

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 6



Componente para la Detección de Logos en Medios Audiovisuales

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO
EN CIENCIAS INFORMÁTICAS**

AUTOR: Dariel Acosta Jiménez

TUTOR: MSc. Yanio Hernández Heredia.

CO-TUTOR: Ing. Jean Michael Suárez Pérez

Ciudad de La Habana, junio 30 de 2011.

Año 52 de la Revolución

DEDICATORIA

Dedico este trabajo a mis padres que siempre han sido mi ejemplo a seguir. A mi padre, que aunque ya no esté con nosotros siempre está presente en mi corazón, a mi madre, que su esfuerzo y dedicación han hecho posible que yo pueda estar aquí hoy, a mi hermano, que todavía le falta todo este camino por recorrer y también a toda mi familia que siempre ha estado a mi lado apoyándome.

AGRADECIMIENTOS

Agradezco a todas las personas que de una forma u otra me han ayudado a conseguir este sueño: a mis padres en primer lugar por apoyar mi decisión de estudiar en la UCI y ayudarme en todo lo que necesitara, a mis tías y tíos, a mi hermano, a mis primas y primos también, a toda mi familia por estar siempre a mi lado. También quisiera agradecer a todos mis amigos: a mis amigos de siempre, a los de la universidad, a los amigos del IPVCE, y a todos los que han estado ahí cuando los he necesitado. Agradecer además a mis tutores, Yanio que me encaminó en el comienzo de la Investigación y Jean Michel que se ha mantenido ayudándome. Agradecer también a mi oponente Zorilín y los miembros del tribunal por sus críticas oportunas que permitieron que este trabajo tuviera la calidad requerida. Además agradecer al profesor Héctor y a los miembros del proyecto, tanto estudiantes como profesores, a los que siempre pude acercarme ante cualquier duda. Son muchas las personas que de una forma u otra me han ayudado a culminar mis estudios. He tratado de no mencionar nombres pues sería injusto si alguno quedara fuera. A todos ustedes, muchas gracias.

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo:

Componente para la Detección de Logos en Medios Audiovisuales

Autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 30 días del mes de junio del año 2011.

Autor: Dariel Acosta Jiménez

Tutor: Ing. Jean Michael Suárez Pérez

DATOS DE CONTACTO

Tutor:

Nombre y apellidos: Yanio Hernández Heredia.

Sexo: M F

Institución: Universidad de las Ciencias Informáticas.

Correo electrónico: yhernandezh@uci.cu.

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Año de graduación: 2007.

Institución donde se graduó: Universidad de las Ciencias Informáticas.

CoTutor:

Nombre y apellidos: Jean Michael Suárez Pérez.

Sexo: M F

Institución: Universidad de las Ciencias Informáticas.

Correo electrónico: jmsuarez@uci.cu.

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Año de graduación: 2009.

Institución donde se graduó: Universidad de las Ciencias Informáticas.

Resumen

El número de canales de un país es bastante grande, lo que dificulta el control efectivo de los mismos; es necesario poder identificar cada transmisión televisiva y realizarlo de forma manual constituye un proceso engorroso debido a su gran cantidad. Cada canal televisivo posee un logo que lo identifica y esta característica resulta de gran utilidad porque permite realizar clasificaciones automáticas de los videos, utilizando como identificador el logo presente en el mismo. Resulta importante también, la detección de la aparición o ausencia del logo en la transmisión porque las televisoras al pasar a comerciales retiran su logo y de este modo es posible contribuir a la detección automática de publicidad. Con la investigación se desarrolló un componente de software, que permite la automatización del proceso de detección de logos para el producto Sistema de Captura y Catalogación de Medias. La investigación arrojó como resultados la documentación de diversas técnicas para la detección de patrones en contenidos audiovisuales y la determinación de técnicas efectivas para la detección de logos además del componente de software que se implementó.

Palabras claves: detección, logos, televisión, video.

Abstract

The number of channels in a country is quite large making it difficult for their effective control, it is necessary to identify each telecast and manually do it becomes a very complicated process because of their large numbers. Each television channel has a logo that identifies it and this feature is useful because it allows automatic classification of videos, using the present logo on it. It is also important to detect the occurrence or absence of the logo because the television stations withdraw their logo from the transmission when they move to commercials and thus may contribute to the automatic detection of advertising. With the research was developed a software component that allows the automated detection of logos for the product SCCM. Results from the research are the documentation of various techniques for detecting patterns and determining the more effective for logos detection and also the software component that was implemented.

Keywords: detection, logos, television, video.

Índice

Introducción	3
CAPÍTULO 1: Fundamentación Teórica del desarrollo del componente para la detección de logos.	7
1.1 Introducción	7
1.2 Conceptos asociados al dominio del problema	7
1.3 Objeto de Estudio	9
1.3.1 Descripción General.....	9
1.4 Descripción actual del dominio del problema	20
1.5 Situación Problemática	20
1.6 Análisis de otras soluciones existentes	21
1.7 Conclusiones	22
CAPÍTULO 2: Descripción y diseño de algoritmos y componente para la detección de logos en materiales audiovisuales.....	24
2.1 Introducción	24
2.2 Tecnologías y tendencias en el procesamiento digital de imágenes	24
2.3 Lenguaje de Programación.....	24
2.3.1 C++	24
2.3.2 Framework Qt 4.7	25
2.5 Bibliotecas utilizadas.	26
2.5.1 OpenCV	26
2.6 El Lenguaje Unificado de Modelado (UML) como soporte de la modelación de la solución propuesta	27
2.7 Requisitos funcionales.....	27
2.8 Diseño de un algoritmo para la detección de logos.	27
2.8.1 Diseño de un algoritmo para Identificar video (Anexo 1).....	28
2.8.2 Diseño de un algoritmo exhaustivo para realizar anotaciones en el video (Anexo 2)	28
2.8.3 Diseño de un algoritmo rápido para realizar anotaciones en el video (Anexo 3).....	29
2.8.4 Diseño de un algoritmo para guardar logos desconocidos (Anexo 4).....	29
2.9 Conclusiones	30

CAPÍTULO 3: Implementación y Pruebas de algoritmos y componente para la detección de logos en materiales audiovisuales.....	31
3.1 Introducción	31
3.2.1 Descripción de las clases del componente para la detección de logos.....	31
3.3 Validación del componente y algoritmo para la detección de logos.....	34
3.3.1 Pruebas de compatibilidad de formatos.	34
3.3.2 Pruebas al algoritmo de detección de logos.....	37
3.4 Conclusiones	39
Conclusiones Generales.....	40
Recomendaciones	41
Referencias Bibliográficas	42
Anexo 1: Algoritmo para identificar video.....	44
Anexo 2: Algoritmo exhaustivo para realizar anotaciones en el video.	45
Anexo 2: Algoritmo exhaustivo para realizar anotaciones en el video.	45
Anexo 3: Algoritmo rápido para realizar anotaciones en el video.	46
Anexo 4: Algoritmo para guardar logos desconocidos.....	47

Introducción

La humanidad, durante toda su historia, ha necesitado almacenar información para las generaciones futuras. La transmisión del conocimiento de generación en generación no fue suficiente y comenzaron a surgir diversos soportes para contener la misma. El lenguaje escrito constituyó un gran paso de avance, con él era posible guardar de forma efectiva el conocimiento acumulado. La tecnología nunca estuvo al margen del desarrollo de estos mecanismos y la aparición de mejores soportes tecnológicos permitió la difusión cada vez más generalizada de la información. La fotografía constituyó un salto en ese sentido pues al poder almacenar una imagen se lograba describir mejor a los objetos. De la fotografía se pasó al cine con la posibilidad de guardar imágenes en movimiento y posteriormente el sonido se agregó a este soporte para contar con un reflejo casi exacto de lo que sucedía en el mundo. El avance de la electrónica y su aplicación a todos los campos de la sociedad posibilitó que surgiera la televisión, que como medio de difusión masivo logró acercarse de forma más efectiva a las masas, puesto que pudo llevar las imágenes y sonidos del cine hasta la propia casa propiciando aún más la interacción con la población.

La televisión continuó evolucionando desde entonces, se cambió de la televisión en blanco y negro a la televisión en colores y posteriormente de la analógica a la digital. La revolución digital tuvo un gran impacto en la televisión: permitió una mayor resolución, mejor calidad de la imagen y del sonido, mayor cantidad de canales e imagen en alta definición. Revolucionó tanto el mundo de la televisión, que en algunos países desarrollados las transmisiones de televisión analógica están a punto de finalizar para siempre convirtiéndose la televisión digital en el estándar. Hoy día el número de canales con que puede contar un país es bastante grande, lo que dificulta el control efectivo de los mismos. Los gobiernos de cada país dictan las leyes referentes a la publicidad y al contenido que se encuentra permitido transmitir, por lo tanto, necesitan poder controlar el correcto cumplimiento de estas leyes. Para ello, es necesario poder identificar cada transmisión televisiva y realizarlo de forma manual se vuelve un proceso muy complicado debido a su gran cantidad.

Cada canal televisivo posee un logo que lo identifica de manera unívoca. La noción de logo se utiliza para referirse, en forma indistinta, a un logotipo (la representación tipográfica), isotipo (un ícono o signo visual) o isologo (combinación de logo e isotipo) **(Definición.de, 2011)**. Se utiliza para identificar al canal emisor de la transmisión y además contribuir a la preservación de su propiedad intelectual, denotando que el

material ha sido producido por la televisora en cuestión. Esta característica resulta de gran utilidad porque permite realizar clasificaciones automáticas de los videos, utilizando para ello el logo presente en el mismo. De este modo es posible identificar si el material transmitido consiste en un documental o un evento deportivo, si porta algún logo de las televisoras especializadas en este campo. Resulta muy importante también la detección de la aparición o ausencia del logo porque las televisoras al pasar a comerciales retiran su logo de la transmisión. Detectar estos momentos donde no está presente el logo puede resultar de gran utilidad para la detección automática de publicidad, lo que permitiría controlar el tiempo establecido para comerciales por las leyes del país en cuestión, así como facilitar la posterior eliminación de estos.

Una técnica novedosa en la tarea de la protección de la propiedad intelectual de las televisoras consiste en la inclusión de marcas de agua en sus videos. Esta no puede ser percibida a simple vista pero es posible detectarla fácilmente utilizando una computadora y la clave adecuada. **(López, 2011)** Del mismo modo que un logo puede contribuir a la integridad de la propiedad de las televisoras pues identifica también al productor del material lo que de una forma más solapada.

El Departamento Señales Digitales perteneciente al centro GEYSED de la Facultad 6 en la Universidad de las Ciencias Informáticas (UCI), creó el producto Sistema de Captura y Catalogación de Medias (SCCM) con el objetivo de permitir a las televisoras digitalizar y catalogar sus archivos de video, este cuenta además con un componente para la monitorización automática de canales de televisión. Para el producto en cuestión resulta necesario conocer la identidad de los canales monitorizados así como la duración del logo presente en el mismo, con el objetivo no sólo de identificarlo, sino también para contribuir a la detección de publicidad, por lo que surge la necesidad de realizar su identificación de manera automatizada.

De la situación anterior se formula el siguiente **problema a resolver**: El control de la identidad de los canales de televisión es ineficiente cuando estos no son monitorizados o cuando este proceso se ejecuta manualmente.

La investigación se enmarca en el **objeto de estudio**: Técnicas para el procesamiento y detección de patrones en contenidos audiovisuales. El **campo de acción** de la misma es la automatización de la detección de logos durante la monitorización automática de canales de televisión en el producto SCCM.

Como **objetivo general** de la investigación se propone: Desarrollar un componente de software que automatice el proceso de detección de logos en transmisiones televisivas.

