

Universidad de las Ciencias Informáticas

Facultad 6



Título:

**Componente para localizar regiones de subtítulos en
videos**

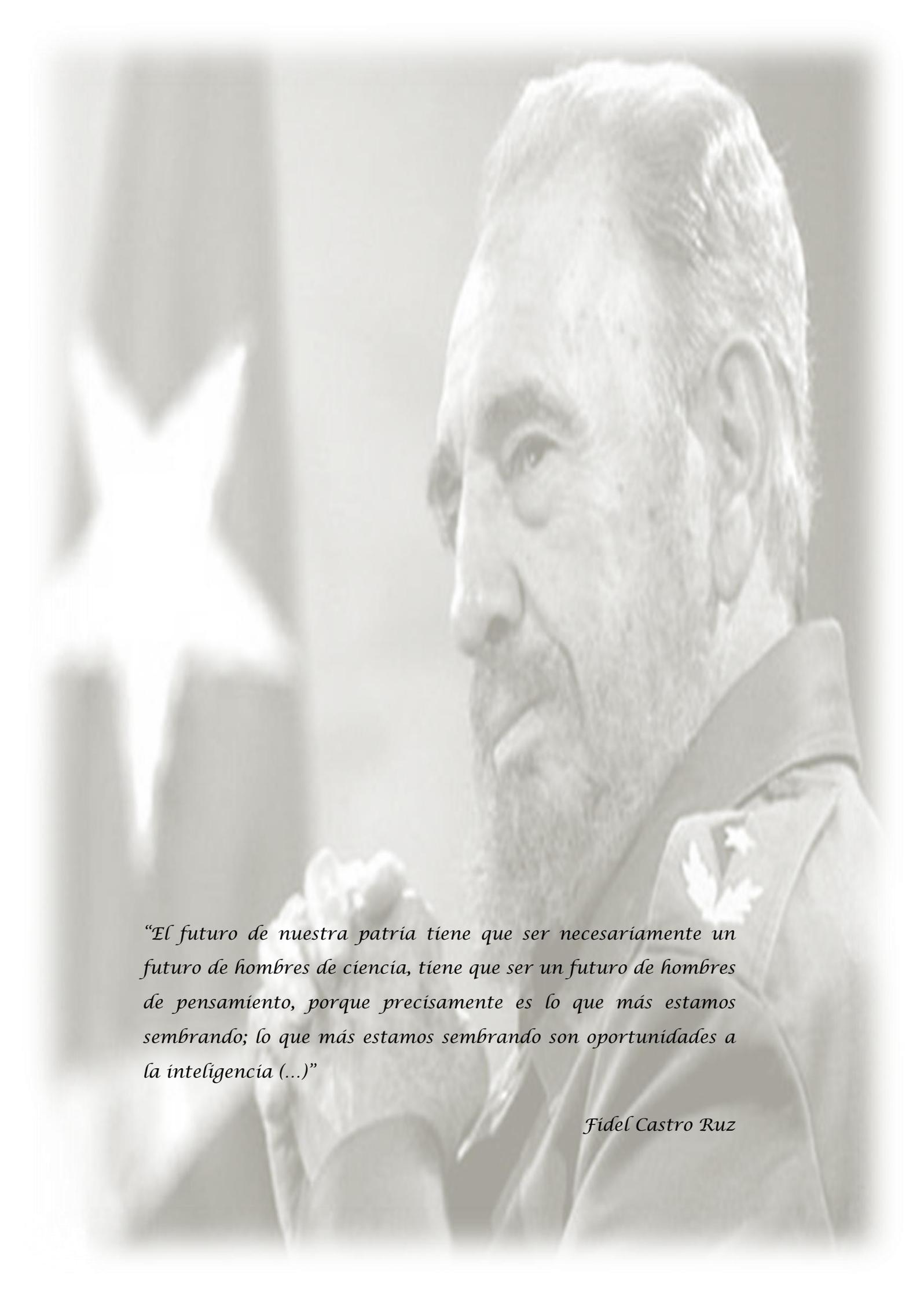
**Trabajo de Diploma para optar por el título de
Ingeniero Informático**

Autor: Alejandro Miguel Rodríguez Veranes.

Co-tutor: Msc. Rafael Leodán Cardero Álvarez.

Tutor: Ing. Jean Michael Suarez Pérez.

Junio 2012



“El futuro de nuestra patria tiene que ser necesariamente un futuro de hombres de ciencia, tiene que ser un futuro de hombres de pensamiento, porque precisamente es lo que más estamos sembrando; lo que más estamos sembrando son oportunidades a la inteligencia (...)”

Fidel Castro Ruz

DEDICATORIA

A las dos personas que me dieron la vida y quiero con todas las fuerzas de mi corazón, que me guiaron en todo momento, que me aman y protegen siempre: mis padres. A mi familia y todas aquellas personas que han compartido conmigo buenos y malos momentos

AGRADECIMIENTOS

A las personas que más adoro en el mundo, que siempre estuvieron a mi lado apoyándome y aconsejándome para mi bien, que sin su sacrificio y abnegación nunca hubiera salido adelante y por haber soportado mis malcriadeces y mal carácter: a mis padres.

A mis hermanos Leo y Migneiff por siempre estar a mi lado y demostrarme que tenemos más cosas en común que ser hermanos y por permitirme guiarlos y aconsejarlos.

A mi bisabuela Zoila.

A mis abuelos Oneida, Delia y Teobaldo.

A mi abuelo Miguel que a pesar de que la vida no me permitió darle esta alegría siempre estuvo en mis pensamientos.

A mis tíos Mariela, Guillermo, Osvaldo, Alejandro, Rosario, Luisito, Telma, Vivi, Idelio, Susana e Idelio por siempre estar a mi lado.

A mis primos por siempre estar pendientes de mí.

A Daniel por ser más que un padre para mí y demostrarme que no solo los de tu sangre son los que siempre estarán contigo.

A Rafael y Ariel por siempre aconsejarme en las buenas y en las malas.

A las amistades de mis padres por siempre preocuparse y estar ahí siempre que los necesité.

A Martica por siempre estar pendiente de mí y preocuparse a pesar de estar lejos.

A mi otra familia por no olvidarse de mí y estar pendiente: pipo, mima, Lisi, Felipe, Renier, Lismary, Fide, Bolo, Luis Daniel, Julia Mercedes, Vicente, Eliani, Elena, Nidia, Leraíne y Felicidad.

A Lusmey por darme ánimo y aconsejarme acerca el sacrificio que hacían mis padres porque yo me graduara.

A los padres de Yara por darme el mejor de sus consejos.

A mis padres blancos Gregorio y Liuba, a mi hermana Nanda, a mi profe Amador.

A mi compañero de tesis.

A mis hermanos blancos Alex, Yoandri, Sisley y Dainovy.

A mis compañeros de aula.

A mi profesor Pupo por ayudarme cuanto tenía dudas.

A Susej, Cecilia, Adys, Arianna, Mahelis, Lisbet, Yaineris, Shogun, Leidiana, Jorgito, Merlin, Maydelin y Yainelis por estar a mi lado en los buenos y malos momentos que pasé, aconsejarme y por siempre apoyarme.

A los integrantes del palenque: Jhony, Luis Miguel, Yanetzi y Leonel.

A Elizabeta, Ody, Yara, Melixa, Lisandra, Dixi, Yuliet, Diana, Nixys, Michelena, Mailen, Grechin, Alejandro, Eliadys, Leidys, las 3 Yanet, Sachie y Aliuska.

A mi piquete Yudi, Yaneisi, Javier, Adrián, Yenía, Pedro, LG, Lili "La Maquina", Yuliet, Surelis, Yailyn, La Gorda, Felix, Reynel, Mary y Aliuska.

A mis nuevo amigos Lafis, Cuca, Elizabeth, Dayana, Odalis, Mirita, Dayatni, Alejandro, Yanet, Andry, Darlys, Jose Carlos, Tony, Isenith, Erick, Dalilys, Carmen, Abel, Yudith, Pablo, Yuliet, Lianet y Leonel.

A mi tutor, a mi oponente y al tribunal de tesis por su apoyo, comprensión y entrega desde el primer día.

A todas mis amistades que de una forma u otra me ayudaron y compartieron conmigo buenos y malos momentos.

Quise dejar a esta persona para último pues el agradecimiento para ella es muy especial a pesar de haber compartido solo uno meses de mi vida. A Anabel por aparecer en mi vida en el momento justo y darme la confianza de haber seguido adelante, estar a mi lado, apoyarme en los momentos en que lo necesitaba.

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Centro GEYSED de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Alejandro Miguel Rodríguez Veranes (autor)

Jeans Michael Suárez Pérez (tutor)

DATOS DE CONTACTO

Co-tutor: Msc. Rafael Leodán Cardero Álvarez.

Formación Académica: Ingeniero en Ciencias Informáticas (Julio/2008).

Centro Laboral: Universidad de las Ciencias Informáticas (UCI).

Correo Electrónico: rlcordero@uci.cu

Otros Datos de Interés:

Profesor del Departamento de Señales Digitales, Facultad 6.

Tutor: Ing. Jeans Michael Suárez Pérez.

Formación Académica: Ingeniero en Ciencias Informáticas (Julio/2009).

Centro Laboral: Universidad de las Ciencias Informáticas (UCI).

Correo Electrónico: jmsuarez@uci.cu

Otros Datos de Interés:

Diploma de Oro.

Premio Mella.

Vinculado al Proyecto Productivo UCITeVe y CCM durante 3 años de estudios universitarios.

Líder de Proyecto CCM (Captura y Catalogación de Medias).

Profesor del Departamento de Señales Digitales, Facultad 6.

OPINION DEL TUTOR

RESUMEN

En la actualidad existe la necesidad de efectuar búsquedas en videos digitales por su contenido. En este aspecto se crean archivos confeccionados manualmente en los cuales se recopilan la duración, la velocidad, el tamaño, entre otras características descriptivas que no consiguen esquematizar el contenido de los mismos. Esto origina que, numerosas veces la calidad de las búsquedas no cumpla el objetivo planteado.

El presente trabajo propone desarrollar un componente que garantice la localización de subtítulos mediante el monitoreo constante de los mismos agilizando el proceso y así evitar que existan ausencias repetidas de palabras y caracteres. Para su desarrollo se propone la utilización de bibliotecas gratuitas como OpenCV, la cual cuenta con funciones factibles para el procesamiento de imágenes contenidas en fotogramas, compatible con el marco de trabajo seleccionado QtCreator apoyado por el lenguaje de programación C++ y la herramienta CASE Visual Paradigm para el modelado de clases.

