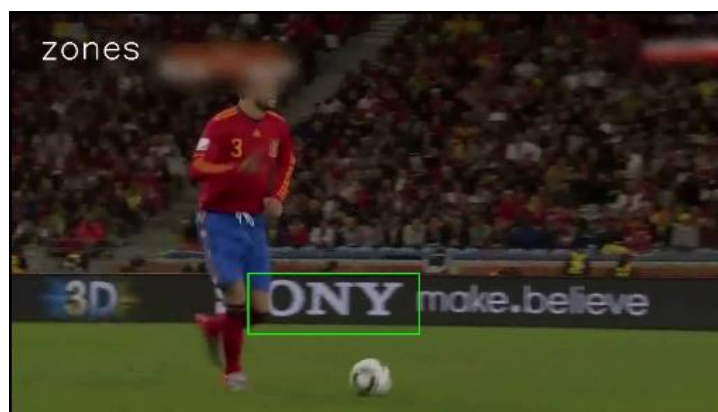
La figura 5-3 muestra los resultados de la prueba de detección temporal. En ella, la altura de las barras verdes y rojas representan el número de matches en cada fotograma. También se muestra la diferencia de los resultados usando un umbral óptimo para esta entrada.

### 5.3 Análisis de los resultados



**Figura 5-4: Distribución de las distancias de los matches (rojo - FPs, verde - TPs). Las distintas intensidades se deben a superposición.**

En el gráfico de la figura 5-2 se aprecia como todas las detecciones correctas se sitúan entre una distancia de 50 y una de 90, mientras que gran parte de las falsas detecciones tienen una distancia mayor a 90.



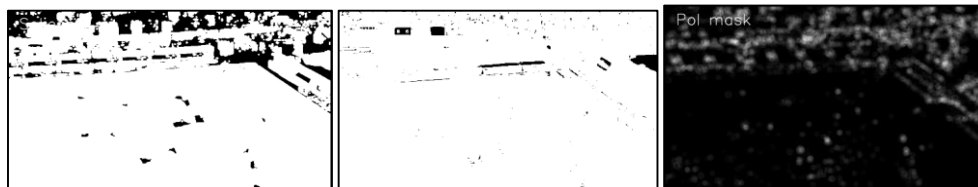
**Figura 5-5: Captura de detección con oclusión**

Los resultados en las secuencias de fútbol 1, 2 y 3 son bastante favorables, resistentes a fuertes cambios de perspectiva y oclusiones (figura 5-3).

La razón de que en la secuencia 2, donde se detecta el logo de Visa™ tenga un número tan alto de logos perdidos es que no hay separación entre un logo y otro, por lo que las regiones tienden a solaparse y dos logos forman parte de la misma. Cuando esto ocurre la homografía suele no ser coherente y se descarta.

Con respecto a las pruebas para otros deportes, los resultados no han sido nada positivos. Aunque estas pruebas son muy útiles para ver las carencias y defectos de este sistema y en qué circunstancias su comportamiento es incorrecto. Se procede a continuación a analizar cada uno de éstos casos de prueba:

- Secuencia 4: Bádminton  
El principal problema en éste entorno, y esto también sucederá en deportes similares como el tenis, es que todos los logos se implantan en la misma gama de colores y están muy cercanos, por lo que la máscara de puntos es incapaz de distinguir la región de un logo de las de los logos o elementos cercanos.
- Secuencia 5: Hockey  
El principal problema de ésta secuencia es la baja definición, añadida a unos movimientos de cámara muy rápidos que producen un marcado desenfoque de movimiento. Los puntos de interés detectados en ésta secuencia son poco certeros. Además, la máscara de saliencia y color tiene un resultado menos eficaz que en el futbol debido a que el fondo de los paneles de publicidad es blanco al igual que el terreno de juego.
- Secuencia 6: Waterpolo



**Figura 5-6: Máscaras de saliencia, color y PoIs de un fotograma de la secuencia 6**

Como se puede apreciar en la figura 5-4, dada la luminosidad, variabilidad y dinamismo de los colores de las tomas de vídeo de waterpolo, el preprocesado es prácticamente inútil, activando la imagen completa. Esto hace además que haya muchos puntos de interés y por ello, sea muy complicado aislar los logotipos.

- Secuencia 7: Ciclismo de pista  
En el caso del ciclismo, las dos causas principales del mal resultado son el intenso desenfoque de movimiento, al igual que en la secuencia 5, junto con el hecho de que los logos están implantados en superficies curvas, y por tanto la transformación geométrica del logo no puede describirse por medio de una transformación afín del logo original.