Por lo planteado anteriormente la **idea a defender** consiste en que: El componente para la detección de logos en transmisiones televisivas automatizará la identificación de las televisoras monitorizadas en el producto SCCM.

Para alcanzar el objetivo planteado anteriormente se han establecido las siguientes **tareas de la investigación**:

1. Caracterizar las variantes de técnicas y algoritmos existentes de mejores resultados, así como las tendencias utilizadas en la solución del problema planteado, a partir de un estudio del estado del arte.
2. Seleccionar una o varias técnicas eficientes para detección de logos en video a utilizar en el componente a desarrollar con las justificaciones pertinentes.
3. Realizar los algoritmos necesarios para la detección de logos en materiales audiovisuales.
4. Determinar las tecnologías apropiadas para la implementación del componente.
5. Implementar el componente de detección de Logos en video teniendo en cuenta las técnicas seleccionadas y los algoritmos realizados.
6. Validar el componente implementado realizando pruebas de caja negra.

Para la realización de la investigación fue necesario emplear métodos teóricos y empíricos.

Métodos Teóricos:

Histórico-Lógico: El método histórico estudia la trayectoria real de los fenómenos y acontecimientos en el transcurso de su historia. El método lógico investiga las leyes generales del funcionamiento y desarrollo de los fenómenos. **(Álvarez, 1997) (Alonso, 1998)**

Permite analizar el avance y las tendencias en las aplicaciones y bibliotecas que se dedican a la detección de patrones en imágenes y medios audiovisuales.

Analítico-Sintético: El análisis es una operación intelectual que posibilita descomponer mentalmente un todo complejo en sus partes y cualidades. La síntesis es la operación inversa, que establece mentalmente la unión entre las partes, previamente analizadas y posibilita descubrir relaciones y características generales entre los elementos de la realidad. **(Álvarez, 1997) (Alonso, 1998)**

Permite valorar y definir las distintas técnicas que son posibles emplear para la detección de un logo en una imagen para así definir cuál es la que mejor fiabilidad brinda.

Modelación: El modelo es una reproducción simplificada de la realidad, que cumple una función heurística, ya que permite descubrir y estudiar nuevas relaciones y cualidades del objeto de estudio. **(Álvarez, 1997) (Alonso, 1998)**

Permite modelar las clases y la creación de artefactos de ingeniería para el posterior desarrollo de la aplicación.

Métodos Empíricos:

Entrevista: La entrevista es una conversación planificada entre el investigador y el entrevistado para obtener información. **(Álvarez, 1997) (Alonso, 1998)**

Al jefe de proyecto Video Vigilancia para conocer las variantes existentes en el mundo del software libre para la detección de patrones en imágenes así como la mejor forma de realizar la implementación debido a la gran experiencia que posee en dicho tema.

Descripción de los Capítulos

Capítulo 1: En este capítulo se abordan temas referentes a las técnicas para la detección de patrones en imágenes y videos. Sus ventajas y desventajas principales y se propone la técnica a implementar para la detección de los logos en los materiales audiovisuales.

Capítulo 2: En este capítulo se abordan los temas relacionados con las tecnologías que se utilizarán en la solución, lenguaje de programación y de modelado y se explicará por qué fueron seleccionados. Además se desarrollaron los algoritmos necesarios para dar solución al problema planteado.

Capítulo 3: En este capítulo se abordan los temas relacionados a la implementación del componente de software así como las pruebas realizadas al mismo para validar el algoritmo propuesto.

CAPÍTULO 1: Fundamentación Teórica del desarrollo del componente para la detección de logos.

1.1 Introducción

En este capítulo se abordan los temas referentes a las imágenes y videos digitales. Se brinda información del estado actual de las bibliotecas y aplicaciones que realizan detección de logos en imágenes, así como las diferentes técnicas que pueden ser utilizadas para esto.

1.2 Conceptos asociados al dominio del problema

1.2.1 Imagen digital:

Una imagen digital es una matriz de $M \times N$ elementos que se obtiene mediante la discretización de la imagen analógica, tanto en las dimensiones geométricas, mediante la generación de celdas por muestreo equiespaciado de la superficie, como en sentido radiométrico, mediante la cuantificación.

(González, 2002)

Imagen digital es la representación bidimensional de una imagen empleando bits. De acuerdo a la resolución que presenta la imagen, estática o dinámica, se podrá hablar de gráfico rasterizado (o mapa de bits; representa una rejilla rectangular de píxeles o punto de color, la cual se puede visualizar en un monitor, en un papel o en cualquier otro dispositivo de representación que se emplee) o de gráfico vectorial (imagen producto de objetos de geométricos independientes).

(DefiniciónABC, 2011)

1.2.2 Video digital:

El vídeo digital es un tipo de sistema de grabación de vídeo que funciona usando una representación digital de la señal de vídeo. Es el resultado de la combinación de secuencias de imágenes, audio, texto y movimiento que se obtienen por muestreo y cuantificación de la señal de video analógica.

(Bimbo, 2005)

1.2.3 Logo:

Distintivo compuesto por letras e imágenes, peculiar de una empresa, una marca o un producto.

(Definición.de, 2011)

1.2.4 Regiones de Interés (ROI):

Subvolumen del volumen de datos sobre el que serán calculadas las estadísticas. **(SourceForge, 2011)** Área rectangular de una imagen delimitada por cuatro puntos. **(UPV, 2011)**

1.2.5 Descriptores:

Permiten analizar y caracterizar el contenido de fuentes audiovisuales para su posterior indexación, búsqueda o comparación. **(Pérez Noriega, y otros, 2006)**

1.2.6 Marcas de agua:

La marca de agua digital es un código de identificación que se inserta directamente en el contenido de un archivo multimedia (imagen, audio, video), de manera que sea difícil de apreciar por el sistema perceptual humano, pero fácil de detectar usando un algoritmo dado y una clave, en un ordenador. **(López, 2011)**

1.2.7 Histograma:

Representación gráfica de una distribución de frecuencias por medio de rectángulos, cuyas anchuras representan intervalos de la clasificación y cuyas alturas representan las correspondientes frecuencias. **(RAE, 2011)**

1.2.8 Fotogramas:

También denominados frames, constituyen la unidad básica de información y generalmente son filmados a una velocidad de 1/25 o 1/30 segundos. Básicamente consisten en imágenes estáticas cuya reproducción a la velocidad adecuada crean la sensación de movimiento característica de un video. **(Sáez Peña, 2006)**

1.2.9 Tomas de cámara:

También denominados shots, son el conjunto de frames filmados por una cámara desde que esta comienza a grabar hasta que deja de hacerlo. Son los segmentos elementales de un video, los átomos sintácticos. La unión entre diferentes shots se realiza por medio de efectos de edición. **(Sáez Peña, 2006)**

1.3 Objeto de Estudio

1.3.1 Descripción General

La búsqueda de patrones es una técnica de análisis que se puede aplicar en detección de objetos, reconocimiento, seguimiento y correspondencia. Dada una imagen (un patrón o modelo), encontrar sus apariciones dentro de otra imagen mayor. No se buscan sólo las apariciones “exactas”, sino permitiendo cierto grado de variación respecto al patrón. **(García, 2010)**

La detección de patrones en imágenes es un amplio campo que recién comienza su desarrollo en la informática actual, constituyendo la base fundamental de la visión por computadoras. En este sentido existen diversas técnicas que permiten realizar esta tarea, las cuales serán explicadas a continuación con el objetivo de determinar cual ofrece mejores resultados para la tarea específica de detectar logotipos en videos.

1.3.1.1 Template Matching

El método más sencillo para realizar esta tarea consiste en la comparación de plantillas.

Sea A una imagen (de tamaño $W \times H$), y sea P un patrón (de $w \times h$), el resultado es una imagen M (de tamaño $(W-w+1) \times (H-h+1)$), donde cada píxel $M(x, y)$ indica la “verosimilitud” (probabilidad) de que el rectángulo $[x, y] - [x+w-1, y+h-1]$ de A contenga el patrón P . La imagen M se define usando alguna función de diferencia (o similitud) entre dos trozos de imagen.

$$M(x, y) := d(\{A(x, y), \dots, A(x+w-1, y+h-1)\}, \{P(0,0), \dots, P(w-1, h-1)\})$$

Una de las principales aplicaciones del Template Matching es la detección de objetos.

Proceso de detección de objetos usando búsqueda de patrones.

1) Conseguir un patrón, P , representativo de la clase de objetos a buscar.

2) Aplicar el Template Matching a la imagen, obteniendo M .

3) Buscar los máximos (o mínimos) locales de M .

3.1) Buscar el máximo global, $(l_x, l_y) = \operatorname{argmax}_{x,y} M(x, y)$.

3.2) Si $M(l_x, l_y)$ es menor que cierto umbral, acabar.

3.3) Añadir la posición (l_x, l_y) a una lista de localizaciones resultantes del proceso.

3.4) Poner a cero en M el rectángulo $[l_x-w, l_y-h] - [l_x+w, l_y+h]$.

3.5) Volver al paso 3.1.

Problemas:

- La técnica es muy sensible a cambios de escala, rotación o deformaciones 3D de los objetos.

Posibles Soluciones:

- Utilizar varios patrones, con distintos tamaños y rotaciones.
- Hacer una búsqueda multiescala. Aplicar el proceso escalando la imagen a: 50%, 75%, 100%, 125%, (...)
- Usar alguna técnica de atención selectiva. Por ejemplo, usar color o bordes para centrar la atención en ciertas partes de la imagen.

La aplicación de la técnica es muy costosa. Cuando la resolución aumenta al doble, el tiempo se multiplica por 16. (**García, 2010**)

1.3.1.2 Integrales proyectivas.

Una integral proyectiva (o, simplemente, una proyección) de una imagen es la media de los píxeles por filas (proyección vertical), por columnas (proyección horizontal) o a lo largo de un ángulo cualquiera.

Definición: Sea **A** una imagen de WxH. La **integral proyectiva vertical**, denotada por **P_{VA}**, es una tabla de tamaño H definida por:

$$P_{VA}(y) = 1/W \cdot \sum_{x=0..W-1} A(x, y); \forall y= 0, \dots, H-1$$

La **integral proyectiva horizontal**, denotada por **P_{HA}**, es:

$$P_{HA}(x) = 1/H \cdot \sum_{y=0..H-1} A(x, y); \forall x= 0, \dots, W-1$$

Las integrales proyectivas se pueden usar en detección, seguimiento y segmentación. Normalmente como fase previa a otros procesos. La principal característica es la reducción de dimensiones de imágenes 2D a proyecciones 1D. Son más rápidas de procesar pero se puede perder información relevante. La técnica es aplicable cuando los objetos de interés son distinguibles por intensidad (más claros o más oscuros). Si esto no es posible, también se pueden aplicar las proyecciones sobre imágenes de bordes.

Ventajas:

- Se trabaja con información simplificada y acumulada. Más eficiencia y menos sensibilidad a ruido.

Inconvenientes:

- En algunas aplicaciones, la proyección puede suponer perder información relevante.
- Difícil establecer umbrales máximos y mínimos adecuados. **(García, 2010)**

1.3.1.3 Análisis de color.

Diversas técnicas utilizan el color para su funcionamiento y entre ellas se puede encontrar:

- Reproyección del histograma.
- Detección de zonas de color.
- Comparación de histogramas

La **reproyección del histograma** permite detectar un color partiendo de una imagen con regiones amplias de ese color. La idea consiste en usar histogramas para modelar un color objetivo. Es un caso de la detección de zonas de color. Para cada color se define una verosimilitud obtenida mediante un histograma de color.