Este componente cuenta con los elementos requeridos en materia de procesamiento de imágenes digitales. Hace uso además, de novedosas técnicas en el campo de reconocimiento de patrones para detectar y localizar los subtítulos contenidos en videos digitales.

PALABRAS CLAVES

Contenido, detección de bordes, fotogramas claves, imagen digital, subtítulo

ABSTRACT

At present there is a need to search in digital video for their content. In this aspect made manually files are created which are collected in the length, speed, size, duration, and other descriptive characteristics that fail to outline the contents thereof. This causes, many times the search quality does not meet the objective.

The present work aims to develop a component that ensures the location of subtitles through constant monitoring of these streamlining the process and so avoid having repeated absences of words and characters. For its development we propose the use of free libraries like OpenCV, which has functions feasible for processing images in frames, compatible with the framework chosen QtCreator supported by the programming language C++ and CASE tool Visual Paradigm for modeling classes.

This component system has the required elements in digital image processing. Also makes use of new techniques in the field of pattern recognition to detect and locate the subtitles in digital video content.

KEY WORDS

Content, digital image, edge detection, key frames, subtitle

INDICE DE TABLAS

Tabla 1 Resultados de las pruebas realizadas al algoritmo29

INDICE DE IMÁGENES

Ilustración 1 Imagen con borde	7
Ilustración 2 Detección de bordes.....	8
Ilustración 3 Imágenes procesadas por el detector de Canny y el operador Sobel.....	9
Ilustración 4 Imagen con región de interés definida.....	19
Ilustración 5 Imagen transformada a escala de grises	20
Ilustración 6 Imagen con detección de bordes.....	20
Ilustración 7 Imagen binarizada.....	21
Ilustración 8 Imagen dilatada	22
Ilustración 9 Imagen con comparación de áreas de blobs.....	23
Ilustración 10 Diagrama de Clases del Diseño	26
Ilustración 11 Diagrama de Componentes	27
Ilustración 12 Modelo para la detección de subtítulos	34
Ilustración 13 Esquema del algoritmo de detección basado en bordes.....	35
Ilustración 14 Diferentes resultados de la ejecución del algoritmo	36

INDICE GENERAL

INTRODUCCION.....	1
CAPÍTULO 1: Fundamentación teórica.....	4
1.1 Introducción	4
1.2 Conceptos asociados al dominio del problema.....	4
1.3 Etapas de la extracción de textos en imágenes contenidas en videos.....	5
1.4 Objeto de Estudio	6
1.4.1 Descripción General.....	6
1.4.2 Métodos basados en la detección de bordes.....	7
1.4.2.1 Detector de Canny.....	8
1.4.2.2 Operador Sobel.....	10
1.4.3 Tipos de subtítulos.....	10
1.4.3.1 Según su aspecto formal.....	10
1.4.3.2 Según su aspecto lingüístico	11
1.4.3.3 Según su aspecto técnico	11
1.4.4 Situación del dominio actual del problema	12
1.4.5 Situación Problemática	12
1.5 Definición de la tecnología de desarrollo.....	12
1.5.1 Biblioteca.....	13
1.5.1.1 OpenCV.....	13
1.5.2 Lenguaje de Programación	14

1.5.2.1	C++.....	14
1.5.3	Entorno de Desarrollo Integrado	15
1.5.3.1	QtCreator.....	15
1.5.4	Herramienta CASE	15
1.5.4.1	Visual Paradigm.....	16
1.6	Conclusiones parciales	16
CAPÍTULO 2: Diseño e implementación del modelo para la detección de subtítulos e identificación del cambio de los mismos en fotogramas de video.....		17
2.1	Introducción	17
2.2	Definición de los Requisitos funcionales para la integración en la Plataforma de Codificación e Indexación.....	17
2.2.1	Mostrar nombre del proceso en ejecución.....	17
2.2.2	Visualizar porcentaje de realización del proceso.....	17
2.2.3	Cancelar proceso.....	17
2.2.4	Enviar errores encontrados durante el proceso.....	18
2.3	Descripción general de la propuesta de modelo.....	18
2.3.1	Captura de los fotogramas del video	19
2.3.2	Definición de las regiones de interés.....	19
2.3.3	Transformación a escala de grises	19
2.3.4	Detección de bordes.....	20
2.3.5	Binarización	21
2.3.6	Dilatación.....	21

2.3.7	Identificación de componentes conexas para la detección de regiones de subtítulos	22
2.3.8	Comparación entre fotogramas continuos para la detección de cambio de subtítulos y selección de fotogramas claves	23
2.4	Conclusiones parciales	23
CAPÍTULO 3: Validación del componente propuesto.....		25
3.1	Introducción	25
3.2	Modelo de diseño.....	25
3.2.1	Clases del diseño.....	25
3.2.2	Diagrama de clases del diseño.....	25
3.3	Modelo de Implementación	27
3.3.3	Diagrama de componentes.	27
3.4	Validación en base de datos heterogénea de video con subtítulos.....	28
3.5	Conclusiones parciales	29
CONCLUSIONES GENERALES.....		30
RECOMENDACIONES.....		31
BIBLIOGRAFÍA		32
ANEXOS		34
GLOSARIO		37

INTRODUCCION

Con el surgimiento de las tecnologías de la informática y las comunicaciones (TICs) el hombre ha controlado la información, ha transformado las señales sonoras y visuales en información digital y esto le permitió analizar un rango más amplio de dichas señales así como su almacenamiento y transmisión global. Muchas de las decisiones diarias de los individuos en los tiempos actuales involucran el uso de tecnologías de software y hardware computacionales, las cuales ofrecen vías alternativas para acceder y desplegar información.

En los últimos años los archivos multimedia en específico los videos digitales han hecho fija su presencia en el marco de la transmisión y divulgación de la información a través de los diferentes medios de difusión. Dentro del marco tecnológico, el tratamiento de los videos digitales es un proceso de gran importancia debido a la gran variedad de tareas que se puede realizar con dichos archivos multimedia en las que se incluyen la edición, conversión, extracción de sonidos, entre otros trabajos. Todas estas tareas poseen gran valor, pero, una de las más importantes incluidas en el procesado de videos es el trabajo con los subtítulos hallados en dichos archivos multimedia. La función fundamental del subtítulo no solo se corresponde a la transcripción o traducción del audio al texto en el idioma deseado o para expresar la idea del material, sino que también es usado como referencia para las personas con debilidad auditiva al mostrar en texto plano la trama del mismo. El subtítulo, implícito en los fotogramas¹, es un componente importante en cuanto a la información acerca del contenido del video.

Actualmente es posible realizar búsquedas por contenidos personalizados en videos digitales, estos permiten recuperar en detalles gran parte de la información que contienen, además de ampliar la posibilidad de apuntar a citas concretas, frases platicadas y afirmaciones que pueden tener múltiples aplicaciones en centenares de campos. Por otra parte estas acciones se pueden realizar en dos tipos de textos embebidos: los captions² los cuales forman textos superpuestos artificialmente sobre la secuencia, y los textos vinculados naturalmente a la escena como señales o anuncios. Estos últimos ofrecen la posibilidad de brindar información semántica de la secuencia, pero la información brindada por los captions está más centrada al contenido por lo que ambos son útiles a la hora de indexar el video.

1 Conjunto de imágenes que se suceden en una película y que representa el contenido de la película en un instante de tiempo.

2 Se define como la palabra impresa debajo de una imagen o dibujo animado que explica de qué se trata.

En la Universidad de la Ciencias Informáticas (UCI) en específico en la facultad 6, se localiza el centro Geoinformática y Señales Digitales (GEYSED) al cual pertenece el departamento Señales Digitales, este último tiene como principal línea de trabajo el desarrollo de soluciones, productos y servicios para la automatización, transmisión y gestión en el área de procesamiento digital de imágenes y señales. Actualmente el proyecto Captura y Catalogación de Medias (CCM) posee la necesidad de ampliar el campo de trabajo con metadatos que brinden una información consistente acerca el contenido de archivos multimedia como los videos digitales. En la actualidad el proceso de transcripción del audio al texto se define como un trabajo monótono, pues el mismo se realiza de forma manual trayendo consigo un conjunto de errores como el surgimiento de errores humanos y procesos equívocos. Por otra parte fue necesario realizar un visionado constante de los fotogramas de un video en ejecución para captar la información plasmada en los subtítulos como proceso análogo a la transcripción de audio, arrojando resultados similares al proceso anteriormente mencionado pues además de ser ejecutado manualmente adicionalmente existían problemas si el video en análisis presentaba una baja resolución o simplemente, un background³ complejo.

Atendiendo a la problemática antes valorada el presente trabajo está enfocado a resolver el siguiente **problema a resolver**: ¿Cómo garantizar la localización ágil de subtítulos mediante el monitoreo constante de los mismos, sin la presencia de repeticiones del propio subtítulo y ausencia de palabras o caracteres?, teniéndose como **objeto de estudio** la detección de textos contenidos en imágenes, enmarcándose en el **campo de acción** del proceso de localización de subtítulos en imágenes contenidas en videos. Planteándose entonces como **idea a defender** que con la elaboración de un componente que garantice la localización de subtítulos mediante el monitoreo constante de los mismos se podrá agilizar el proceso evitando que existan ausencias repetidas de palabras y caracteres.

El **objetivo general** de la investigación es desarrollar un componente que localice las regiones de los subtítulos en imágenes contenidas en videos.

Para darle cumplimiento al objetivo general se trazaron las siguientes **tareas de investigación**:

- Caracterización de los procesos de indexación automática de metadatos utilizando los subtítulos de video.
- Selección de las técnicas para localización de texto en imágenes.

³ Se define como toda tarea o trabajo que se realiza en segundo plano, es decir, algo que se está llevando a cabo con una prioridad baja.

- Identificación regiones con subtítulos en imágenes de video.
- Elaboración del modelo para la detección de subtítulos e identificación del cambio del mismo en fotogramas de video.
- Definición de los requisitos para la integración en la Plataforma de Codificación e Indexación.
- Implementación de la estructura de negocio asociada a las pautas identificadas en la Plataforma.
- Validación del componente en base de datos heterogénea de video con subtítulos.

Con el correcto cumplimiento de las tareas se espera obtener el siguiente resultado:

- Un componente desarrollado en C++ que a partir de un video extrae los fotogramas significativos (de acuerdo a los cambios de subtítulos) e indica la ubicación del subtítulo en cada fotograma.