Finalmente, este código no sólo está diseñado para buscar un logo dentro de un fotograma dado pero para permitir hacer un análisis temporal de la cantidad de apariciones de un logo dado en una secuencia de vídeo completa.

Para esta prueba se ha usado de nuevo una secuencia de fútbol, y el logo que se desea encontrar es el de Hyundai™.

En la figura 5-3 se puede comprobar la bondad de nuestro algoritmo en una secuencia completa de vídeo. Sólo durante la sección azul el logotipo de Hyundai™ está presente en las marquesinas de publicidad del campo. Aplicando un umbral de valor 75 se obtiene una media de 1,7 detecciones del logo por fotograma en la sección donde aparece el logo y de 0,14 en la sección donde no aparece (falsas detecciones). Incluso sin aplicar ningún umbral la cantidad de detecciones erróneas es marcadamente baja como se aprecia en la figura 5-3, donde puede verse el efecto de aplicar un umbral (eliminando 10 de las 13 detecciones falsas y sólo 7 de 29 certeras).

Una vez más, como ocurría en las pruebas de hockey hielo y ciclismo en pista, se aprecia claramente en el gráfico los efectos que tiene el movimiento rápido de la cámara sobre las detecciones de nuestro algoritmo.



## 6 Conclusiones y trabajo futuro

---

### 6.1 Conclusiones

Se ha logrado estudiar con profundidad las técnicas que se han usado en una gran variedad de aproximaciones distintas en el campo de las detecciones automáticas de logotipos, comprobando que es un mundo muy activo donde se siguen realizando nuevos desarrollos y mejoras a lo largo de todo el mundo.

En paralelo con el estudio del campo, se han sabido detectar cuáles son los retos más importantes a la hora de hacer frente a un proceso de detección automática, y se han tenido en cuenta debidamente para lograr las estrategias presentadas.

Se ha cubierto positivamente el objetivo de diseñar un método rápido y efectivo de minimizar el área de búsqueda de logotipos, haciendo que la velocidad del programa sea considerablemente mayor que sin este paso además de reducir las posibilidades de obtener falsas detecciones.

Esta componente ha sido la más exigente, con cambios continuos y mejoras en los módulos de saliencia y color, incluyendo la codificación de varias variantes en ambos casos que, aunque permanecen en el código del sistema, no son utilizadas. El caso de la distancia Delta-E es un ejemplo de un desarrollo que ha sido muy costoso y finalmente se ha optado por retirar del flujo.

Se ha codificado un método para detectar la perspectiva de una imagen, aunque la utilidad de éste no ha sido tan alta en el sistema completo como podía esperarse al comienzo del desarrollo, y finalmente esta componente ha representado una parte pequeña de todo el desarrollo.

A partir de la salida de datos del sistema, se pueden lograr los objetivos deseados de información temporal sobre la presencia de un logo a lo largo de una secuencia, aunque estos datos se presenten a nivel de fotograma. También se ha trabajado en tener una interfaz funcional que permita ver el resultado del análisis en tiempo real.

Para concluir, el código creado en este proyecto ha cumplido con creces los objetivos esperados aunque existen muchos puntos donde la mejora es posible.

Merece la pena recordar que este trabajo estaba destinado a una investigación genérica de técnicas que permitiesen la detección de logotipos en transmisiones deportivas y el resultado final es un código funcional que en el ámbito de transmisiones de fútbol, devuelve unos resultados con una precisión y un *recall* bastante decentes en relación a otros trabajos analizados con una eficiencia temporal muy alta.

## **6.2 Trabajo futuro**

A continuación, se detallan algunas posibles mejoras a este sistema:

### **6.2.1 Mejora del módulo de preprocesado**

En algunas secuencias, los umbralizados automáticos no funcionan como deberían, y devuelven una máscara binaria con un porcentaje de la imagen cercano al 100% activo. Este comportamiento podría modificarse haciendo un control de las zonas activas y variando la llamada al umbralizado hasta obtener un resultado razonable.

### **6.2.2 Mejora de la eliminación de falsos *matches***

Una vez obtenida la región, a parte de usar el umbral de distancia configurable y los otros métodos de descarte citados en la sección 4.2.7 podría usarse comparación de histogramas para añadir otro nivel de seguridad.

### **6.2.3 Mejora de la detección de regiones en la máscara de Pols**

Sin duda es uno de los puntos más débiles de éste proyecto como se ha podido deducir de los resultados de las pruebas. Sería necesario realizar pruebas más intensas con la creación de las nubes de PoIs y el umbralizado posterior. Quizá podría utilizarse algún algoritmo que detectase las distancias entre los puntos que complementase al modelo actual.