Proceso:

- 1) Calcular el histograma de color de una zona de la imagen con el color objetivo.
- 2) Usar el histograma como una medida de verosimilitud, probabilidad o distancia en el espacio de color.
- 3) Dada una imagen nueva, aplicar los valores del histograma para detectar las zonas de color similar al color objetivo. **(García, 2010)**

La **detección de zonas de color** consiste en seleccionar y modelar un color objetivo y encontrar regiones conexas de ese color. Se aplica en sistemas croma-key o segmentación por color, detección de objetos como caras humanas usando color de piel y en seguimiento basado en color.

Para su aplicación en un sistema croma-key se toma una imagen del panel de color usado como fondo y se calcula su histograma. Después se usa la reproyección para obtener la máscara de segmentación. **(García, 2010)**

La **comparación de histogramas** permite obtener una medida subjetiva de similitud entre imágenes, basada en color. Es útil en aplicaciones como indexación de imágenes. Consiste en definir una medida de diferencia o similitud entre dos histogramas. La diferencia entre dos imágenes se reduce a la diferencia entre los histogramas correspondientes. Se pueden usar distintas medidas como: suma de diferencias al cuadrado, producto vectorial o solapamiento. **(García, 2010)**

Hasta el momento se ha analizado las técnicas que se aplican a imágenes de forma particular pero en la práctica muchas de las aplicaciones de detección de patrones no están diseñadas para trabajar exclusivamente con imágenes sino que también pueden procesar archivos de video directamente lo que facilita el trabajo de la misma aplicación de detección porque no es necesario segmentar el video en cada uno de los frames que lo componen sino que estos son accedidos directamente desde el archivo de video lo que conlleva un ahorro de espacio y de tiempo significativos.

Debido a que el formato MPEG se encuentra hoy día muy extendido, es conveniente el desarrollo de algoritmos de segmentación temporal de video que operen directamente en dominio comprimido, lo cual presenta algunas ventajas. En primer lugar, se consigue evitar el tiempo invertido en la descompresión de la secuencia de video. Por otra parte, se maneja una menor cantidad de información. Finalmente, el vídeo comprimido contiene algunas características de las imágenes ya calculadas tales como vectores de movimiento o intensidad media de bloques, las cuales son muy útiles para los algoritmos de segmentación temporal de video. Todo ello redundando en una mayor velocidad de los algoritmos que operan en dominio comprimido, posibilitando de esta forma el procesamiento de video en tiempo real. **(Sáez Peña, 2006)**

Mucho se ha escrito sobre la detección de patrones en videos, fundamentalmente sobre la detección de shots y segmentación temporal del video con el objetivo de detectar comerciales y publicidad, así como realizar resúmenes o índices visuales de los mismos, que faciliten encontrar información específica de forma rápida en los múltiples archivos de video con que puede contar una biblioteca audiovisual. También existen técnicas especializadas en la detección y seguimiento de objetos en videos fundamentalmente aplicadas en video vigilancia.

El elemento clave en los sistemas de segmentación temporal de video son los algoritmos de detección de efectos de edición. Los citados efectos de edición pueden ser clasificados en dos grandes grupos:

1. Transiciones abruptas: Se caracterizan porque emplean un solo frame para enlazar dos shots. Son las transiciones más abundantes existentes en video y existe únicamente un tipo de efecto de edición representativo de las transiciones abruptas, el corte.
2. Transiciones graduales: Se caracterizan porque emplean más de un frame para enlazar dos shots. A diferencia de las transiciones abruptas, las transiciones graduales ofrecen una gama amplísima de efectos entre los cuales se pueden citar el disolver, que es posiblemente el más utilizado.

La representación de las características perceptuales de las imágenes es fundamental para la segmentación temporal de video. Los algoritmos de análisis de imagen y de reconocimiento de patrones proporcionan la forma de obtener descriptores numéricos, los cuales ofrecen una medida cuantitativa de estas características. **(Sáez Peña, 2006)**

Características fundamentales:

Color: Hace referencia a los atributos cromáticos de la imagen. Las cualidades cromáticas del color se representan a través de los modelos geométricos de color, que generalmente son tridimensionales. Estos permiten la distinción entre colores y la estimación de la similaridad entre los mismos. Los colores son representados como puntos en el espacio de color. La diferencia entre colores se mide como la distancia entre los puntos que los representan, empleando una función de distancia.

Textura: Es un término amplio utilizado en reconocimiento de patrones para establecer que una imagen está compuesta de fragmentos, de cualquier tamaño, que se caracterizan por tener una intensidad no uniforme.

Forma: La forma de un objeto lo identifica de manera significativa. La manera tradicional de representar una forma es a través de un conjunto de características, obtenidas mediante herramientas de procesamiento de imágenes. Estas características pueden describir aspectos globales de la forma tales como el área, la longitud, la orientación del eje principal, etc. o elementos locales de su contorno como esquinas o puntos característicos.

Movimiento: El movimiento es el elemento característico del video y hace referencia al cambio en la posición relativa de las entidades espaciales o al movimiento de cámara. El análisis y la representación del movimiento es un tema ampliamente estudiado en reconocimiento de patrones y visión artificial. Los métodos que realizan una estimación aproximada del movimiento siguen dos enfoques distintos. Uno de ellos tiene en cuenta la variación en los niveles de gris de los píxeles entre un frame y el siguiente, calculando así el flujo óptico para cada píxel de la imagen. **(Sáez Peña, 2006)**

1.3.1.4 Métodos para la segmentación temporal de videos

Métodos basados en pixeles

Las primeras propuestas consideraron el valor de los pixeles como característica fundamental para describir el contenido de los frames, y la comparación entre frames adyacentes la clave para la detección de transiciones, particularmente las abruptas. **(Sáez Peña, 2006)**

Comparación entre dos frames consecutivos

Uno de los primeros métodos desarrollados consiste en la detección de transiciones a través del valor absoluto de la diferencia de la suma de las intensidades de todos los pixeles entre dos frames consecutivos. En el mismo trabajo se propone una variación consistente en calcular la suma de las diferencias de luminosidad entre los pixeles de ambos frames. Cada pixel es comparado con el que ocupa su misma posición en el otro frame. Esta misma idea se propone también para imágenes en color, basándose no sólo en la intensidad de los pixeles sino además en la información del color. En cualquier caso, se detecta una transición cuando la medida calculada supera cierto umbral. **(Sáez Peña, 2006)**

Variación temporal de la luminosidad

El análisis restringido a un par de frames expuesto en el apartado anterior puede extenderse al análisis de las variaciones de intensidad de los pixeles a lo largo del tiempo. Así, se etiqueta cada pixel según se mantenga constante, varíe súbitamente, varíe de forma lineal, o presente valores aleatorios a consecuencia del movimiento. La distribución final de los valores de las etiquetas asignadas indicará la presencia de una transición abrupta o gradual. **(Sáez Peña, 2006)**

Métodos basados en histogramas

Es posible realizar la comparación de frames empleando características globales tales como el histograma. La principal ventaja de los métodos basados en histogramas es que son más robustos a los movimientos de los objetos y de cámara. Sin embargo, su mayor debilidad se manifiesta cuando dos frames pertenecientes a distintos shots presentan histogramas similares. En este caso, la transición entre los shots puede no ser detectada. **(Sáez Peña, 2006)**

Diferencia de histogramas

Se calcula el histograma de luminosidad para cada frame que se pretende comparar. Se obtiene una medida de la similaridad entre frames mediante la suma de las diferencias en valor absoluto de los bins correspondientes de los histogramas involucrados en la comparación. Si la medida obtenida excede cierto umbral, se declara un corte. **(Sáez Peña, 2006)**

Diferencia ponderada de histogramas

Se emplean diferencias de histogramas de color en el que no se pondera de igual manera a todos los planos de color del modelo. Esto permite definir la importancia que se le asigna a cada componente de color pudiendo determinarse los valores de los pesos mediante un método de aprendizaje. **(Sáez Peña, 2006)**

Intersección de histogramas

La intersección de histogramas se realiza habitualmente a través del operador mínimo, de forma que al comparar los bins de dos histogramas el resultado es siempre el menor de los dos. Esta aproximación ha sido utilizada para varios espacios de color. **(Sáez Peña, 2006)**

Métodos basados en movimiento

Movimiento global

Para este método se definen los videos o bits como la colección de imágenes que pueden obtenerse a partir de una información dada, aplicando las transformaciones pertinentes, mediante el uso de los ocho parámetros del modelo. Si dos frames pertenecen al mismo shot, estarán en el mismo video o bit. De esta forma, se considera que se ha producido una transición cuando la distancia entre los videos o bits de dos frames consecutivos sea superior a un umbral. **(Sáez Peña, 2006)**

Flujo óptico.

El flujo óptico es una técnica de análisis de imágenes que se aplica en secuencias de vídeo. En concreto, este define los vectores de movimiento de diferentes trozos de la imagen. Este se aplica en detección de movimiento, seguimiento de objetos por partes, compresión de vídeo, composición, etc. Existen diversas formas de calcular el flujo óptico. Una forma sencilla está basada en la técnica del Template Matching: dividir la imagen en bloques, para cada bloque de una imagen buscar la correspondencia en la otra.

Parámetros para el cálculo del flujo óptico:

Tamaño de los bloques a usar:

- Ni muy pequeños ni muy grandes. Si son pequeños, contienen pocas características y el matching es poco fiable.
- Si son grandes, se pierde resolución (menos vectores de movimiento). También hay problemas si el bloque se sale de la imagen.

Radio de búsqueda. Determina el tamaño de la zona, en la imagen t , donde se busca el bloque de entrada de la $t-1$.

- Cuanto más grande, más tiempo de ejecución.
- Si es muy pequeño y el movimiento es mayor, el resultado será impredecible.

La función de matching a emplear puede ser una suma de diferencias. Para conseguir invariancia a cambios de iluminación es mejor utilizar un producto vectorial normalizado.

Ventajas:

- Permite comprender mejor la información contenida en un vídeo, la evolución en la escena: detectar si hay cambios en la escena, en qué posiciones, qué cantidad, etc.

Inconvenientes:

- La técnica es muy lenta. Es inviable aplicarla en tiempo real.
- Difícil ajustar los parámetros para un funcionamiento óptimo: tamaño de bloques y radio de búsqueda. **(García, 2010)**

1.3.1.5 Detección de bordes.

En la actualidad buena parte de los logos presentes en las transmisiones son semitransparentes o presentan algún grado de transparencia, debido a esto se dificulta en gran medida la aplicación de las técnicas para la detección de objetos vistas con anterioridad. Los logos mantienen sus contornos de forma relativamente estable en un gran número de fotogramas por lo que un algoritmo capaz de detectar dichos contornos sería capaz de discriminar la información que no resulte relevante antes de aplicar alguna técnica de detección.

El algoritmo de Canny es usado para detectar todos los bordes existentes en una imagen. Este algoritmo está considerado como uno de los mejores métodos de detección de contornos mediante el empleo de máscaras de convolución y basado en la primera derivada. Los puntos de contorno son como zonas de píxeles en las que existe un cambio brusco de nivel de gris. En el tratamiento de imágenes, se trabaja con píxeles, y en un ambiente discreto, es así que en el algoritmo de Canny se utiliza máscaras, las cuales representan aproximaciones en diferencias finitas. **(Rebaza, 2010)**

Criterios del Algoritmo de Canny

En 1986, Canny propuso un método para la detección de bordes, el cual se basaba en tres criterios, estos son:

- Un criterio de detección expresa el hecho de evitar la eliminación de bordes importantes y no suministrar falsos bordes.
- El criterio de localización establece que la distancia entre la posición real y la localizada del borde se debe minimizar.
- El criterio de una respuesta que integre las respuestas múltiples correspondientes a un único borde.