Para adquirir los conocimientos necesarios que permitan el cumplimiento del objetivo trazado y desempeñar las tareas planeadas, se lleva a cabo una investigación en la que se utilizan los siguientes métodos científicos:

Métodos teóricos:

- Analítico – Sintético: Permitió concretar los elementos más importantes relacionados con las técnicas de extracción de textos en imágenes contenidas en videos utilizadas en el componente que necesita el proyecto de CCM para su sistema.
- Histórico – Lógico: Ayuda a la comprensión de la evolución en el mundo de los sistemas para la búsqueda y detección de textos en videos a lo largo de la historia.

Métodos empíricos:

- Observación: Este método es de suma importancia puesto que permite observar mediante el registro visual lo que ocurre en la situación real que se analiza.

CAPÍTULO 1: Fundamentación teórica.

1.1 Introducción

En este capítulo se muestran a un conjunto de conceptos asociados al tema del procesamiento de imágenes los cuales son necesarios explicar para lograr un mayor entendimiento del problema así como una descripción de las etapas del proceso de extracción de textos en imágenes contenidas en videos. Además se realiza un estudio del estado del arte de los disimiles métodos de detección de texto en imágenes contenidas en videos conjunto a la definición de la tecnología de desarrollo.

1.2 Conceptos asociados al dominio del problema

Los **subtítulos** se definen como el texto escrito que aparece en algunas películas, generalmente en la parte inferior de la pantalla, sobre las imágenes y que traduce el texto original de los diálogos en películas no dobladas (1). Los subtítulos son relevantes debido a la información brindada acerca la trama de la película por lo que se consideran ampliamente importantes como metadato. Los **metadatos** se concretan como los datos estructurados que describen y permiten encontrar, comprender o preservar documentos archivísticos al largo del tiempo (2). Además es toda aquella información descriptiva sobre el contexto, calidad, condición o características de un recurso, dato u objeto que tiene la finalidad de facilitar su recuperación, autenticación, evaluación, preservación o interoperabilidad (3). La existencia del subtítulo como metadato es de sumo valor debido a que constituye un descriptor de los videos. El **video** hace referencia a la captación, procesamiento, transmisión y reconstrucción de una secuencia de imágenes y sonidos que representan escenas en movimiento (4). En el video, el subtítulo como texto puede ser buscado como contenido debido a que el mismo en la mayoría de los casos describe de qué se trata dicho archivo multimedia. El **texto** se plantea como lo que se pone en el cuerpo de una obra manuscrita o impresa, a diferencia de lo que en ella va por separado; como las portadas, las notas, los índices, etc. (1). A su vez el **contenido** trata acerca lo que algo que se contiene dentro de alguna cosa; también se define como tabla de materias, a modo de índice (1). Por lo que el subtítulo en el video se definiría como un texto el cual en forma de contenido brinda información acerca la trama del video.

1.3 Etapas de la extracción de textos en imágenes contenidas en videos

Para dar cumplimiento al objetivo propuesto se realiza una investigación acerca el proceso de extracción de textos en imágenes contenidas en videos. El mismo está dividido en 4 etapas y las mismas se dividen en subetapas:

1. Obtención de Imagen

En esta etapa hay dos alternativas en cuanto a cómo obtener los datos de entrada al sistema. Cada alternativa manejará tipos de datos distintos y por ende modificará el tipo de procesamiento de las etapas siguientes. A continuación se explicará las dos alternativas o modos para la obtención de imágenes.

✓ Offline

En este modo se trabaja con imágenes que contienen el texto a reconocer. En esta modalidad se busca analizar la morfología estática de la figura.

✓ Online

En este modo se trabaja con información obtenida del dispositivo de entrada del usuario, por lo que generalmente se trata con información vectorial que describe trazos. Está asociado directamente al reconocimiento de texto manuscrito en tiempo real (5).

2. Segmentación

La segmentación es la etapa en la que se busca extraer cada carácter de una línea de texto. La complejidad del procesamiento de esta etapa es mucho mayor para el caso de texto manuscrito a la de texto impreso. Esto se debe a que en el texto manuscrito pueden existir, según el estilo de escritura, diversos casos de separación entre caracteres. Las condiciones más favorables implicarían que los caracteres estuvieran perfectamente separados para su extracción. Para otros casos, donde los caracteres se encuentren unidos, sería necesario utilizar algún otro parámetro como puede ser la longitud de los caracteres o la morfología de los mismos (5).

3. Extracción de características

La extracción de características es una fase de suma importancia en el proceso de extracción de texto en imágenes. En esta etapa se busca obtener los valores de ciertas características predefinidas que

permiten identificar patrones. Existen diversas posibilidades a la hora de especificar qué características se van a utilizar, y la toma de esta decisión definirá en gran medida el rendimiento del sistema.

Al elegir las se debe tener presente que las características de distintas instancias de una clase deben seguir un patrón, mientras que las características de clases distintas deben ser lo suficientemente distintas como para poder diferenciarlas. Si es diseñada correctamente, la tarea del clasificador se verá beneficiada por esta etapa, ya que se le pasará un conjunto de datos de dimensionalidad reducida, con la información redundante de la imagen eliminada.

Es también una posibilidad obviar por completo esta etapa y alimentar directamente la fase de clasificación luego de binarizar⁴ la imagen preprocesada. No obstante, es probable que el rendimiento del clasificador sea menor al operar de esta forma (5).

4. Clasificación

El objetivo de esta última etapa es el de asociar el conjunto de datos obtenido en la etapa de extracción de características de un carácter a una clase en particular (5).

1.4 Objeto de Estudio

Para el desarrollo de la solución informática es necesario estudiar los diferentes métodos que permiten la detección de subtítulos contenidos en imágenes.

1.4.1 Descripción General

Es posible encontrar frecuentemente textos insertados en las imágenes tanto nombres de personas, como títulos o fechas de eventos. Esta circunstancia plantea una necesidad básica: la detección de textos contenidos en las imágenes deseadas.

El método habitual de detección, que es la detección manual del texto contenido en la imagen, se muestra limitado e inadecuado por disímiles motivos. Fundamentalmente es posible subrayar dos dificultades:

- La cantidad de trabajo que se solicita para adquirir informaciones contenidas en las imágenes conjunto a la necesidad de expresar el rico contenido presente en ellas es enorme.

⁴ Consiste en convertir la imagen digital en una imagen en blanco y negro, de tal manera que se preserven las propiedades esenciales de la imagen.

- Resulta tedioso y difícil recobrar las informaciones contenidas en las imágenes.

Para darle solución a estos problemas se estudiaron una serie de métodos o procedimientos que se pueden emplear para la detección de textos contenidos en imágenes.

1.4.2 Métodos basados en la detección de bordes

Un borde se define como la frontera entre dos regiones con niveles de grises relativamente diferentes. La detección de bordes reduce significativamente la cantidad de información irrelevante en una imagen, conservando, al mismo tiempo, las características fundamentales de su estructura. Existen muchas técnicas diferentes para la detección de bordes pero, en la mayoría de los casos, la idea básica consiste en aplicar localmente algún operador derivativo como el gradiente.

Tipos de bordes:

- Cambio brusco en la distancia cámara-objeto (dc)
- Cambio en la normal del objeto (n)
- Cambio en la reflectancia del objeto (r)
- Cambio en la proyección de la luz incidente (s)

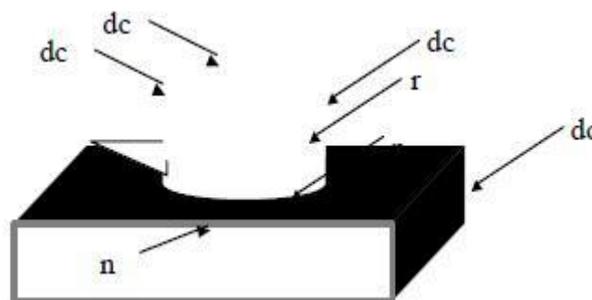


Ilustración 1 Imagen con borde

- Filtrado + Extracción = Detección de bordes
- Umbralización:
 - Selecciona píxeles etiquetados como bordes.

➤ Localización:

- Suministra información exacta de la posición y orientación del borde.

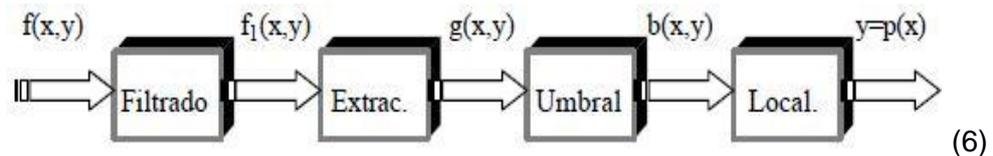


Ilustración 2 Detección de bordes

El proceso de detección de bordes utilizan disímiles operadores. Entre ellos es posible destacar dos ramas (6):

Técnicas basadas en cruces por cero:

- Detector de Canny (7)

Técnicas basadas en el gradiente:

- Operador de Sobel (8)

1.4.2.1 Detector de Canny

Entre los filtros más utilizados actualmente se encuentra el operador de Canny, el cual es un operador diferencial combinado a un filtro gaussiano. En 1986, Canny propuso un método para la detección de bordes, el cual se basaba en tres criterios, estos son:

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

Uno de los métodos relacionados con la detección de bordes es el uso de la primera derivada, la cual 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 (9), característica que es usada para detectar un borde, y en la que se basa el algoritmo de Canny.

El operador Canny trabaja en diferentes etapas. En primer lugar, se suaviza la imagen fusionándola con una función Gaussiana de dos dimensiones. Esto sirve para filtrar el ruido presente en la imagen a costa de perder un poco de detalle. El grado de suavizado se puede controlar mediante la anchura de la Gaussiana. Como ya se ha comentado anteriormente, esta función se aproxima en la práctica mediante una máscara que recorrerá toda la imagen píxel a píxel. El tamaño de dicha máscara determinará el grado de suavizado.

El algoritmo básico del detector de bordes de Canny actúa de la siguiente forma:

- ✓ Aplicar la máscara de suavizado sobre la imagen.
- ✓ Aplicar algún operador derivativo para encontrar bordes y calcular su magnitud y dirección.
- ✓ Realizar la supresión de no-máximos sobre los bordes encontrados.

Como se puede apreciar al procesar la imagen (Figura 3 a), mediante Canny (Figura 3 b) se han detectado todos los bordes y se ha suprimido casi todo el ruido. El resultado usando el operador Sobel es mucho menos satisfactorio (Figura 3 c).