### **6.2.4 Adaptación a secuencias distintas a fútbol**

También se ha podido comprobar en la sección de pruebas que los resultados han sido bastante negativos para todas aquellas secuencias de vídeo que no provenían de un partido de fútbol. Sería posible entrenar al sistema manualmente y crear una serie de parámetros predefinidos para cada deporte, de modo que el usuario pueda introducir el deporte del vídeo de entrada y obtener un análisis específico para ese tipo de imágenes.

## **6.3 Conclusión personal**

La elección de este trabajo en particular por mi parte se debe a mi vocación por el diseño de logotipos e imagen de marca, campos en los que espero de centre mi actividad profesional en el futuro.

Por lo anterior, además de la oportunidad de aprender técnicas informáticas de visión artificial con una utilidad muy fuerte hoy en día, el desarrollo de este trabajo me ha permitido ligar los dos campos que forman mi perfil académico y profesional y conocer a par detalles técnicos sobre visibilidad y detección de logotipos en determinados entornos que es probable que me sean de utilidad en el futuro.

Haciendo un análisis personal de los objetivos logrados, lo cierto es que hace un año y no teniendo ningún conocimiento en absoluto de esta materia, me era difícil imaginar que finalmente, el resultado de este proyecto iba a ser un código de detección rápido, robusto y aceptablemente funcional.

# Referencias

---

- [1] Jianlong Fu, Jinqiao Wang, and Hanqing Lu, Effective Logo Retrieval with Adaptive Local Feature Selection. ACM, 2010.
- [2] Syed Yasser Arafat, Syed Afaq Husain, Iftikhar Azim Niaz , Muhammad Saleem. Logo Detection and Recognition in Video Stream. ICDIM, 2010
- [3] Martin Köstinger, Peter M. Roth, and Horst Bischof. Planar Trademark and Logo Retrieval. ICG-TR, 2010
- [4] Stefan Romberg, Lluís Garcia Pueyo, Rainer Lienhart, Roelof van Zwol. Scalable Logo Recognition in Real-World Images. ICMR, 2011
- [5] Stefan Romberg, Lluís Garcia Pueyo, Rainer Lienhart, Roelof van Zwol, Michele Trevisiol, Yannis Kalantidis. Scalable Triangulation-based Logo Recognition. ICMR, 2011
- [6] Alok Watve, Shamik Sural, Soccer video processing for the detection of advertisement billboards. ELSEVIER, 2008
- [7] Jim Kleban, Xing Xie, Wei-Ying Ma. Spatial pyramid mining for logo detection in natural scenes. IEEE, 2008
- [8] Ke Gao, Shouxun Lin, Yongdong Zhang, Sheng Tang, Dongming Zhang. Logo detection based on spatial-spectral saliency and partial spatial context. IEEE 2009
- [9] Andrew D. Bagdanov, Lamberto Ballan, Marco Bertini, and Alberto Del Bimbo. Trademark matching and retrieval in sports video databases. MIR 2007.
- [10] Lamberto Ballan, Marco Bertini, Alberto Del Bimbo and Arjun Jain. Automatic trademark detection and recognition in sport videos. IEEE 2008
- [11] <https://www.blinkfire.com>
- [12] David G Lowe. Distinctive image features from scale-invariant keypoints. International journal of computer vision, 91–110, 2004.
- [13] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. Computer Vision–ECCV 2006, 404–417, 2006.
- [14] J. Matas, O. Chum, M. Urban, and T. Pajdla. "Robust wide baseline stereo from maximally stable extremal regions." Proc. of British Machine Vision Conference, 384-396, 2002.
- [15] Gonzalez, R.C., Woods, R.E., 2004. Digital Image Processing, second ed. Pearson Education.
- [16] Martin A. Fischler, Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. 1981.
- [17] P.-E. Forssén and D. Lowe. Shape descriptors for maximally stable extremal regions. In Proc. IEEE Intern. Conf. on Computer Vision, 2007.
- [18] J. Matas, T. Obdrzalek, and O. Chum. Local affine frames for widebaseline stereo. In Proc. Intern. Conf. on Pattern Recognition, volume 4, 363-366, 2002.
- [19] [http://resources.esri.com/help/9.3/arcgisengine/java/gp\\_toolref/coverage\\_tools/transform\\_arc3.gif](http://resources.esri.com/help/9.3/arcgisengine/java/gp_toolref/coverage_tools/transform_arc3.gif)
- [20] Nobuyuki Otsu (1979). "A threshold selection method from gray-level histograms". IEEE Trans. Sys., Man., Cyber. 9 (1): 62–66.
- [21] M. K. Hu, "Visual Pattern Recognition by Moment Invariants", IRE Trans. Info. Theory, vol. IT-8, pp.179–187, 1962
- [22] <http://ieeexplore.ieee.org/Xplore/>
- [23] <http://www.springer.com>
- [24] OpenCV[24].org – BSD License [http://OpenCV\[24\].org/license.html](http://OpenCV[24].org/license.html)
- [25] ISO 11664-4:2008(E)/CIE S 014-4/E:2007: Joint ISO/CIE Standard: Colorimetry — Part 4: CIE 1976 L\*a\*b\* Colour Space
- [26] Sharma, G., Wu, W. and Dalal, E. N. (2005), The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. Color Res. Appl., 30: 21–30. doi: 10.1002/col.20070
- [27] Marius Muja and David G. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration", 2009
- [28] <https://github.com/gfiumara/CIEDE2000>
- [29] Canny, J., A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 679–698, 1986.
- [30] Hough, P.V.C. Method and means for recognizing complex patterns. U.S. Patent 3,069,654, Dec. 18, 1962.