Algoritmo de Canny para la detección de bordes

Uno de los métodos relacionados con la detección de bordes es el uso de la primera derivada, la que es usada por que toma el valor de cero en todas las regiones donde no varía la intensidad y tiene un valor constante en toda la transición de intensidad. Por tanto un cambio de intensidad se manifiesta como un cambio brusco en la primera derivada, característica que es usada para detectar un borde, y en la que se basa el algoritmo de Canny **(Rebaza, 2010)**.

El algoritmo de Canny consiste en tres grandes pasos:

- Obtención del gradiente: en este paso se calcula la magnitud y orientación del vector gradiente en cada píxel.
- Supresión no máxima: en este paso se logra el adelgazamiento del ancho de los bordes, obtenidos con el gradiente, hasta lograr bordes de un píxel de ancho.
- Histéresis de umbral: en este paso se aplica una función de histéresis basada en dos umbrales; con este proceso se pretende reducir la posibilidad de aparición de contornos falsos.

Obtención del gradiente

Para la obtención del gradiente, lo primero que se realiza es la aplicación de un filtro gaussiano a la imagen original con el objetivo de suavizar la imagen y tratar de eliminar el posible ruido existente. Sin embargo, se debe de tener cuidado de no realizar un suavizado excesivo, pues se podrían perder detalles de la imagen y provocar un pésimo resultado final. Este suavizado se obtiene promediando los valores de intensidad de los píxeles en el entorno de vecindad con una máscara de convolución de media cero y desviación estándar σ . Una vez que se suaviza la imagen, para cada píxel se obtiene la magnitud y módulo (orientación) del gradiente, obteniendo así dos imágenes **(Rebaza, 2010)**.

Supresión no máxima al resultado del gradiente

Las dos imágenes generadas en el paso anterior sirven de entrada para generar una imagen con los bordes adelgazados. El procedimiento es el siguiente: se consideran cuatro direcciones identificadas por las orientaciones de 0° , 45° , 90° y 135° con respecto al eje horizontal. Para cada píxel se encuentra la dirección que mejor se aproxime a la dirección del ángulo de gradiente. Posteriormente se observa si el valor de la magnitud de gradiente es más pequeño que al menos uno de sus dos vecinos en la dirección del ángulo obtenida en el paso anterior. De ser así se asigna el valor 0 a dicho píxel, en caso contrario se asigna el valor que tenga la magnitud del gradiente. La salida de este segundo paso es la imagen con los bordes adelgazados, después de la supresión no máxima de puntos de borde. **(Rebaza, 2010)**.

Histéresis de umbral a la supresión no máxima

La imagen obtenida en el paso anterior suele contener máximos locales creados por el ruido. Una solución para eliminar dicho ruido es la histéresis del umbral. El proceso consiste en tomar la imagen obtenida del paso anterior, tomar la orientación de los puntos de borde de la imagen y tomar dos umbrales, el primero

más pequeño que el segundo. Para cada punto de la imagen se debe localizar el siguiente punto de borde no explorado que sea mayor al segundo umbral. A partir de dicho punto seguir las cadenas de máximos locales conectados en ambas direcciones perpendiculares a la normal del borde siempre que sean mayores al primer umbral. Así se marcan todos los puntos explorados y se almacena la lista de todos los puntos en el contorno conectado. Es así como en este paso se logra eliminar las uniones en forma de Y de los segmentos que confluyan en un punto. **(Rebaza, 2010).**

1.3.1.6 Utilización de marcas de agua.

Un sistema de marcas de agua involucra un proceso de marcado y otro de detección que, generalmente, requieren una clave de propósito similar a la clave utilizada en los sistemas criptográficos. El nivel de disponibilidad de la clave, determinará quién o quiénes podrán leer o detectar la marca de agua. En la práctica, la mayoría de las técnicas de marcas de agua pueden considerarse como sistemas criptográficos simétricos, en los que se emplea una sola clave, variando en ellos el nivel de acceso a esa clave. **(López, 2011)**

Las marcas de agua deben su robustez a que no son perceptibles por los humanos, además estas deben estar implementadas de forma tal que no se afecten tras la conversión, redimensionamiento o modificación del archivo original. Su fortaleza no debe radicar en la longitud de la clave que se utilice sino en el algoritmo de cifrado.

La imperceptibilidad o transparencia de la marca tiene como base el comportamiento del sistema perceptual humano. Una marca de agua es imperceptible, si la degradación que causa en los archivos donde se ha insertado es muy difícil de apreciar. Este concepto se contrapone al de la robustez, si se tiene en cuenta que un sistema robusto debe insertar la marca en las regiones perceptualmente significativas del archivo. En algunas aplicaciones se puede aceptar una pequeña degradación de los datos, a cambio de lograr mayor robustez o menor costo del sistema.

La indetectabilidad está relacionada con el modelo estadístico del archivo antes y después de ser marcado. Se dice que la marca es indetectable si después de haberla insertado, el archivo marcado conserva las mismas propiedades estadísticas que su original. Lo que quiere decir que una persona no autorizada no podrá detectar la presencia de la marca utilizando métodos estadísticos. Esta propiedad es muy deseable en el caso de las comunicaciones encubiertas en las que el principal objetivo es ocultar la presencia del mensaje incrustado en el archivo. **(López, 2011)**

1.4 Descripción actual del dominio del problema

La digitalización audiovisual es un proceso de vital importancia para cualquier televisora del mundo en nuestros días y la monitorización automática de los canales de televisión resulta imprescindible para los gobiernos. Debido a esto en la Universidad de las Ciencias Informáticas se ha decidido implementar un sistema que resuelva esta problemática, para ello surge el proyecto Captura y Catalogación de Medias con el objetivo de obtener un producto personalizable para cualquier cliente interesado en el mismo.

Como resultado de este proyecto se obtuvo el producto Sistema de Captura y Catalogación de Medias (SCCM) y este cuenta con un módulo dedicado a la monitorización automática de señales de televisión. Este módulo se encargará de la detección de comerciales y la monitorización constante de forma automatizada de los canales que se le asignen, para realizar esta tarea necesita conocer la identidad de dichos canales y para ello se auxilia de la detección automática de su logo.

En un futuro, sería conveniente la implementación de un algoritmo para la detección de marcas de agua, que aumentaría la fiabilidad y eficiencia del módulo ante posibles cambios de logo por parte de los canales emisores, así como la capacidad de detectar estos nuevos logos de forma automatizada.

1.5 Situación Problemática

La monitorización de canales de televisión es una tarea que se realiza de forma ineficiente si es realizada manualmente porque el número de canales ha aumentado continuamente con el avance de las tecnologías de la información y la digitalización de la televisión. Resulta necesario conocer de forma automatizada o con la mínima interacción humana posible la identidad del canal que transmite así como el contenido transmitido para de este modo ser capaz de aplicar las leyes vigentes en el país que se esté utilizando el producto sobre el tiempo establecido para la publicidad.

También es necesario realizar anotaciones automáticas sobre el contenido del material que se analice en cuanto a la naturaleza del mismo y para ello es de gran utilidad la detección del logotipo presente en este porque hay gran variedad de televisoras especializadas en diversas temáticas tales como música, deportes, noticias, documentales etc. que con su correcta identificación puede brindar la temática que aborda el material.

La presencia de estas características de monitorización automatizada resulta altamente deseable en cualquier producto dedicado a la digitalización y control de medios audiovisuales. SCCM es un producto

que pretende insertarse en el mercado para satisfacer las necesidades de monitorización de señales de TV en empresas dedicadas a esta actividad, por tanto su correcta y eficiente implementación es imprescindible para la aceptación del mismo por parte de los clientes y que este sea capaz de competir con las demás soluciones de este tipo que existen en el mercado.

1.6 Análisis de otras soluciones existentes

En el mundo de la digitalización de archivos audiovisuales y la monitorización de señales de televisión existen diversos productos que realizan esta tarea. La principal motivación para implementar uno con similares características radica en que estos solo funcionan bajo tecnologías privativas en su gran mayoría y su implantación redundante en un alto coste. A continuación se mostrarán los principales exponentes de estos productos.

TdTarsys

Tarsys es una potente aplicación orientada al usuario que asegura que cualquier contenido, donde quiera que esté almacenado, pueda ser consultado por los usuarios como un único archivo. En combinación con una gestión de cintas inteligente, hace que el concepto de archivo casi en línea sea indiferente: los usuarios pueden tratar el contenido de la cinta como una parte accesible del almacén, siendo transparente para ello la ubicación de los mismos.

El contenido que está en el sistema de archivo, es indexado en la ingesta generándose una copia proxy en baja resolución. Se puede realizar cualquier trabajo sobre esta copia proxy de baja resolución minimizando así las transferencias de red. De esta forma, sólo aquellas partes del archivo en alta calidad que sean necesarias serán las recuperadas, maximizando así la productividad. Soporta cualquier formato de media y encapsulado: DV, MPEG, MXF, IMX, Quick Time, WMV y H264. Posee herramientas de edición al corte en el servidor para todos los formatos de media soportados y cuenta con generación automática del storyboard. **(Tedral, 2011)**

Videoma Broadcast Monitor (VBM)

Incorpora la grabación y monitorización multicanal de la señal de Televisión Digital Terrestres (TDT), satelital, multicast, frecuencias analógicas y CCTV. Además de ello realiza el streaming, clipping, catalogación, búsqueda, visionado y almacenamiento del contenido en tiempo real. VBM realiza funciones básicas de gran ayuda para una posterior edición. Visionado, generación de clips para la creación de

EDLs (Edit Decision Lists), inserción de marca de agua y efectos, asociación de documentos (Word, Pdf, PowerPoint, Excel) al contenido, extracción de fotogramas procedentes de un vídeo... etc. La catalogación de media se organiza de forma jerarquizada por esquemas o plantillas de catalogación, como núcleo principal del que derivan los metadatos. VBM permite editar los metadatos. Posee una herramienta de control informativo en campañas electorales. Auditoría y seguimiento de mítines, spots, propaganda política, marketing político. **(VBM, 2011)**

TD Indexer

Es un módulo de indexación y catalogación automática. Entre sus posibilidades está la búsqueda avanzada, reconocimiento de patrones, la compatibilidad con diversidad de formatos, así como su catalogación y acceso inmediato a cualquier fragmento del archivo. Indexer trabaja con los principales formatos contenedores de media actuales: MXF, QuickTime, ISO-File Format, AVI, ASF, ProgramStream, TransportStream permitiendo el desencapsulado y encapsulado de la media contenida en un formato a otro, sin variar la esencia ni realizar transformaciones innecesarias, además también trabaja con las principales codificaciones de video y audio actuales: MPEG2, DV, H264, AAC, WMV, Flash permitiendo la decodificación y codificación de un tipo en otro.

TD Indexer agrupa gran variedad de herramientas para el análisis automático y la indexación de video, audio e imágenes digitales. Desarrollado en base a estándares como el MPEG-7 y MPEG-21 para el control de metadatos técnicos y protección de contenidos, TD Indexer incluye entre otras funciones la generación de copias en baja resolución (proxy), obtención de storyboards, transcodificación de formatos y detección de contenidos en base a huellas digitales. **(Tedral, 2011)**

Los diversos productos existentes analizados con anterioridad poseen implementado de algún modo las funcionalidades para el procesamiento automático de los materiales con los que trabajan. Su principal desventaja radica en que la implementación de sus tecnologías para servidores está sobre plataformas privativas a pesar de ser sus clientes aplicaciones web y su alto coste es también una limitante importante.