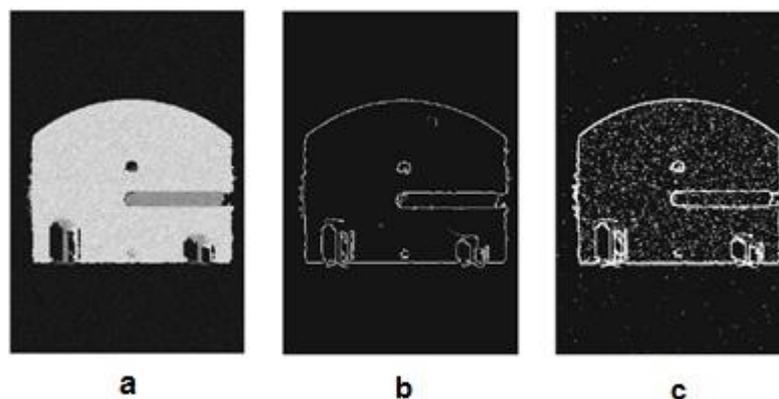


Ilustración 3 Imágenes procesadas por el detector de Canny y el operador Sobel

1.4.2.2 Operador Sobel

El operador de Sobel calcula el gradiente de la intensidad de brillo de cada píxel, brindando así la dirección del mayor incremento posible de negro a blanco. El resultado muestra qué tan abrupto o suave puede ser el cambio producido en la imagen en cada punto analizado, y al mismo tiempo que tanto un punto determinado representa un borde en la misma, además de brindar la orientación a la que tiende ese borde. Este operador utiliza dos máscaras de 3x3 elementos fusionándola con la imagen original con el fin de calcular las aproximaciones a las derivadas. Uno de estos es empleado en los cambios horizontales mientras que el segundo para los verticales.

Si se precisa A como la imagen original y G_x y G_y se definen como las máscaras que representan las derivadas de la intensidad el resultado se calcula como:

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * A \quad \text{y} \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

En cada punto de la imagen, se pueden combinar los resultados de las aproximaciones de los gradientes horizontal y vertical para obtener la magnitud del gradiente, a través:

$$G = \sqrt{G_x^2 + G_y^2}$$

Con la información obtenida se puede calcular también la dirección del gradiente:

$$\theta = \arctan\left(\frac{G_y}{G_x}\right)$$

En la cual θ es 0 para bordes verticales con puntos más oscuros al lado izquierdo.

1.4.3 Tipos de subtítulos

Dependiendo de los diferentes aspectos, existen varios tipos de subtítulo:

1.4.3.1 Según su aspecto formal

- ✓ Subtítulos tradicionales

Subtítulos reducidos: son empleados en determinados programas de televisión como son los informativos y los documentales. Ofrecen un primer resumen de lo que se dice en la lengua original.

Subtítulos bilingües: son usados en aquellas zonas donde conviven varias lenguas. A cada una de las dos líneas del subtítulo se le asigna una lengua diferente.

- ✓ Subtítulos simultáneos o a tiempo real

Los subtítulos en tiempo real se crean según van sucediendo los eventos. Un transcriptor (a menudo un reportero del tribunal o taquígrafo experto) usa un estenotipo con un teclado fonético y un programa especial. Una computadora traduce los símbolos fonéticos a subtítulos en inglés casi instantáneamente. El leve retraso está basado en la necesidad del transcriptor de oír la palabra y en el tiempo de procesamiento de la computadora. Los subtítulos en tiempo real se pueden usar en los programas que no tienen ningún guión; sucesos en vivo, incluidos las reuniones del congreso; los programas de noticias; y las reuniones que no son transmitidas, como por ejemplo las reuniones nacionales de asociaciones profesionales.

1.4.3.2 Según su aspecto lingüístico

- ✓ Subtítulos intralingüísticos

También denominados subtítulos de teletexto, están escritos en la misma lengua que la de la banda sonora sus receptores se encuentran entre personas con problemas auditivos y entre personas que están aprendiendo y quieren practicar lenguas.

- ✓ Subtítulos interlingüísticos

Son aquellos que traspasan a otra lengua la banda sonora original.

1.4.3.3 Según su aspecto técnico

- ✓ Subtítulos abiertos

Aparecen en todos los televisores y pueden verse sin un descodificador. Antiguamente, algunos noticiarios, discursos presidenciales o la programación creada por o para las personas sordas y los televidentes con problemas de audición ofrecían subtítulos abiertos. Hoy en día rara vez se utilizan.

- ✓ Subtítulos cerrados

Los televidentes necesitan un descodificador que se coloca en la parte de arriba del televisor o un televisor con sistema de circuitos de descodificación incorporado (10).

1.4.4 Situación del dominio actual del problema

En la actualidad está vigente la necesidad de efectuar búsquedas en videos por su contenido. Usualmente se crean catálogos elaborados de forma manual donde se coleccionan las características de dichos videos como la duración, el título, el género, entre otras, sin embargo con estos datos no se logra sintetizar el contenido a dichos archivos multimedia. Luego de ser procesada manualmente dicha información se archiva para posteriores búsquedas pero este tipo de procesamiento causaría diversos problemas. El trabajo con metadatos como el subtítulo se considera de gran importancia en el procesamiento de videos debido a que constituye un descriptor importante de la trama de dichos archivos multimedia. El subtítulo posee determinadas características en los fotogramas de los videos como la posición o ubicación facilitando mediante las diferentes técnicas incluidas en el procesamiento de imágenes su detección. En el componente propuesto permite comprobar generalmente la existencia de textos y posteriormente especificar cuáles se pueden clasificar como subtítulos, así como la identificación del cambio de estos últimos en fotogramas de video.

1.4.5 Situación Problemática

Actualmente el proyecto Captura y Catalogación de Medias (CCM) posee la necesidad de ampliar el campo de trabajo con metadatos que brinden una información consistente acerca el contenido de archivos multimedia como los videos digitales. En la actualidad el proceso que se ocupa de la traslación del audio al texto se puntualiza como un trabajo tedioso, debido a que este se efectúa manualmente por lo ocasiona un conjunto de errores como el surtimiento de errores humanos y procesos erróneos. Además fue preciso ejecutar un visionado constante de los fotogramas de un video en ejecución para obtener la información que se encontraba en los subtítulos como proceso semejante a la transcripción de audio, arrojando resultados relacionados al proceso anteriormente mencionado pues adicionalmente existían problemas si el video en análisis exhibía una baja resolución o sencillamente un background complejo.

1.5 Definición de la tecnología de desarrollo.

Un elemento fundamental para un correcto desarrollo, así como para un ahorro considerable en el tiempo de desarrollo, es sin dudas, la buena selección de las tecnologías.

1.5.1 Biblioteca

Es un conjunto de subprogramas utilizados para desarrollar software. Las bibliotecas contienen código y datos, que proporcionan servicios a programas independientes, es decir, pasan a formar parte de estos. Esto permite que el código y los datos se compartan y puedan modificarse de forma modular (11).

1.5.1.1 OpenCV

Se tiene la necesidad de cargar y leer imágenes, detectar bordes a las mismas así como su conversión a colores. Existen bibliotecas que tienen muchas de estas operaciones implementadas. Una de las bibliotecas que contiene en su estructura más de 500 funciones para cálculos algebraicos y combinaciones de estos aplicados a la visión artificial es OpenCV. Se seleccionó esta biblioteca debido a que implementa una gran diversidad de herramientas para la interpretación de la imagen. Es compatible con Intel IPL⁵ que implementa algunas operaciones en imágenes digitales. Esta librería es nativa en C++ por lo que seguir con este lenguaje aumentaría la eficiencia en las operaciones.

A pesar de primitivas como binarización, filtrado, estadísticas de la imagen, OpenCV es primordialmente una librería que te brinda gran variedad de funcionalidades. Se necesita hacer cálculos matemáticos con matrices, realizar operaciones con las imágenes por lo que OpenCV es útil para implementar dichas funciones por lo que nos brinda varias funcionalidades:

- Lectura de imágenes de diferentes formatos, aquí cabe destacar que permite recibir las imágenes en formatos BMP, DIB, JPEG, JPG, JPE, PNG, PBM, PGM, PPM, SR, RAS, TIFF y TIF. Esto le permitirá al componente realizar diferentes tipos de pruebas para la lectura de imágenes para su posterior modificación y guardado.
- Permite trabajar con varias funciones para el tratamiento de matrices, la mayoría de ellas se basan en la realización de operaciones elemento por elemento. Permite operaciones tales como multiplicación e inversión de matrices.
- Asiente binarizar imágenes para la detección de posibles cajas de texto o sea debe ser capaz de permitir tomar una imagen en escala de grises (imagen de intensidad) y convertirla en imagen binaria, mediante la aplicación de un umbral. Todos los valores superiores al valor

⁵ Image Processing Library, en español Librería de procesamiento de imágenes.

umbral serán binarizados a 1 y los que no superen es valor lo serán a 0.

- Permite realizar el escalado el cual es una transformación afín que puede aumentar o disminuir un objeto por un determinado factor. Esta operación es importante para realizar el primer paso para el proceso de detección de textos basados en bordes.

El rasgo fundamental de la biblioteca es su desempeño. Los algoritmos están asentados en estructuras de datos muy flexibles, conectados con estructuras IPL; más de la mitad de las funciones ha sido perfeccionada beneficiándose de la Arquitectura de Intel (12).

Está compuesta por cuatro módulos fundamentales, donde cada uno tiene asignada una función específica:

- cv tiene las funciones principales de OpenCV.
- cvaux contiene las funciones auxiliares de OpenCV.
- cxcore se encarga de la estructura de datos y el soporte de álgebra lineal.
- highgui posee las funciones gráficas y de interfaz de usuario.

1.5.2 Lenguaje de Programación

Es un conjunto de sintaxis y reglas semánticas que definen los programas del computador. Es una técnica estándar de comunicación para entregarle instrucciones al computador. Un lenguaje le da la capacidad al programador de especificarle al computador, qué tipo de datos actúan y que acciones tomar bajo una variada gama de circunstancias, utilizando un lenguaje relativamente próximo al lenguaje humano (13).

1.5.2.1 C++

C++ es una versión ampliada del lenguaje C. En él se pueden encontrar las mismas funcionalidades que en el lenguaje C, incluyendo además algunas mejoras considerables y soporte para la programación orientada a objetos.