## Glosario

---

Binarización	Operación sobre una imagen en la que todos los píxeles cuyo valor esté por encima de uno dado toman un valor máximo y aquellos que estén por debajo uno mínimo.
Canal	Una imagen a color está formada por la combinación de varias magnitudes, cada canal es la imagen con una sola de ellas.
Canny	Método de detección de bordes de una imagen
CIE	<i>Commission internationale de l'éclairage</i> – Autoridad internacional en el estudio de color y luz
Delta-E	Método de distancia de color para el espacio CIE-Lab
Descriptores	Un ~ visual es un método estándar de almacenar información de forma, color, textura, etc. en una imagen o región de esta.
Eficacia	Capacidad del código de realizar el cometido que se espera de éste.
Eficiencia	Cuanto mayor es la eficiencia de un código, menor es el tiempo que necesita para ejecutarse
Fotograma	Cada una de las imágenes que forman un vídeo
Homografía	Transformación proyectiva que determina una correspondencia entre dos figuras geométricas planas
Hough	(Transformada de) método matemático de detección de varios tipos de formas (rectas, circunferencias...) en una imagen binaria
HSV	Espacio de color con canales de tonalidad, saturación e intensidad
Imá. de contornos	Máscara donde se muestran los bordes y fronteras entre regiones diferenciadas de una imagen
Logo	Imagen que representa a una marca o empresa mediante un conjunto determinado de formas y colores
Máscara	Imagen procesada resultante de aplicar una o varias operaciones a una imagen de entrada.
RGB	Espacio de color con canales rojo, verde y azul
Saliencia	Medida de lo llamativa que es una zona de una imagen al ojo humano
Template/Plantilla	Imagen cargada en el programa del logo que debe ser detectado en el vídeo.
Umbralización	Binarización con más de dos valores
WXGA	Estándar de resolución de pantalla. 1280x720
XGA	Estándar de resolución de pantalla. 1024x768
SVGA	Estándar de resolución de pantalla. 800x600

## Anexos

---

### ***A Manual de uso del programa***

El programa se ejecuta aceptando los siguientes parámetros por consola:

```
./<Programa> <Umbral> <ruta_logo> <ruta_video> <salto_fotogramas>  
<limite_fotogramas>
```

Siendo el salto de fotogramas cuantos fotogramas se analizan (si es 5 se analiza uno de cada 5) y el limite marca cuando se termina de ejecutar el programa, si está vacío terminara el video.

Además, no es necesario introducir todos los parámetros anteriores, aunque sí seguir ese orden, es decir, para introducir el tercero es necesario haber introducido los dos anteriores.

Al iniciarse la ejecución, se muestra una ventana que informa al usuario de los primeros pasos genéricos que se ejecutan antes del procesado de cada fotograma.

Esta ventana se cierra para solicitar al usuario introducir si lo desea, una selección manual del logo en el vídeo que será usada para sustituir a la plantilla en el programa. Si se opta por usar la plantilla basta con clicar la tecla escape.

A partir de este momento, se verá por pantalla el resultado de cada análisis y, si unas determinadas banderas están activadas en el código, estos resultados se almacenarán como datos en un archivo de texto según el nombre de los parámetros de entrada y como imágenes con los logos detectados en ellas.