1.7 Conclusiones

En el presente capítulo se ha podido apreciar que en la detección de patrones se utilizan diversas técnicas, pero no todas son aplicables a la detección de logos. Para esta tarea en específico según lo visto anteriormente pueden ser utilizadas las Integrales Proyectivas y el Template Matching porque permiten identificar un patrón definido con anterioridad, pero para poder lograr un mejor balance entre

falsos positivos y falsos negativos el Template Matching ofrece los mejores resultados a pesar de su alto costo computacional porque en la aplicación de las proyecciones es posible perder información relevante y por tanto errar en la detección. Se evidenció además que la presencia de características de monitorización automática es imprescindible en este tipo de productos puesto que los líderes a nivel mundial en este campo las traen implementadas.

CAPÍTULO 2: Descripción y diseño de algoritmos y componente para la detección de logos en materiales audiovisuales.

2.1 Introducción

En este capítulo se realiza un estudio sobre las tecnologías actuales para el desarrollo de aplicaciones orientadas hacia el procesamiento digital de imágenes. Se tratan temas como tecnologías que soportan el componente, lenguaje de programación y algoritmos utilizados. El propósito de dicho estudio es dar a conocer cuáles tecnologías se utilizarán y por qué, además se explicarán más a fondo los algoritmos utilizados.

2.2 Tecnologías y tendencias en el procesamiento digital de imágenes

El procesamiento digital de imágenes es un campo que se abre paso cada vez con más fuerza en el mundo de hoy. Este se ve reflejado desde las aplicaciones de video vigilancia, reconocimiento de objetos en contenidos audiovisuales así como en el procesamiento automatizado de información. Para lograr dotar al programador de este tipo de aplicaciones, con las herramientas básicas para comenzar a desarrollar una aplicación de este tipo, existen diversas bibliotecas que brindan las funciones para el procesamiento digital imágenes y entre las más importantes se encuentran la Open Computer Vision (OpenCV) desarrollada por Intel y The Visualization ToolKit (VTK).

2.3 Lenguaje de Programación

Las técnicas y algoritmos para el procesamiento digital de imágenes consumen gran parte de la capacidad de procesamiento de la computadora donde se utilicen por lo que resulta imprescindible aprovechar este recurso al máximo. Para la realización del componente se escogió el lenguaje de programación C++ porque este posee una serie de ventajas para el trabajo de procesamiento aprovechando mejor las potencialidades de la PC donde se programa, además fue escogido por los arquitectos del departamento Señales Digitales para el desarrollo de aplicaciones de escritorio.

2.3.1 C++

El C++ es a la vez un lenguaje procedural (orientado a algoritmos) y orientado a objetos. Como lenguaje procedural se asemeja al C y es compatible con él, aunque ya se ha dicho que presenta ciertas ventajas.

Como lenguaje orientado a objetos se basa en una filosofía completamente diferente, que exige del programador un completo cambio de mentalidad. Las características propias de la Programación Orientada a Objetos de C++ son modificaciones mayores que sí cambian radicalmente su naturaleza. Los programas escritos en C o C++ tienen otras ventajas sobre el resto. Con la excepción del ensamblador, generan los programas más compactos y rápidos. El código es portable, es decir, un programa ANSI en C o C++ podrá ejecutarse en cualquier máquina y bajo cualquier sistema operativo. Y si es necesario, proporcionan un acceso a bajo nivel de hardware sólo igualado por el ensamblador. **(UsabilidadWeb, 2011)**

2.3.2 Framework Qt 4.7

Qt es un framework multiplataforma para desarrollar interfaces gráficas de usuario y para el desarrollo de programas sin interfaz gráfica como herramientas de la consola y servidores. Es producido por la división de software Qt de Nokia de la empresa noruega Trolltech.

Qt es utilizada en KDE, un entorno de escritorio para sistemas como GNU/Linux o FreeBSD, entre otros. Utiliza el lenguaje de programación C++ de forma nativa, adicionalmente se le pueden incorporar otros lenguajes de programación y funciona en las principales plataformas. El API de la biblioteca cuenta con métodos para acceder a bases de datos mediante SQL, así como uso de XML, gestión de hilos, soporte de red y una API multiplataforma unificada para la manipulación de archivos. Cuenta además con disímiles estructuras de datos tradicionales. Distribuida bajo los términos de GNU Lesser General Public License (LGPL), QT es software libre y de código abierto.

2.4 Entorno de Desarrollo Integrado (IDE).

Para el desarrollo de aplicaciones utilizando el framework Qt Nokia ofrece el QtCreator, un IDE multiplataforma.

Características principales:

- Posee un avanzado editor de código C++.
- Posee también una GUI¹ integrada y diseñador de formularios.
- Herramienta para proyectos y administración.

¹ Graphic User Interface (Interfaz Grafica de Usuario).

- Ayuda integrada sensible al contexto.
- Depurador visual.
- Resaltado y auto-completado de código.
- Soporte para refactorización de código.

Qt Creator es distribuido bajo tres tipos de licencias: Qt Commercial Developer License, Qt GNU LGPL v. 2.1, Qt GNU GPL v. 3.0 y está disponible para las plataformas: Linux, Mac OSX; Windows, Windows CE, Symbian y Maemo. **(Pixelco, 2011)**

2.5 Bibliotecas utilizadas.

Las bibliotecas de clases son de gran utilidad para la programación de cualquier tipo de aplicaciones puesto que estas contienen un conjunto de funcionalidades ya implementadas que abstraen al programador de realizar las funciones más básicas. Estas agilizan en gran medida el desarrollo que de otro modo sería en extremo complicado y tedioso.

2.5.1 OpenCV

OpenCV (Open Source Computer Vision Library) es una biblioteca de código abierto que proporciona un gran número de funciones para el procesamiento de imágenes. Permite a los programadores crear aplicaciones poderosas en el dominio de la visión digital. Ofrece muchos tipos de datos de alto nivel como juegos, árboles, gráficos, matrices, entre otros. Esta biblioteca incluye funcionalidades como: realizar operaciones básicas, procesado de imágenes, análisis estructural, análisis de movimiento, reconstrucción 3d, calibración de la cámara, segmentación y reconocimiento de objetos, entre otras.

Implementa gran variedad de herramientas para la interpretación de la imagen. Además de incluir operaciones primitivas como binarización, filtrado, estadísticas de la imagen y pirámides. Los algoritmos están basados en estructuras de datos muy flexibles y usa la estructura Iplimage para crear y manejar imágenes. Además brinda funciones que posibilitan convertir una imagen en diferentes formatos de color. En cuanto a seguimiento de objetos y análisis de movimiento, brinda una funcionalidad interesante donde incorpora funciones básicas para el modelado de fondos en su posterior sustracción. Genera imágenes de movimiento MHI (Motion History Images) para determinar dónde hubo movimiento y en qué dirección. **(Intel, 2003)**

2.6 El Lenguaje Unificado de Modelado (UML) como soporte de la modelación de la solución propuesta

UML es un lenguaje para visualizar, especificar, construir, y documentar los artefactos que se crean durante el proceso de desarrollo. Es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. No es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. UML no es método, ni una metodología

La representación en UML de un software está formada por las 4+1 vistas o modelos parciales separados, relacionados entre sí, estas vistas son:

- Vista de casos de uso.
- Vista lógica.
- Vista de procesos.
- Vista de implementación.
- Vista de despliegue.

UML es un lenguaje gráfico con sintaxis y semántica bien definidas. La sintaxis de la notación gráfica se especifica mediante su correspondencia con los elementos del modelo semántico subyacente, cuya semántica se define por medio de un meta-modelo, textos descriptivos y restricciones (**Zarzuela, 2003**).

2.7 Requisitos funcionales

A continuación se presentan las capacidades o condiciones que el sistema debe cumplir:

R1. Identificar logo. El sistema permitirá detectar el logo de una televisora.

R2. Realizar anotaciones del video. El sistema debe permitir almacenar el tiempo de aparición y ocultamiento de un logo.

R3. Guardar logos no definidos. El sistema debe permitir la detección de logos que no se tengan guardados y que aparezca un alto número de veces en un video.

2.8 Diseño de un algoritmo para la detección de logos.

Para la realización de los algoritmos siguientes la técnica que se utilizará es el Template Matching, esta es muy sensible a los cambios de escala, por tanto, en la base de conocimientos de logos se encuentran almacenadas además de la imagen que se utiliza como plantilla, la resolución del video del que fue

extraída la misma, para poder escalarla con bastante aproximación a la resolución del video que se esté procesando y de este modo poder obtener buenos resultados con esta técnica.

2.8.1 Diseño de un algoritmo para Identificar video (Anexo 1).

- 1- Extraer fotograma del video mediante las funciones que brinda el OpenCV.
- 2- Establecer las regiones de interés del fotograma en análisis para limitar la búsqueda en esta área.
- 3- Aplicar filtros a la imagen con el objetivo de mejorar la visibilidad del logo en la misma (Bordes, Dilatación).
- 4- Aplicar Template Matching al fotograma utilizando cada uno de los logos hasta encontrar una coincidencia.
- 5- Si se encontró retornar el nombre de la televisora asociada al logo. En caso contrario capturar el próximo fotograma y volver al paso 4.
- 6- Si es el fin del video terminar.

El objetivo de este algoritmo radica en la necesidad de identificar el material en cuestión de manera rápida. Se basa en la premisa de que cada material posee el logo de una sola televisora y si esta falla el resultado del mismo no sería confiable porque retornaría la primera coincidencia positiva. Está pensado para permitir la clasificación rápida de los materiales.

2.8.2 Diseño de un algoritmo exhaustivo para realizar anotaciones en el video (Anexo 2).

- 1- Extraer fotograma del video mediante las funciones que brinda el OpenCV.
- 2- Establecer las regiones de interés del fotograma en análisis para limitar la búsqueda en esta área.
- 3- Aplicar filtros a la imagen con el objetivo de mejorar la visibilidad del logo en la misma (Bordes, Dilatación).
- 4- Aplicar Template Matching al fotograma utilizando cada uno de los logos en BD hasta encontrar una coincidencia.
- 5- Si se encontró anotar como aparición del logo. Si no volver al paso 1
- 6- Capturar próximo fotograma hasta que no se detecte el logo.
- 7- Cuando no se detecte el logo anotar tiempo de ocultamiento.

8- Volver al paso 1 hasta que finalice el video.

Este algoritmo tiene como objetivo realizar un análisis del video de forma exhaustiva comparando en cada fotograma con todos los logos de la base de conocimientos. De este modo es posible realizar las anotaciones de los tiempos de aparición y ocultamiento de los logos aún en el caso de que el material esté compuesto por fragmentos de varias televisoras y por tanto presente más de un logo en su contenido. Al ser necesaria una comparación con todos los logos la ejecución del mismo será bastante lenta.

2.8.3 Diseño de un algoritmo rápido para realizar anotaciones en el video (Anexo 3).

- 1- Identificar video.
- 2- Extraer fotograma del video.
- 3- Establecer las regiones de interés del fotograma en análisis para limitar la búsqueda en esta área.
- 4- Aplicar filtros a la imagen con el objetivo de mejorar la visibilidad del logo en la misma (Bordes, Dilatación).
- 5- Aplicar Template Matching al fotograma utilizando el logo identificado en el paso 1.
- 6- Si se encontró anotar como aparición del logo. Si no volver al paso 1
- 7- Capturar próximo fotograma hasta que no se detecte el logo.
- 8- Cuando no se detecte el logo anotar tiempo de ocultamiento.
- 9- Volver al paso 1 hasta que finalice el video.