Se puede decir que debido a las muchas mejoras y características nuevas que posee, se convierte sencillamente en un C mejorado, independientemente de la programación orientada a objetos. Una particularidad del C++ es la posibilidad de redefinir los operadores, es decir, permite sobrecarga de

operadores, y de poder crear nuevos tipos que se comporten como tipos fundamentales. Permite trabajar tanto a alto como a bajo nivel.

Es importante señalar que aunque todo lo que se conozca sobre el lenguaje C se puede aplicar a C++, comprender sus características más avanzadas requerirá una importante inversión de tiempo y esfuerzo. De cualquier manera las recompensas de programar en este lenguaje justificarán de sobra el esfuerzo realizado (14).

1.5.3 Entorno de Desarrollo Integrado

Un Entorno de Desarrollo Integrado (IDE) es un programa que está compuesto por un conjunto de herramientas para un programador. Facilita un marco de trabajo amigable para una gran cantidad de lenguajes de programación tales como C++, Java, C#, logrando utilizarse en el mismo uno o varios lenguajes de programación.

1.5.3.1 QtCreator

QtCreator es un entorno multi-plataforma de desarrollo (IDE), de código abierto. Desarrollado por la empresa Nokia en el Framework de desarrollo QT, adaptado a las necesidades de los desarrolladores de Qt. Este brinda las prestaciones tales como editor de código C++, diseño de interfaz de usuario integrado, proyecto y construcción de herramientas de gestión, soporte para el control de versiones, creación de librerías, implementación de señales y slots y soporte a proyectos de escritorio (15). Por estas razones se recomienda el uso de QtCreator como IDE para el desarrollo del componente debido a que cumple con todo lo necesario para un desarrollo ágil y óptimo (15).

1.5.4 Herramienta CASE

Las herramientas CASE (Computer Aided/Assisted Software/System Engineering) o (Ingeniería de Software Asistida por Computadora) brindan soporte a un enfoque de ingeniería en el desarrollo de software en alguna o en todas las fases de este proceso (16).

Tienen como objetivo:

- Permitir la aplicación práctica de metodologías estructuradas.
- Facilitar la realización de prototipos y el desarrollo conjunto de aplicaciones.
- Simplificar el mantenimiento de los programas.

- Mejorar y estandarizar la documentación.
- Aumentar la portabilidad de las aplicaciones.
- Facilitar la reutilización de componentes del software.
- Permitir un desarrollo y un refinamiento visual de las aplicaciones.

1.5.4.1 Visual Paradigm

Visual Paradigm para UML es una herramienta CASE, que utiliza UML 2.1 como lenguaje de modelaje y soporta el ciclo de vida completo del desarrollo de software. Es multiplataforma y posibilita la generación de código y documentación. Permite dibujar 13 tipos de diagramas diferentes a través de un intuitivo modelado visual y la aplicación de ingeniería inversa. Admite la importación y exportación de XML e imágenes, la administración de requerimientos, la creación de esquemas de clases a partir de una base de datos y viceversa, además cuenta con un soporte que facilita el trabajo simultáneo sobre un mismo diagrama entre dos desarrolladores en un tiempo real (17).

1.6 Conclusiones parciales

Después de realizar un riguroso estudio de las técnicas de detección de bordes se arriba a la conclusión de que el operador Sobel es una buena opción para desarrollar la solución que se presenta. Es necesario detectar la mayor cantidad de bordes horizontales, y sin dudas dicho operador es el método más factible pues el mismo posee como ventaja principal la detección de bordes en todas las direcciones ofreciendo un mayor peso en los bordes horizontales con respecto a las diagonales. Además de reduce de manera significativa la cantidad de información contenida en las imágenes filtrando todos aquellos datos irrelevantes para la detección del texto, preservando así las características primordiales de la imagen. Además con el uso correcto de las técnicas anteriormente mencionadas auxiliadas de la biblioteca OpenCV, el lenguaje de programación C++, la herramienta CASE Visual Paradigm y el IDE QtCreator se espera obtener como resultado un componente confiable y robusto.

CAPÍTULO 2: Diseño e implementación del modelo para la detección de subtítulos e identificación del cambio de los mismos en fotogramas de video.

CAPÍTULO 2: Diseño e implementación del modelo para la detección de subtítulos e identificación del cambio de los mismos en fotogramas de video.

2.1 Introducción

En el presente capítulo se realiza el diseño e implementación de un modelo para dar solución al problema que dio origen a la presente investigación. Se plantean con argumentos sólidos los diferentes pasos por los que está constituido el diseño para la detección de subtítulos e identificación de cambio de los mismos en fotogramas de videos. Además se definen los requisitos funcionales para realizar la integración en la Plataforma de Codificación e Indexación conjunto a la implementación de la estructura de negocio asociada a las pautas identificadas en la Plataforma.

2.2 Definición de los Requisitos funcionales para la integración en la Plataforma de Codificación e Indexación

“La IEEE Standard Glosar of Software Engineering Terminology define un requerimiento como condición que tiene que ser alcanzada por un sistema o componente de un sistema para satisfacer un contrato, estándar u otro documento impuesto formalmente.” Los requerimientos se clasifican en funcionales y no funcionales (16). A continuación se muestran los requisitos definidos para la integración con la plataforma.

2.2.1 Mostrar nombre del proceso en ejecución

Este requisito permite saber el nombre que identifica al proceso de detección de subtítulos presentes en fotogramas de videos.

2.2.2 Visualizar porcentaje de realización del proceso

El objetivo de este requisito se basa en permitirle al usuario observar por cual porciento de realización se encuentra el proceso en ejecución.

2.2.3 Cancelar proceso

La aplicación brinda la opción de cancelar el proceso cuando el usuario lo desee, facilitando así que se eliminen de forma automática todos los registros creados en el servidor por dicho proceso y que no se pueda evidenciar la existencia del mismo.

CAPÍTULO 2: Diseño e implementación del modelo para la detección de subtítulos e identificación del cambio de los mismos en fotogramas de video.

2.2.4 Enviar errores encontrados durante el proceso

Este requisito se realiza con el objetivo de que el usuario visualice los diferentes errores que pueden ocurrir antes, durante y después de la realización del proceso de detección de subtítulos en los fotogramas del video. Se pueden encontrar errores tales como:

- El fichero posee un formato incorrecto.
- El fichero se encuentra vacío.

2.3 Descripción general de la propuesta de modelo

El diseño del modelo (Ver Anexo 1) se hace siguiendo el orden en que se identifican los requisitos funcionales. A partir de un video original el cual se captura mediante funciones de lectura de la biblioteca OpenCV, luego se procede a la captura de los fotogramas del mismo a los cuales se les establece una región de interés para evitar procesar toda el área de los fotogramas. Esto se definió estableciendo las regiones donde se intensifica la aparición de los subtítulos. Luego a dichas regiones de interés de cada fotograma se les realiza una transformación a escala de grises. Para el proceso de detección de bordes se utilizará el método de detección de bordes Sobel seguido de la binarización de los fotogramas, luego se procede a la dilatación quedando representadas en blanco las posibles regiones de texto. Seguidamente se realiza la identificación de las componentes conexas para posteriormente identificar cuáles regiones se pueden definir como subtítulos. Seguido de esto se realiza una comparación de las regiones de interés de fotogramas continuos para detectar cambios de subtítulos en los mismos. En este sentido, antes de iniciar el recorrido completo del video es necesario extraer el primer fotograma para luego en la primera corrida del ciclo extraer el siguiente y a partir de este momento en cada recorrido usar el fotograma actual y el consecutivo.

El concepto de consecutivo manejado en la implementación está controlado por una variable que es un salto entre fotogramas de modo que se pueda optimizar el procesamiento pues no es necesario utilizar todo el video sino que procesando fotogramas separados por un paso se obtienen resultados similares, durante la detección de subtítulos. Por último se identifica la imagen con subtítulo y la aparición de esta en el video. Estas operaciones se repiten hasta que no haya fotogramas en el video.

CAPÍTULO 2: Diseño e implementación del modelo para la detección de subtítulos e identificación del cambio de los mismos en fotogramas de video.

2.3.1 Captura de los fotogramas del video

Un video está formado por un número consecutivo de imágenes. En este paso se obtendrían los fotogramas, es decir, las imágenes contenidas en el video para poder procesarlas y extraer los datos necesarios para el procesamiento. Para ello se utiliza el método de lectura de video que posee OpenCV por lo que es necesario crear una estructura de tipo **cvCapture** donde se captura el video que se pasa como parámetro al método principal y con la función **cvQueryFrame** se accede de manera consecutiva a los fotogramas que componen el video.

2.3.2 Definición de las regiones de interés

Seguido de la captura de los fotogramas se procede a realizar el procesamiento de los fotogramas para hallar las regiones de subtítulos. Para ello se definen las regiones en las cuales se intensifica la aparición de los subtítulos o sea las regiones de interés. Inicialmente se crea una estructura de tipo **cvSize** utilizado todo a la hora de crear imágenes nuevas, para definir sus dimensiones. Seguido de esto se crea una estructura de tipo **cvRect** estableciéndole a este último las dimensiones (x, y, alto y ancho) de la estructura **cvSize** utilizada para definir las dimensiones de un rectángulo en píxeles. Mediante la función de OpenCV **cvSetImageROI** teniendo como parámetros el fotograma original y el rectángulo se logra establecer en dicho fotograma la región de interés a procesar o sea de dicho fotograma se analiza la parte con las dimensiones de dicho rectángulo.

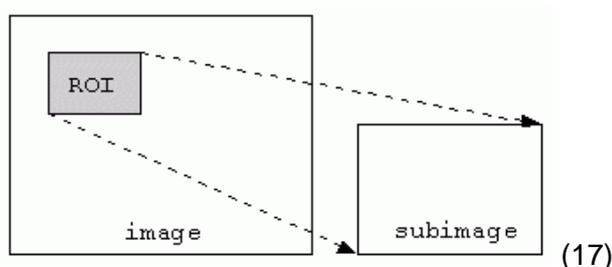


Ilustración 4 Imagen con región de interés definida

2.3.3 Transformación a escala de grises

Luego de haber captado los fotogramas del video se procede a la transformación de los fotogramas a escala de grises. La conversión de una imagen de color a una imagen en escala de grises no se realiza de forma única, sin embargo en su acercamiento más común, se trata de retener la información de la luminosidad y descartar los valores de tono y saturación. Asumiendo que los colores, rojo, verde y azul son señales, la aproximación de imagen en escala de grises a partir de una imagen de color

CAPÍTULO 2: Diseño e implementación del modelo para la detección de subtítulos e identificación del cambio de los mismos en fotogramas de video.

está dada por la ecuación que se muestra a continuación en donde 0 es el valor de menos intensidad, refiriéndose al color negro y 1 es el valor de mayor luminosidad o el color blanco (18).

$$GRAY = (0.30 * R) + (0.59 * G) + (0.11 * B)$$

Se crea una imagen temporal de tipo *IplImage* la cual inicialmente se le establece el mismo tamaño que el fotograma a procesar en ese momento. Luego se aplica la función *cvCvtColor* estableciéndole como parámetros la imagen origen, la imagen destino y la operación de conversión a llevar a cabo, es decir, el color al cual se procede a convertir la imagen.

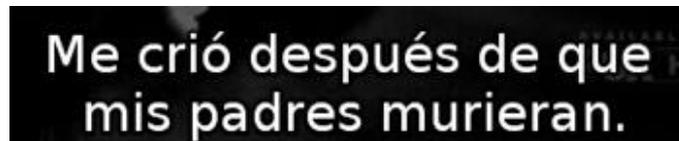


Ilustración 5 Imagen transformada a escala de grises

2.3.4 Detección de bordes

Una vez transformada la imagen a escala de grises se procede a la detección de bordes (Ver Anexo 2). Con la aplicación de dicha detección se obtiene como borde un cambio significativo en la imagen en el nivel de grises entre dos o más píxeles adyacentes. Los cambios abruptos en los valores de la función intensidad de una imagen están asociados a valores elevados de la primera derivada de dicha función (19). El operador más común de detección de bordes en análisis de imágenes es el gradiente del nivel de gris:

$$\nabla z(x, y) = [Z_x, Z_y] = \left[\frac{\theta z(x, y)}{\theta x}, \frac{\theta z(x, y)}{\theta y}, \right]$$

A esta se le aplica la técnica de detección de bordes Sobel mediante la función *cvSobel* la cual recibe como parámetros la imagen origen, la imagen destino, el valor de la derivada *x*, el valor de la derivada *y*, y el tamaño de la matriz de convolución.



Ilustración 6 Imagen con detección de bordes

CAPÍTULO 2: Diseño e implementación del modelo para la detección de subtítulos e identificación del cambio de los mismos en fotogramas de video.

2.3.5 Binarización

Luego de obtener la imagen con los bordes se procede a realizarle la binarización con el objetivo de separar el fondo de los objetos a analizar por lo que se define:

$$g(x, y) = \{1 \text{ si } f(x, y) > T\}$$

$$g(x, y) = \{0 \text{ si } f(x, y) \leq T\}$$

Para definir el tamaño del umbral se recurre al algoritmo iterativo:

1. Se obtiene un estimado inicial del umbral T .
2. Se calcula la media de los valores debajo (m_1) y sobre (m_2) del umbral T .
3. Se calcula el nuevo umbral: $T = \frac{1}{2}(m_1 + m_2)$
4. Se repite el paso 2 hasta que el nuevo umbral sea igual al umbral anterior (20).

A dicha imagen esta operación se le realiza mediante la función **cvThreshold** la cual recibe como parámetros la imagen que se analiza, la imagen donde se guarda la modificación, el valor umbral, el valor máximo a utilizar y el tipo de binarización a realizar.



Ilustración 7 Imagen binarizada

2.3.6 Dilatación

Una vez obtenida la imagen binarizada se dispone a aplicarle la dilatación. La misma operación se realiza por un elemento estructurante para conectar contornos de caracteres de una misma línea de texto. La dilatación es la transformación morfológica que combina dos vectores utilizando la suma (21). Formalmente se define por la siguiente ecuación.

$$A \oplus B = \{p \in E^2, p = a + b, a \in A \text{ y } b \in B\}$$

CAPÍTULO 2: Diseño e implementación del modelo para la detección de subtítulos e identificación del cambio de los mismos en fotogramas de video.

Esta operación se realiza mediante la función **cvDilate** la cual recibe como parámetros la imagen origen, la imagen destino, el valor del elemento estructurante utilizado para la dilatación y el número de iteraciones.



Ilustración 8 Imagen dilatada

2.3.7 Identificación de componentes conexas para la detección de regiones de subtítulos

Después de haber obtenido la imagen dilatada quedando en blanco las posibles regiones de texto se dispone a realizar la identificación de componentes conexas para la detección de los blobs que se encuentran en dichas regiones. El algoritmo que emplea es “Connected Component Algorithm o Algoritmo de Componentes Conexas, dicho algoritmo funciona mediante el escaneo de una imagen, píxel por píxel (de arriba a abajo y de izquierda a derecha) con el fin de identificar relacionadas regiones de píxeles, es decir, las regiones de los píxeles adyacentes que comparten el mismo conjunto de valores de intensidad V ” (22).

Este algoritmo se basa en un análisis que realiza para cada uno de los píxeles y una comparación que hace con los píxeles vecinos, reuniendo los que tengan valores similares, obteniendo de esta forma todas las regiones conexas. A cada una de estas regiones se le asignará una etiqueta característica de cada cual. Se realiza un filtrado por área a dichas regiones donde se detectan los blobs⁶ que cumplan con los tamaños especificados para ser definidos como texto y posible subtítulo. Los blobs que cumplan con los tamaños establecidos se guardan y se les dibuja su ubicación en el fotograma utilizando la función de OpenCV **cvDrawRect** conjunto con una estructura **cvPoint**. En este punto se crean 2 variables (**pointSUP** y **pointINF**) de tipo **cvPoint** y se le asigna los valores del tamaño de los blobs en cada iteración del ciclo. El valor que se obtiene de la resta entre el máximo y mínimo del punto **x** se compara con el valor de la resta entre el máximo y mínimo del punto **y**. Si el valor de la resta de los valores del punto **y** es mayor que los valores del punto **x**, es decir, si la dimensión del alto

⁶ librería de etiquetado de partes conexas en imágenes binarias.

CAPÍTULO 2: Diseño e implementación del modelo para la detección de subtítulos e identificación del cambio de los mismos en fotogramas de video.

del blob es mayor que la dimensión del largo del mismo el blob que se esa analizando se descarta y se elimina sino se dibuja las coordenadas de dicho blob y se procede verificar el siguiente.

2.3.8 Comparación entre fotogramas continuos para la detección de cambio de subtítulos y selección de fotogramas claves

Luego de ser detectadas las regiones de los blobs y guardada la ubicación de los que cumplen la condición para ser clasificado como subtítulo se procede a la detección del cambio de los subtítulos. Para la detección de cambio de subtítulos se realizó una comparación entre los blobs que encierran dichos subtítulos contenidos en fotogramas continuos con el objetivo de evitar cometer errores en la selección de los fotogramas claves y que en la acción del guardado de los fotogramas estos últimos se guarden con subtítulos repetidos. En cada comparación que se cumpla la condición de llama a un método local de guardado de archivos XML con las coordenadas de los blobs analizados. Para la detección de cambio de subtítulo se realizó una comparación del tamaño de los blobs de cada fotograma continuo en cada iteración. En este caso se guardan los blobs en una lista de estructura **QList** y se clona para otra lista de la misma estructura para que se posibilite en la otra iteración el llenado de la lista anterior y realizar comparaciones tanto del tamaño de las listas de blobs de cada fotograma como del tamaño de los mismos así como del área que ocupa cada blob.

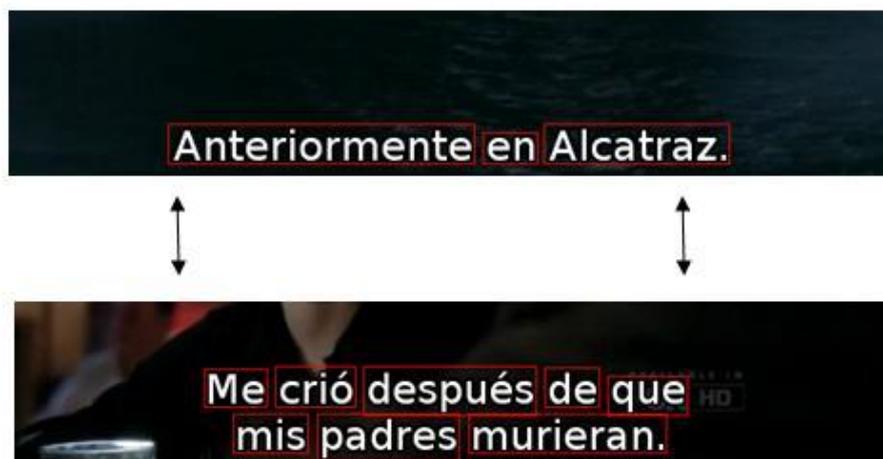


Ilustración 9 Imagen con comparación de áreas de blobs

2.4 Conclusiones parciales

Luego de la selección de diferentes funciones de la biblioteca OpenCV que trabajan con imágenes digitales conjunto con la definición de los requisitos funcionales y descripción de una propuesta para la

CAPÍTULO 2: Diseño e implementación del modelo para la detección de subtítulos e identificación del cambio de los mismos en fotogramas de video.

detección de subtítulos e identificación del cambio de los mismos en fotogramas de video, se concluye que la implementación de dicha propuesta cumple con el objetivo planteado. Para confirmar que el componente realmente cumple con las expectativas requeridas, se necesita llevar a cabo un grupo de pruebas que validen la propuesta ejecutada.

CAPÍTULO 3: Validación del componente propuesto.

3.1 Introducción

En el presente capítulo se describen los flujos de trabajos implementación y pruebas, donde se modela el diagrama de componentes y el diagrama de clases del diseño, posteriormente se documentan las diferentes pruebas que se le realizan al componente para validar su correcto funcionamiento donde se realizarán test para comprobar la eficiencia del componente propuesto.

3.2 Modelo de diseño

El modelo de diseño proporciona detalles acerca de la estructura de los datos, las arquitecturas, las interfaces y los componentes de software que son necesarios para implementar el componente. La meta de la ingeniería de diseño es producir un modelo de representación que muestre firmeza, comodidad y placer. Firmeza: el programa no debe tener ningún error que inhiba su función. Comodidad: un programa debe cumplir con los propósitos para los que fue creado. Placer: la experiencia de usar el programa debe ser agradable. Estos tres elementos de la teoría de diseño son conocidos como los principios del diseño.