Este algoritmo suple las deficiencias en cuanto a velocidad del anterior partiendo de que el material solo posee el logo de una televisora. Permite anotar los tiempos de aparición y ocultamiento del logo en la transmisión de forma mucho más rápida pues sólo realiza la comparación con el logo que se identificó en el primer paso pero de igual modo que el algoritmo para identificar video su resultado solo es aplicable a materiales de una sola televisora.

2.8.4 Diseño de un algoritmo para guardar logos desconocidos (Anexo 4).

- 1- Extraer los fotogramas que componen el video.
- 2- Establecer las regiones de interés del fotograma en análisis.
- 3- Aplicar filtros a la imagen.
- 4- Aplicar detección de bordes a los fotogramas.

5- Si la imagen resultante contiene un área cuyos bordes no cambian por un tiempo considerable almacenar esta área de la imagen original como nuevo logo.

Este algoritmo permite identificar en la imagen una zona de contornos estables la cual tiene una alta probabilidad de ser un logo y guarda este fragmento de la imagen como un nuevo logo en la base de datos.

2.9 Conclusiones

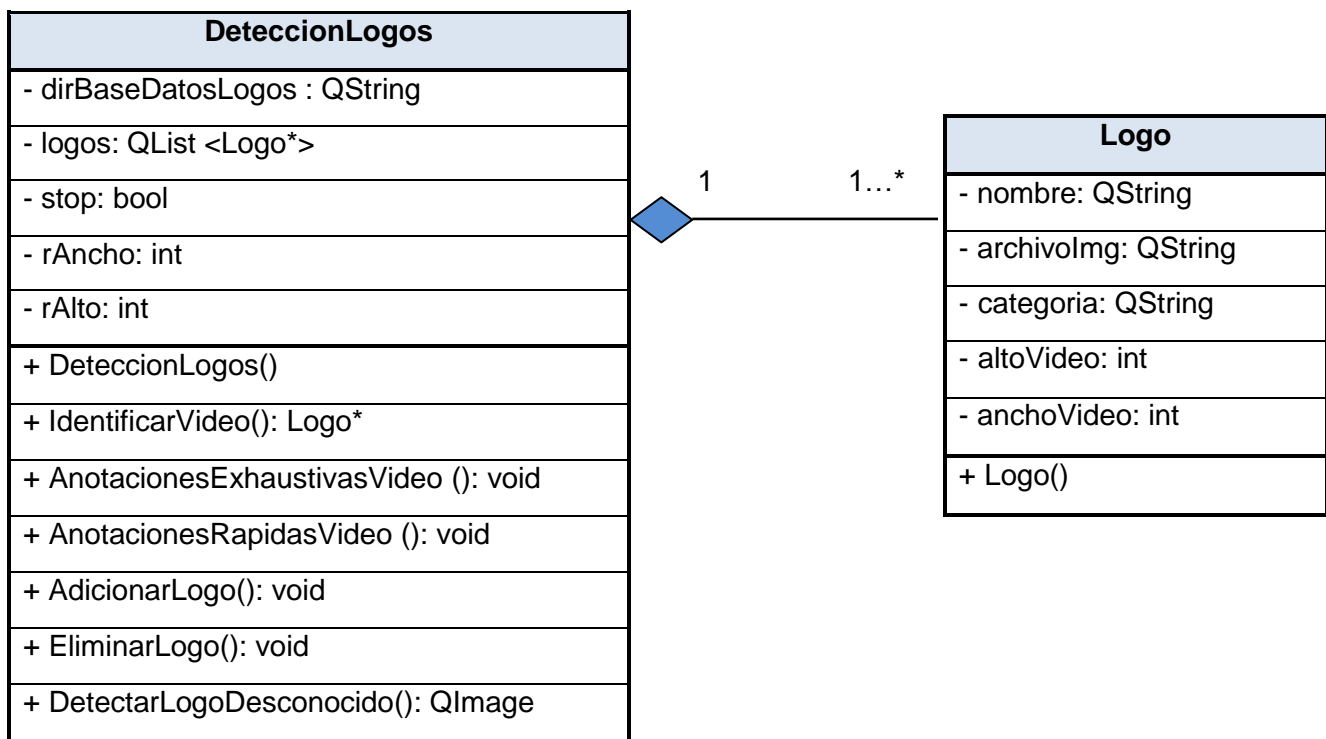
En el presente capítulo se ha podido apreciar que resulta imprescindible el aprovechamiento de los recursos de la computadora, fundamentalmente el microprocesador, en las aplicaciones que realicen procesamiento digital de imágenes, de ahí la importancia de la selección del lenguaje de programación C++ para la implementación del componente. Además se evidenció la importancia de tener en cuenta las limitaciones de la técnica escogida, como son los cambios de escala en el Template Matching, para poder actuar en consecuencia y solucionarlas.

CAPÍTULO 3: Implementación y Pruebas de algoritmos y componente para la detección de logos en materiales audiovisuales.

3.1 Introducción

En el presente capítulo se realiza una descripción detallada de la solución propuesta, se muestra el diagrama de clases generado y se describen cada una de sus funcionalidades. Se presentan además las pruebas realizadas al componente y los algoritmos diseñados para lograr su validación.

3.2 Diagrama de clases del componente para la detección de logos.



3.2.1 Descripción de las clases del componente para la detección de logos.

- **DeteccionLogos**

Atributos:

dirBaseDatosLogos:

Este atributo almacena la dirección donde se encuentra la carpeta que contiene los logos que el sistema tiene registrados.

logos:

Este atributo contiene una lista de los logos que el sistema tiene en su base de conocimientos.

stop:

Este atributo se utiliza para informar a los métodos que deben detenerse antes de que finalice normalmente su ejecución.

rAlto, rAncho:

Estos atributos se utilizan para almacenar el valor de escalado de una plantilla en caso de que no coincida la resolución del video que se procesa con el video de donde fue extraída originalmente.

Métodos:**DeteccionLogos():**

El constructor de la clase se encarga de inicializar el objeto, cargando desde un fichero de configuración la dirección de la carpeta donde se encuentran los logos del sistema y realiza el llenado de la lista de logos.

IdentificarLogo(pArchivoVideo: QString, preview: bool):

Este método, a partir de un video pasado por parámetro devuelve el logo de la televisora asociada al logo que detecte en el video. Este método además utiliza los atributos rAlto y rAncho para almacenar los valores de la plantilla si esta fue escalada, para que puedan ser utilizados posteriormente por el método AnotacionesVideoRapido. El umbral que utiliza este método para aplicar el Template Matching es de 0.3 permitiéndole obtener una mayor exactitud en el proceso de identificación y de este modo evitar una identificación errónea. Posee además el parámetro preview que le indica si debe mostrar la imagen con el resultado de la identificación al terminar su ejecución.

AnotacionesExhaustivasVideo(pArchivoVideo: QString, pArchivoSalida: QString, preview: bool, pBD: bool):

Este método, partiendo de un video pasado por parámetro, escribe en un fichero, pasada también su dirección por parámetro, los tiempos de aparición y ocultamiento de los logos detectados, o puede escribirlos directamente en la base de datos del producto SCCM en dependencia de si es verdadero o no el atributo pBD. Se utiliza en el caso de que el material sea producto de fragmentos de múltiples televisoras porque su ejecución es significativamente más lenta. En este caso el umbral para la aplicación

del Template Matching es de 0.4 pues de este modo se consigue una mejor relación entre los falsos positivos y los falsos negativos propiciando que en ocasiones puedan aparecer falsos positivos pero aumentando la detección en la mayoría de los casos y disminuyendo de este modo los falsos negativos. El atributo preview en este caso le indica al método si debe mostrar en una ventana la imagen que está procesando y resaltar el logo si este fue detectado.

AnotacionesRapidasVideo(pArchivoVideo: QString, pArchivoSalida: QString, preview: bool, pBD: bool):

Este método es similar al anterior en cuanto a los parámetros, pero a diferencia de este, llama al método IdentificarVideo, que le permite conocer que logo tiene que buscar en el video lo que agiliza en gran medida su ejecución limitándolo a materiales producidos por una sola televisora. En este caso el umbral utilizado para el Template Matching fue de 0.5 para mejorar aún más la detección pues al comparar con un solo logo existe un chance aun menor de producir falsos negativos.

AdicionarLogo(pNombre: QString, pCategoría: QString, pArchivolmg: QString, pAltoV: int, pAnchoV: int):

Este método permite insertar un nuevo logo al componente, tomando como parámetros el nombre de la televisora, la categoría de la misma así como la dirección del archivo de imagen que contiene el logo, necesitando también las dimensiones del video de donde fue tomado el logo. Utilizando estos datos crea un nuevo logo y lo agrega a la base de logos del sistema.

EliminarLogo(pNombre: QString):

El método se encarga de eliminar un logo de la base de logos del sistema utilizando como parámetro el nombre de la televisora del mismo.

DetectarLogoNoDefinido(pArchivoVideo: QString):

Este método dado un video pasada su dirección por parámetro buscará la posible aparición de un logo en el mismo, en caso de encontrarlo devuelve la imagen del mismo.

▪ Logo

Atributos:

nombre:

Este atributo almacena el nombre de la televisora asociada a un logo.

archivolmg:

Este atributo almacena el nombre del fichero del logo de la televisora.

categoria:

Este atributo almacena la categoría asociada a la televisora en cuestión, por ejemplo noticias, documentales, series, etc.

anchoVideo, altoVideo:

Este par de atributos almacena la resolución del video de donde fue extraída la imagen del logos para poder escalar la plantilla con bastante aproximación y que se corresponda a la resolución del video que se está analizando.

Métodos:

Logo(nombre: QString, direccion: QString, categoria: QString, altoVideo: int, anchoVideo: int):

El constructor de la clase Logo se encarga de inicializar el objeto asignando a los atributos de este los pasados por parámetros.

3.3 Validación del componente y algoritmo para la detección de logos.

3.3.1 Pruebas de compatibilidad de formatos.

La carga de los videos por parte de la biblioteca OpenCV se realiza mediante su interacción con FFmpeg². Por tanto resulta necesario realizar las pruebas pertinentes para determinar qué formatos de video son admitidos por el componente para el procesamiento y cuáles no será posible procesar debido a que OpenCV es incapaz de cargarlos.

Contenedor	AVI
Codec Video	MPG4-ASP (XVID)
Resolución	624x352
fps	23.976

² FFmpeg es una solución multiplataforma para la grabación, conversión y emisión de audio y video. (FFmpeg, 2011)

Archivo	ABC.avi
Comentario	El componente es capaz de cargar el archivo de video sin dificultades completando el procesamiento.

Contenedor	OGM
Codec Video	MPG4-ASP (XVID)
Resolución	624x352
fps	23.976
Archivo	SyFy.ogm
Comentario	El componente es capaz de cargar el archivo de video sin dificultades completando el procesamiento.

Contenedor	MP4
Codec Video	MPG4-AVC (H264)
Resolución	704x400
fps	25
Archivo	A3.mp4
Comentario	El componente es capaz de cargar el archivo de video sin dificultades completando el procesamiento.

Contenedor	ASF
Codec Video	WMV9
Resolución	640x480
fps	29.97
Archivo	DSc.wmv
Comentario	El componente es capaz de cargar el archivo de video completando el procesamiento pero la notificación del porcentaje realizado no funciona correctamente debido a la incapacidad del OpenCV de detectar correctamente el número de frames del video.

Contenedor	MKV (Matroska)
Codec Video	MPG4-AVC (H264)
Resolución	624x352
fps	29.97
Archivo	AMC.mkv
Comentario	El componente no es capaz de cargar el archivo de video por lo que le es imposible realizar el procesamiento.