3.2.1 Clases del diseño

Una clase de diseño es una abstracción de una clase o construcción similar en la implementación del sistema y tiene operaciones, parámetros, atributos y tipos; necesarios para su implementación en el lenguaje de programación elegido.

3.2.2 Diagrama de clases del diseño

Un diagrama de clases del diseño muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Este tipo de diagramas se utilizan para modelar la vista de diseño de un sistema, el cual incluye modelar el vocabulario del sistema, o modelar esquemas, que permiten al desarrollador tener una mejor visión de lo que va a implementar.

Los diagramas de clases del diseño también son la base para un par de diagramas relacionados: los diagramas de componentes y los diagramas de despliegue. Estos diagramas de diseño son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa.

CAPÍTULO 3: Validación del componente propuesto.

En la ilustración 10 se muestra el Diagrama de Clases del Diseño del Componente para detectar subtítulos presente en fotogramas de videos, en el cual se presentan todas las clases empleadas en la solución del problema, conjunto con sus atributos, operaciones y relaciones establecidas entre ellas.

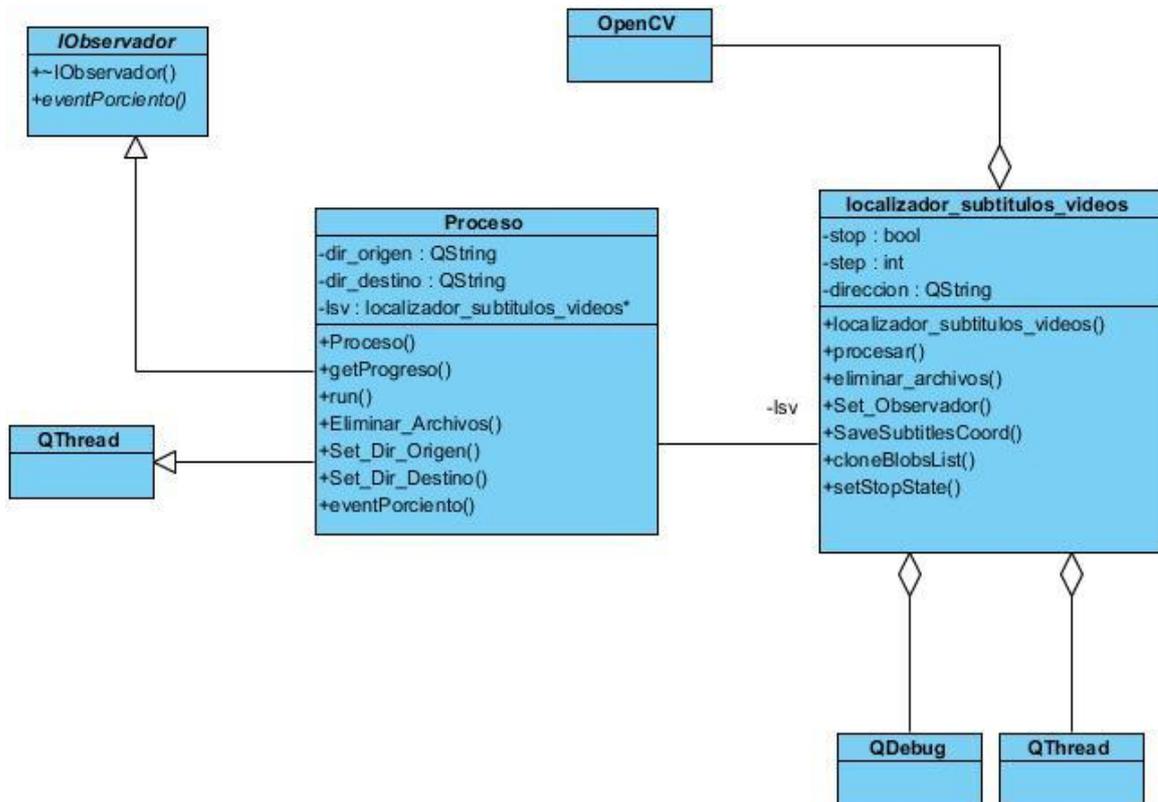


Ilustración 10 Diagrama de Clases del Diseño

En el Diagrama de Clases del Diseño del Componente para detectar subtítulos presente en fotogramas de videos, se muestran las clases que componen dicha aplicación, entre estas se encuentra la clase `localizador_subtitulos_videos`, dicha clase que es la más importante debido a que abarca los principales métodos para la realización de las funcionalidades que proporcionan la solución al problema de la investigación, tales como `procesar()`, `SaveSubtitlesCoord()`, `eliminar_archivos()` y `cloneBlobsList()`. También está presente la clase `proceso`, la cual implementa el método `run()`, este último realiza una llamada al método `procesar()` el cual realiza todo el procesamiento del video para dar cumplimiento al objetivo del componente. Esta clase hereda de `QThread` y de la clase `IObservador`. La clase `IObservador` se encarga de emitir el porcentaje de realización del análisis del video.

3.3 Modelo de Implementación

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes se puede encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos.

Uno de los artefactos que se generan en este modelo es el Diagrama de Componentes, el cual muestra un conjunto de elementos del modelo tales como componentes, subsistemas de implementación y sus relaciones. Se utiliza para modelar la vista estática de un sistema y muestra la organización y las dependencias lógicas entre un conjunto de componentes software.

3.3.3 Diagrama de componentes.

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de módulos y mostrar las relaciones entre los elementos de implementación. En la ilustración 11 se muestra la relación de dependencia que existe entre el paquete de clases Proceso con los paquetes de clases IObservador y Localizador_subtitulos_videos, así como la dependencia de este último con la biblioteca OpenCV. El paquete de clases contiene todos los componentes a desarrollar los cuales se agrupan en el mismo paquete debido a sus similares características.

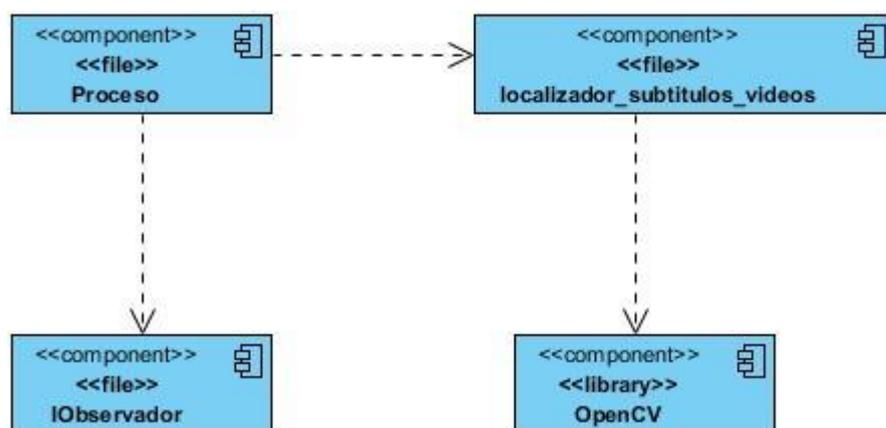


Ilustración 11 Diagrama de Componentes

3.4 Validación en base de datos heterogénea de video con subtítulos

Las pruebas de validación de videos con subtítulos se definen como una prueba a la que es sometido al componente con el objetivo de analizar y documentar si el mismo cumple con los objetivos trazados. Estas pruebas se sustentan en la necesidad de minimizar las falsas detecciones durante el proceso de segmentación de video para su representación sintética. La forma más utilizada para medir el buen funcionamiento del componente de detección de subtítulos y los cambios de los mismos consiste en calcular su recall (R) y su precisión (P). Para probar el algoritmo con materiales audiovisuales su definición es la siguiente:

$$R = \frac{\textit{correctos}}{\textit{correctos} + \textit{falsos negativos}}$$

$$P = \frac{\textit{correctos}}{\textit{correctos} + \textit{falsos positivos}}$$

Estas pruebas se realizaron macheando los diferentes resultados obtenidos (Ver Anexo 3) a partir de las corridas de los métodos utilizados en el componente contra una base de datos heterogénea de videos con subtítulos. Para su documentación se tuvo en cuenta:

- ✓ Nombre de la media que se analiza.
- ✓ Cantidad de fotogramas.
- ✓ Falsos positivos.
- ✓ Precisión del algoritmo.
- ✓ Falsos negativos.
- ✓ Recall.

Tras las corridas del algoritmo, utilizando el método de detección de subtítulos en videos diferente en cada ocasión, se obtuvieron los siguientes resultados.

Video	# fotogramas	FP	Precisión	FN	Recall
-------	--------------	----	-----------	----	--------

CAPÍTULO 3: Validación del componente propuesto.

Alcatraz	269	40	94.9	31	96
Make It Or Break It	281	30	94.8	17	96.9
The Big Bang Theory	274	35	92.8	26	94.5
Walking Dead	270	42	91.9	30	94

Tabla 1 Resultados de las pruebas realizadas al algoritmo

Analizando los resultados anteriores se puede observar que el algoritmo funciona con un recall y precisión considerablemente constante. El recall se mantiene entre 94 y 96 con número de falsos negativos inferior a 32 y la precisión se mantiene entre 91 y 94 con un número de falsos positivos entre 30 y 40. De forma general la precisión del algoritmo, es menor que su recall, esto se debe a su alta sensibilidad lo que proporciona la detección de un número mayor de falsos positivos.

3.5 Conclusiones parciales

Tras las pruebas realizadas se puede asegurar que el componente cumple con las necesidades de detección de subtítulos del proyecto CCM. Siendo capaz de realizar detecciones en varios tipos de videos a varias resoluciones. Se puede asegurar además que se está en presencia de un componente confiable, ya que se tiene una detección válida en el 94.4% de los casos.

CONCLUSIONES GENERALES

En la presente investigación se cumplieron satisfactoriamente los objetivos trazados, llevando a cabo cada una de las tareas propuestas. A continuación se brindan algunos elementos a modo de conclusión:

- ✓ Se logró garantizar la detección y localización de subtítulos además de agilizar el proceso evitando la repetición de caracteres o palabras.
- ✓ El uso de las técnicas de detección de bordes específicamente el Operador Sobel garantiza un componente con una alta escalabilidad⁷, capaz de procesar todos los fotogramas de varios videos.
- ✓ Con el uso del framework⁸ QT se logró crear un componente con un alto grado de reusabilidad, el mismo puede ser utilizado de forma sencilla por cualquier proyecto de la universidad que desee detectar regiones de subtítulos en videos.

⁷ Se define como la propiedad deseable de un sistema, una red o un proceso, que indica su habilidad para extender el margen de operaciones sin perder calidad, o bien manejar el crecimiento continuo de trabajo de manera fluida.

⁸ Es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado.

RECOMENDACIONES

Continuar el desarrollo de esta investigación con el fin de incorporar elementos que permitan la localización de subtítulos utilizándola detección de componentes conexas en 3 dimensiones mediante la técnica de marcado.

BIBLIOGRAFIA

BIBLIOGRAFÍA

1. **WordReference.com.** WordReference.com. [En línea] WordReference.com, 2012. <http://www.wordreference.com/definicion/subtitulo>.
2. **Lacombe Rocha, Claudia.** Seminario El documento electrónico, su gestión y los servicios a la ciudadanía desde una perspectiva archivística. [En línea] 2007. archivistica.blogspot.com/2007_04_01_archive.html.
3. **Senso, José A y de la Rosa Piñero, Antonio.** *El concepto de metadato. Algo más que descripción de recursos electrónicos.*
4. **Aprendizaje, Entorno Virtual de.** Entorno Virtual de Aprendizaje. [En línea] 2009. http://eva.uci.cu/file.php/72/Bibli_1/Introduccion_al_Video_y_Sonido_Digital.pdf.
5. **Constenla, Gabriel Roberto.** *Reconocimiento Óptico de Dígitos con Redes.* s.l.: Facultad de Tecnología Informática, 2010.
6. **Industrial, Grupo de Tecnología.** Detección de bordes en una imagen. [En línea] 2002. <http://www.isa.umh.es/doct/pava/DeteccionBordes.pdf>.
7. **Rebaza, Jorge Valverde.** Detección de bordes mediante el algoritmo de Canny. [En línea] 2007. <http://www.seccperu.org/files/Detecci%C3%B3ndeBordes-Canny.pdf>.
8. **REPOSITORIO_DIGITAL.** *Operaciones con imágenes.* 2010.
9. **Imagen, Laboratorio de Procesado de.** Realce de imagen. Tratamiento de la Información I. [En línea] 2004. <http://poseidon.tel.uva.es/~carlos/ltif10001/realce.pdf>.
10. **Valladolid, Universidad de.** Tipos de traducción. [En línea] 2012. <http://tipos-de-traduccion.wikispaces.com/>.
11. **Psicologia2000.** Portal On-line. [En línea] Psicologia2000, 2011. <http://www.psicologia2000.com/es/enciclopedia-general-psicologia-on-line-wiki-letra-b/28834-biblioteca-informatica.html>.
12. **Bogotá, Universidad Javeriana.** Opencvjaveriana . [En línea] 2009. <http://opencvjaveriana.wikispaces.com/>.

BIBLIOGRAFIA

13. **Tucumán, Facultad Regional.** Nacional. [En línea] Universidad Tecnológica Nacional de Argentina, 2012. <http://www.frt.utn.edu.ar/sistemas/paradigmas/lenguajes.htm>.
14. **Schildt, Herber.** C ++. Guía de autoenseñanza. *C ++. Guía de autoenseñanza*. España : s.n., 2001.
15. **Nokia.** QT home. [En línea] Nokia, 2008. <http://qt.nokia.com/products/developer-tools?currentflipperobject=821c7594d32e33932297b1e065a976b8>.
16. **Kendall, K.** *Análisis y Diseño de Sistemas*. 2005.
17. UML, BPMN and Database Tool for Software Development. [En línea] Visual Paradigm. <http://www.visual-paradigm.com>.
18. **Software, Visión de Synergix de los Sistemas de Información y la Ingeniería del.** Tipos de requisitos: Funcional vs No Funcional. [En línea] 2008. <http://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional/>.
19. **nashruddin.com.** Free PHP scripts, tutorials and downloads. [En línea] 2012. http://nashruddin.com/OpenCV_Circular_ROI.
20. **González, Rafael Alberto González.** Procesamiento Digital de Imágenes, Capítulo 2. [En línea] 2010. http://catarina.udlap.mx/u_dl_a/tales/documentos/mel/gonzalez_g_ra/capitulo2.pdf.
21. **T. Hermosilla, E. Bermejo, A. Balaguer, L.A. Ruiz.** Detección de bordes con precisión subpíxel en imágenes digitales. [En línea] 23 de noviembre de 2006. http://www.cgat.webs.upv.es/bigfiles/Charla_Hermosilla_JMA_06.pdf.
22. **UDLAP.** Técnicas de procesamiento digital de imágenes. [En línea] Universidad de las Américas Puebla. http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/nieto_b_d/capitulo2.pdf.
23. **Molina, Rafael.** *Introducción al procesamiento y análisis de imágenes digitales*. Granada : s.n., 1998.
24. **Mary, Queen.** Sciencie, School of Electronic Engineering and Computer. [En línea] 2005. <http://www.eecs.qmul.ac.uk/~phao/CIP/Labs/handouts/connected.pdf>.

ANEXOS

Anexo 1. Elaboración del modelo para la detección de subtítulos e identificación del cambio de los mismos en fotogramas de video.



Ilustración 12 Modelo para la detección de subtítulos

Anexo 2. Esquema del algoritmo de detección basado en bordes

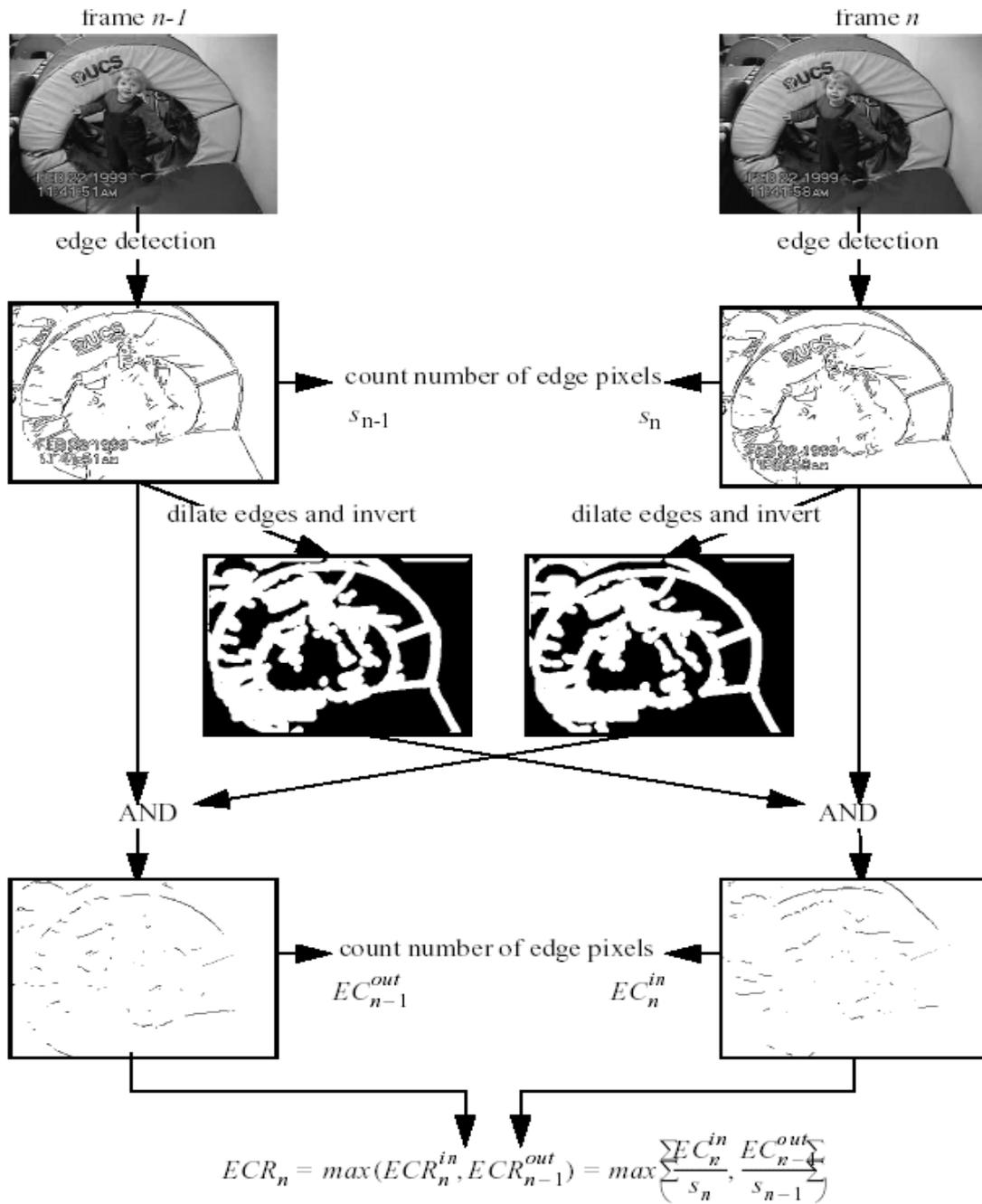


Ilustración 13 Esquema del algoritmo de detección basado en bordes

Anexo 3. Resultado de la ejecución del algoritmo.



Ilustración 14 Diferentes resultados de la ejecución del algoritmo

GLOSARIO

Gradiente: Vector que se encuentra normal a una superficie o curva en el espacio a la cual se le está estudiando, en un punto cualquiera. Mide la tasa y la dirección del cambio en un campo escalar; el gradiente de un campo escalar es un campo vectorial.

Archivos: Espacio que se reserva en el dispositivo de memoria de un computador para almacenar porciones de información que tienen la misma estructura y que pueden manejarse mediante una instrucción única.

Patrón: Disciplina de resolución de problemas reciente en la ingeniería del software que ha emergido en mayor medida de la comunidad de orientación a objetos, aunque pueden ser aplicados en cualquier ámbito de la informática y las ciencias en general.

Morfología: Estudio que se ocupa de la constitución de una unidad menor que es la palabra. Se ha considerado la morfología como el estudio orientado a describir (y caracterizar) las partes constituyentes de las palabras; otra postura se inclina por establecer modelos de formación de palabras atendiendo a sus rasgos gramaticales; una tercera vía pretende describir la estructura de la palabra desde los mecanismos de formación de la palabra.

Reflectancia: Relación entre el rayo incidente y la radiación reflejada por éste en una superficie. También llamado coeficiente de reflexión.

Píxel: Superficie homogénea más pequeña de las que componen una imagen, que se define por su brillo y color.