Contenedor	OGG
Codec Video	Theora
Resolución	624x352
fps	23.976
Archivo	CW.ogv
Comentario	El componente es capaz de cargar el archivo de video sin dificultades completando el procesamiento.

Contenedor	MPEG-PS
Codec Video	MPEG-1
Resolución	624x352
fps	23.976
Archivo	CW.mpg
Comentario	El componente es capaz de cargar el archivo de video completando el procesamiento pero la notificación del porcentaje realizado no funciona correctamente debido a la incapacidad del OpenCV de detectar correctamente el número de frames del video.

Estas pruebas de compatibilidad de formato arrojan como resultado que a pesar de ser compatible con una buena parte de los formatos de video habituales presenta algunos problemas con el Windows Media

Video y el MPEG pues no es capaz de leer correctamente el número de frames del archivo de video lo que impide la notificación correcta del porcentaje del análisis realizado. Además fue imposible cargar un video en un contenedor Matroska. También es recomendable que los archivos que se utilicen estén en buenas condiciones para evitar cualquier posible error debido a un fin abrupto del archivo o errores en su cabecera.

3.3.2 Pruebas al algoritmo de detección de logos.

Para la realización de las pruebas del algoritmo para la detección de logos se realizarán pruebas de Precision y Recall para posteriormente realizar un gráfico Precision vs Recall que permita conocer en qué medida resulta efectivo el algoritmo de detección.

Precision consiste en el número de verdaderos positivos detectados por el algoritmo divididos por el número total de elementos positivos detectados (la suma de los verdaderos positivos más los falsos positivos) y quedaría expresado de la siguiente forma.

$$Precision = \frac{vp}{vp + fp}$$

Una Precision de 1 significa que todos los resultados obtenidos son significativos lo que no quiere decir que se obtuvieran todos los que se encontraban en la muestra.

Recall consiste en el número de verdaderos positivos divididos entre el total de elementos positivos (la suma de los verdaderos positivos más los falsos negativos) quedando expresado de esta forma.

$$Recall = \frac{vp}{vp + fn}$$

Un Recall de 1 significa que se obtuvieron todos los resultados significativos lo que no significa que no se hayan obtenido también resultados no significativos.

Archivo	Total	vp	fp	Precision	Recall
ABC.avi	2109	1982	1	0,9994957	0,9397819
CW.avi	1845	1601	2	0,9987523	0,8677507
Syfy.avi	2417	2134	54	0,9753199	0,8829127
Discovery.avi	3151	3130	0	1	0,9933354
CTV2.avi	600	541	0	1	0,9016667

Antena3-3.avi	600	586	0	1	0,9766667
---------------	-----	-----	---	---	-----------

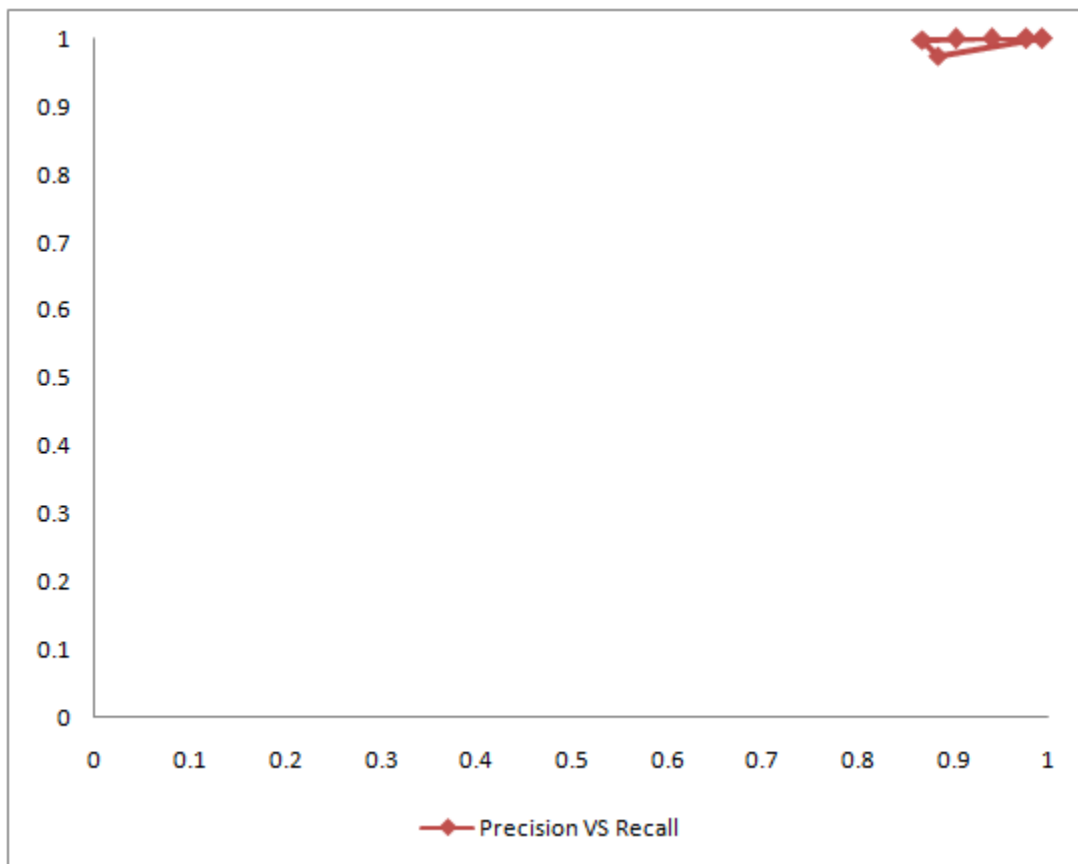
Leyenda:

Archivo: Archivo de video al que se realizó el procesamiento.

Total: Segundos totales de aparición del logo en el video.

vp: Segundos detectados correctamente por el algoritmo (verdaderos positivos).

fp: Segundos detectados incorrectamente por el algoritmo (falsos positivos).



3.4 Conclusiones

En el presente capítulo se ha llegado a la conclusión de que el componente es capaz de cargar los formatos de video más utilizados, aunque presenta problemas con los archivos del contenedor Matroska. Es necesario cuidar que los archivos de video que se utilicen estén con la calidad requerida evitando los archivos incompletos y de ser posible los MKV pues estos últimos no podrán ser procesados en absoluto. Además el componente posee un buen desempeño en la detección de logos demostrado por las pruebas de Precision vs Recall realizadas.

Conclusiones Generales

Con la realización de la investigación se arribó a las siguientes conclusiones:

- La detección efectiva de logos en videos es muy útil para el trabajo con medios audiovisuales porque permite la identificación y clasificación automática de estos.
- La técnica Template Matching puede ser utilizada para la detección de logos, obteniendo buenos resultados cuando se tienen en cuenta sus limitaciones y estas son resueltas.
- La implementación de este componente automatiza la tarea de detección de logos en transmisiones televisivas en el producto SCCM.
- La integración del componente con el producto SCCM posibilita que este se encuentre al nivel de otros existentes en el mundo pudiendo identificar y clasificar automáticamente los materiales que procesa.
- Se logró desarrollar un componente robusto que es capaz de trabajar con los formatos de video más utilizados y detectar los logos presentes en ellos con un porcentaje elevado de efectividad.

Recomendaciones

Se proponen las siguientes recomendaciones:

- Continuar el desarrollo del componente y aumentar la base de logos del mismo para que progresivamente este aumente su funcionalidad.
- Implementar un algoritmo con una técnica invariante a cambios de escala para aumentar la fiabilidad del componente de software desarrollado.
- Implementar la obtención del número de fotogramas del video utilizando una biblioteca externa al OpenCV para solucionar la notificación del porcentaje en los videos MPEG y WMV.

Referencias Bibliográficas

- Alonso, J. 1998. **Curso de metodología de la investigación. Santa Clara, Cuba. Facultad de Ciencias Sociales y Humanísticas. Universidad Central de las Villas. 1998.**
- Álvarez, I. 1997. **Curso de Investigación Científica. Santa Clara Cuba. Facultad de Ciencias Sociales y Humanísticas. Universidad Central de las Villas. 1997.**
- Bimbo. 2005. **2005.**
- Definición.de. 2011. **Definición.de. [En línea] 2011. <http://definicion.de/logo/>.**
- DefiniciónABC. 2011. **Definición ABC. [En línea] junio de 2011.**
- Elsevier Science. 2002. **Dictionary of Video and Television Technology. USA : s.n., 2002. ISBN 1-878707-99-X.**
- Error500. 2004. **Error500. [En línea] 2004.**
http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php.
- FFmpeg. 2011. **FFmpeg. [En línea] 2011. <http://www.ffmpeg.org>.**
- García, Ginés Mateos. 2010. **Universidad de Murcia. [En línea] 2010.**
<http://dis.um.es/~ginesgm/files/doc/pav/tema6.ppt>.
- González, R. C. and Woods, R. E. 2002. **Digital Image Processing. 2002.**
- Intel. 2003. **The Open Computer Vision Library. [En línea] 2003.**
<http://www.intel.com/software/products/opensource/libraries/cvfl.htm>.
- López, Amalia Beatriz Orúe. 2011. **[En línea] 2011.**
<http://www.iec.csic.es/cryptonomicon/articulos/expertos88.html>.
- Masip, David. 2011. **[En línea] 2011. <http://www.desarrolloweb.com/articulos/840.php>.**
- Nokia. **<http://qt.nokia.com/products/developer-tools/>. [En línea]**
- Pérez Noriega, Verónica y Delcor Ballesteros, Jordi. 2006. **Universidad UPC. [En línea] 2006.**
<http://upcommons.upc.edu/pfc/bitstream/2099.1/3855/1/54955-1.pdf>.
- Pixelco. 2011. **Pixelco. [En línea] 2011. <http://pixelcoblog.com/qt-creator-completo-entorno-de-desarrollo-multiplataforma/>.**

PostgreSQL.org. 2011. [En línea] 2011. http://www.postgresql.org.es/sobre_postgresql.

RAE. 2011. **DICCIONARIO DE LA LENGUA ESPAÑOLA**. [En línea] 2011. <http://buscon.rae.es/drae/>.

Rebaza, Jorge Valverde. 2010. **2010**.

Sáez Peña, Edmundo. 2006. **Segmentacion Automatica de Video. Tesis Doctoral. Universidad de Málaga, 2006**.

SourceForge. 2011. [En línea] 2011. <http://amide.sourceforge.net/help/es/ar01s05.html>.

Tedial. 2011. **Productos**. [En línea] 2011. <http://www.tedial.com/index.php?Itemid=64>.

Tedial. 2011. **Tarsys**. [En línea] 2011. <http://www.tedial.com/index.php?Itemid=66..>

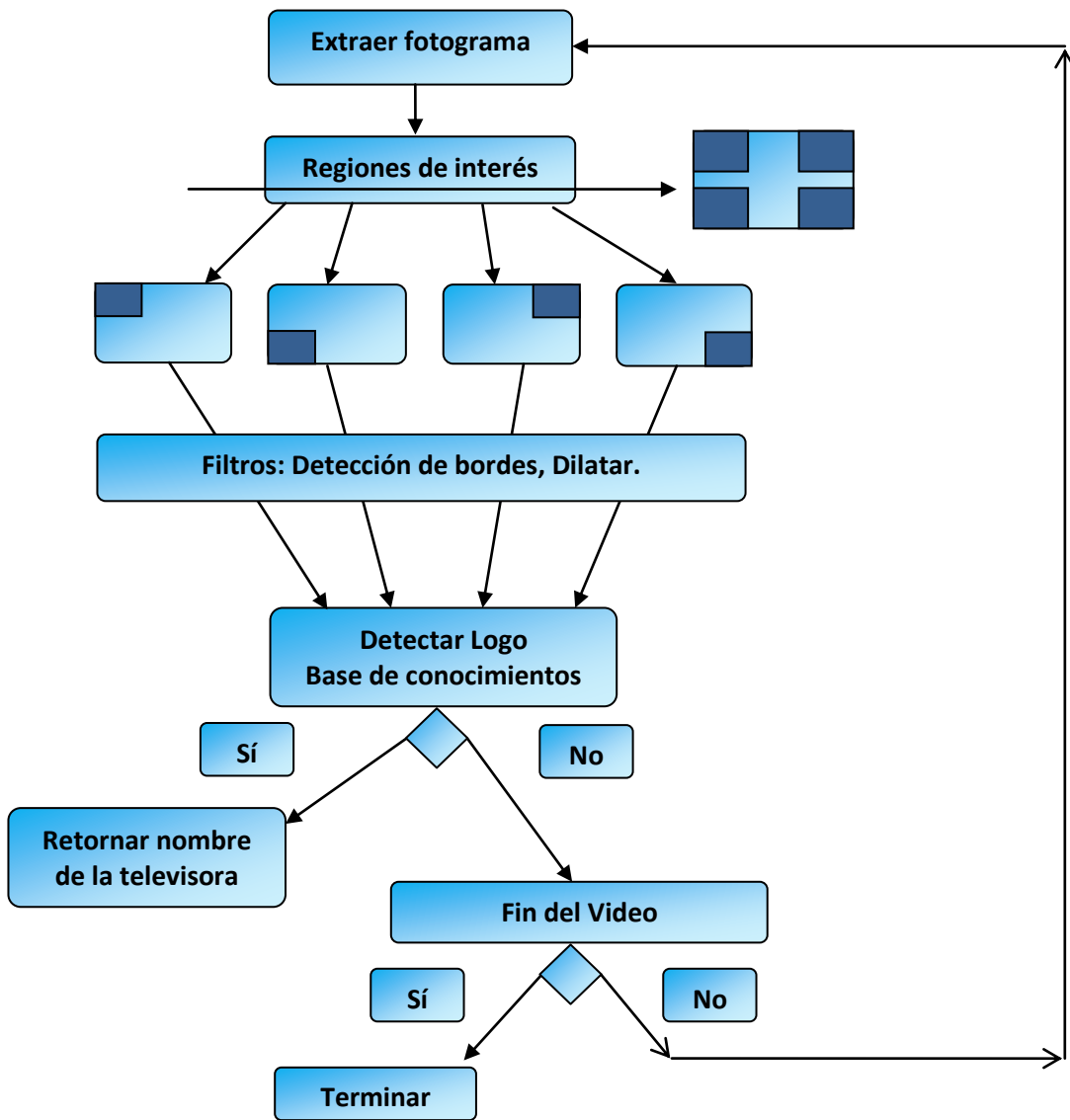
UPV. 2011. **Universidad Politécnica de Valencia**. [En línea] 2011. <http://futura.disca.upv.es/~imd/cursosAnteriors/2k8-2k9/videos/trabajandoConOpenCV/secciones/Interaccion.html>.

UsabilidadWeb. 2011. [En línea] 2011. <http://www.usabilidadweb.com.ar/cpp.php>.

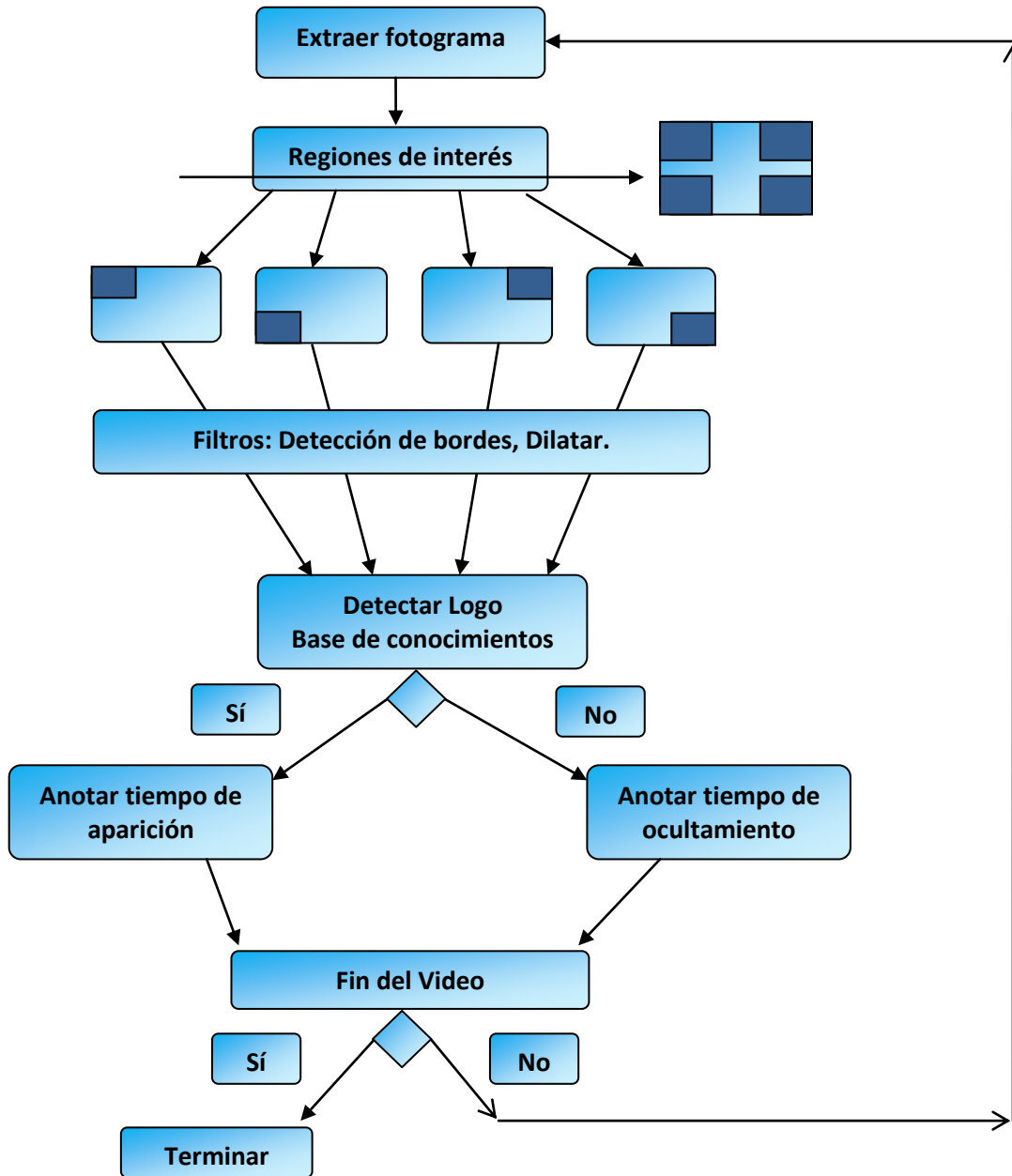
VBM. 2011. **Soluciones Videoma Broadcast Monitor**. [En línea] 2011. http://www.isid.es/spanish/product/producto_v01.htm.

Zarzuela, Jorge Ferrer. 2003. **Metodologías Ágiles**. [En línea] 2003. <http://libresoft.dat.escet.urjc.es/html/downloads/ferrer-20030312.pdf>.

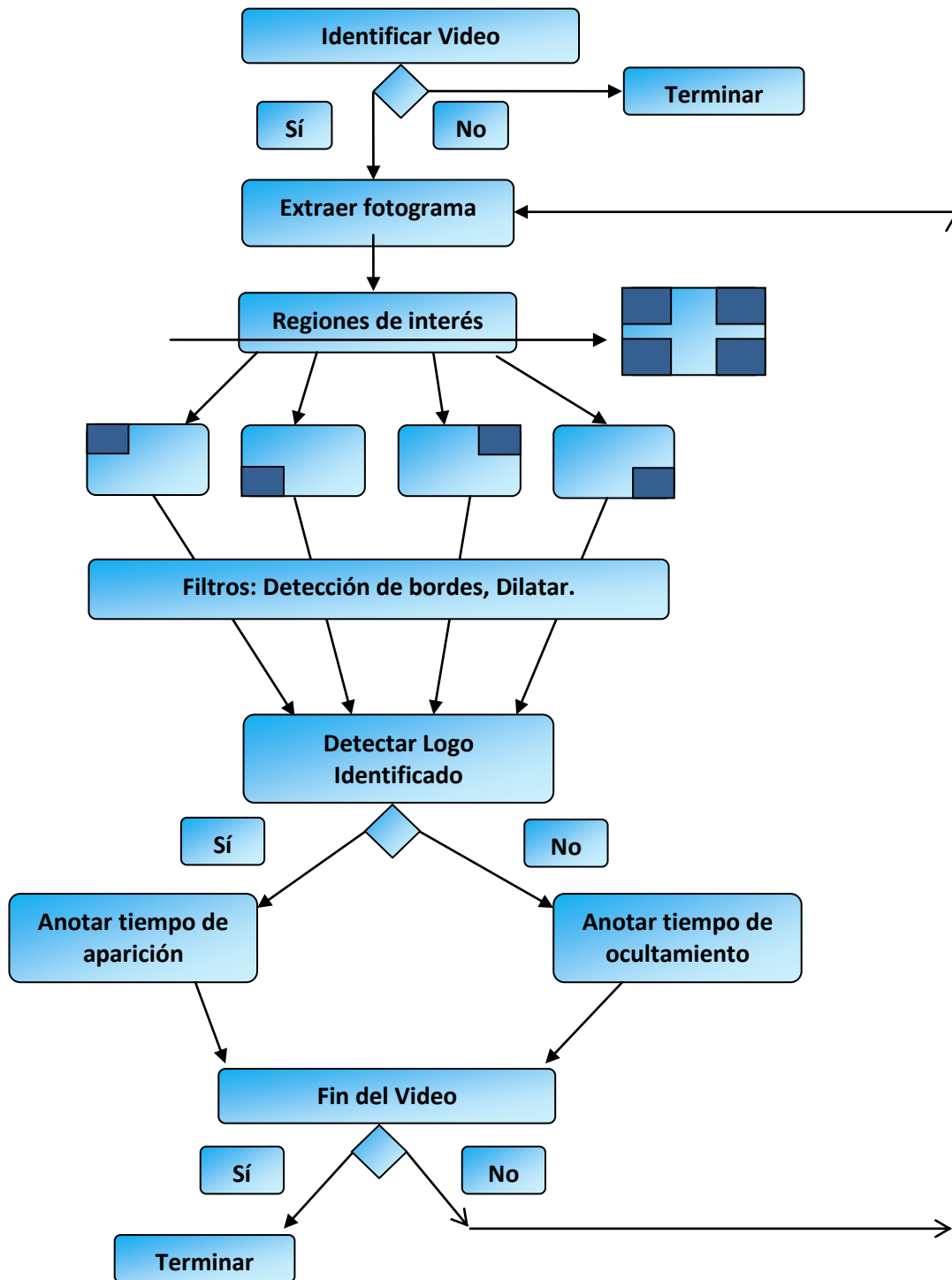
Anexo 1: Algoritmo para identificar video.



Anexo 2: Algoritmo exhaustivo para realizar anotaciones en el video.



Anexo 3: Algoritmo rápido para realizar anotaciones en el video.



Anexo 4: Algoritmo para guardar logos desconocidos.

