

Universidad de las Ciencias Informáticas

Facultad 6



Título: Implementación de un componente para la separación automática de segmentos de videos integrado al Sistema de Captura y Catalogación de Medias.

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Autor: Luis Angel Pupo Rodríguez.

Tutor: MSc. Héctor Raúl González Díez.

Ciudad de la Habana, 24 de junio de 2011.

Dedicatoria.

*A la memoria de mi papá, aunque ya no este aquí, siempre lo tengo
presente.*

A mi mamá, por ser siempre su deseo y mi meta.

A Alberto, por darme la dicha de ser su hijo.

A Aty, ya que este logro es de ambos.

A mis abuelos, por existir.

Agradecimientos.

*A mi mamá que siempre está a mi lado.
A Alberto, por hacerme la persona que soy.
A mis abuelos, por todo lo que hacen por mí.
A Afy, por darme tanto amor, por estar siempre a mi lado y ayudarme
en este logro.
A toda mi familia, por el apoyo brindado.
A mi tutor, por toda la ayuda que me brindó para realizar este
trabajo.
A mi oponente, por ayudarme a crear un trabajo con calidad.
Al tribunal por guiar la investigación por el camino correcto.
A todos los integrantes de mi proyecto SCCM.
A mis amigos, los que conocí en la UCI y los de antes.*

Declaración de autoría.

Declaro que soy el único autor de este trabajo y autorizo al _____ 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 _____.

Luis Angel Pupo Rodríguez

MSc. Héctor Raúl González Díez

Resumen:

El objetivo del presente trabajo es desarrollar un componente de software que permita separar de manera automática segmentos de video de noticias integrado con el producto Sistema de Captura y Catalogación de Medias. Se propone un método basado en umbrales globales para los métodos de detección de cambios de toma. Se hace un análisis de varios descriptores visuales siendo los métodos de histograma, diferencia de bloque de píxel y un descriptor estadístico de segundo orden los que arrojaron los mejores resultados. Se hace un agrupamiento jerárquico de alto nivel para obtener el resumen del video. En la construcción del resumen de video se emplea el concepto de escalabilidad de modo que se puede obtener con una sola corrida del video múltiples resúmenes a solicitud del usuario.

Palabras claves.

“Agrupamiento jerárquico, fotogramas claves, resumen de video, tomas de video”.

Abstract.

The objective of this work is to develop a software component that allows to automatically separating video news segments integrated with the system capture and cataloging medias. We propose a method based on global thresholds for change detection methods making. An analysis of various visual descriptors to be the methods of histogram difference pixel block and a second-order statistical descriptor that yielded the best results. There is a high-level hierarchical clustering to obtain a summary of the video. In building the video summary to the concept of scalability so you can get with a single run of multiple video summaries at the request of the user.

Keywords: “shot, hierarchical clustering, keyframe, scalability, video summarization”.

Índice.

INTRODUCCIÓN	1
CAPÍTULO 1: Fundamentación teórica de la generación automática de resúmenes de video	7
1.1. Conceptos asociados al dominio del problema	7
1.2. Cambios de tomas	9
1.2.1. Cambios de tomas abruptos	10
1.2.2. Cambios de tomas graduales	10
1.3. Etapas para la detección de cambios de tomas	11
1.4. Tendencias en la detección de cambio de toma en video	11
1.5. Estrategias de resúmenes de video	13
1.6. Métodos para el cálculo de diferencias	15
1.6.1. Método basado en píxel	16
1.6.2. Método basado en detección de bordes	17
1.7. Métodos de clasificación	18
1.7.1. Umbral	18
1.7.2. Redes neuronales	18
1.7.3. Máquinas de soporte vectorial (SVM)	19
1.7.4. K-means	21
1.8. Definición de la tecnología de desarrollo	21
1.8.1. Librerías	21
1.8.2. Lenguaje de programación	24
1.8.3. Entorno de desarrollo (IDE)	26
1.9. Conclusiones parciales	27
CAPÍTULO 2: Descripción y diseño de algoritmos y componente para la separación automática de segmentos de video	28

2.1.	Selección de requisitos funcionales del componente de software.....	28
2.2.	Descripción general de la propuesta de algoritmo	28
2.3.	Decodificación del video.....	30
2.4.	Detección de cambios de tomas.....	31
2.4.1.	Cálculo de distancia por bloque de píxeles.....	32
2.4.2.	Cálculo de distancia por histograma de color.....	33
2.4.3.	Cálculo de distancia a partir de las características (bordes).....	34
2.4.4.	Cálculo de distancia a partir de una medida estadística LR1.....	36
2.5.	Ajuste de la función de distancia.....	36
2.6.	Detección de cambios de tomas.....	37
2.7.	Selección de fotogramas claves.....	38
2.8.	Agrupamiento de alto nivel.....	38
2.9.	Exportación de fotogramas como imágenes.....	39
2.10.	Integración.....	40
2.11.	Conclusiones parciales.....	40
CAPÍTULO 3: Validación del componente propuesto.....		42
3.1.	Pruebas de soporte de formato.....	42
3.2.	Pruebas a algoritmos de detección de cambios de tomas.....	44
3.3.	Conclusiones parciales.....	46
Conclusiones Generales.....		47
Recomendaciones.....		48
Bibliografía y referencias bibliográficas.....		49
Anexos.....		52
Anexo 1. Figuras.....		52
Glosario.....		56

INTRODUCCIÓN

Desde los inicios mismos de la informática se trató no solo de hacer cálculos aritméticos, sino también trabajar con otros archivos que contenían más información como es el caso del audio y las imágenes digitales, y la unión de estos, el video digital. Con el paso de los años el trabajo con estos archivos se hizo más común y por ende creció el volumen y la necesidad de estudiarlos y analizarlos en detalle.

En la actualidad, el análisis de grandes volúmenes de video resulta un problema complejo, pues se requieren tecnologías potentes para el procesamiento de cada una de las partes que componen el video. Resaltar que por cada segundo se deben procesar alrededor de 24 a 30 imágenes (Fotogramas), por lo que resulta inviable el análisis de todos estos datos de manera simultánea, si se tiene en cuenta la calidad con que estas fueron capturadas y la complejidad de los algoritmos de procesamiento existentes.

Por tales razones, existen enormes esfuerzos por desarrollar algoritmos de segmentación de videos, los cuales son la base de cualquiera de las técnicas existentes para el análisis de información contenida en los diferentes formatos de videos (Cano, 2001). Algunas de las funciones de estos algoritmos de segmentación de video son:

1. Generar resúmenes, en videos de gran tamaño, de modo que queden reflejados los elementos de mayor interés como parte del mismo. Adicionalmente en el propio proceso de edición de los contenidos resulta de gran valor para el editor poder acceder solo a aquella parte del video que es relevante para el proceso de edición.
2. En los grandes archivos de contenidos audiovisuales de emisoras de difusión masiva, resulta indispensable durante el proceso de digitalización o en el almacenamiento propiamente dicho, ubicar índices asociados a los contenidos de modo que se puedan disponer de la información en el momento que se

requiera de una manera eficiente permitiendo a los usuarios acceder a la información del archivo de manera directa.

En este sentido se reduce el problema de procesar todos los fotogramas por los que está compuesto el video a un grupo de estos que son representativos, por el nivel de similitud que poseen con los que aparecen en su entorno. Además desde el punto de vista de la representación de videos en diferentes herramientas y tecnologías resulta muy cómodo para un usuario acceder a la información que este contiene de manera sintética de modo que se facilita la navegabilidad para conocer el contenido o temática abordada en cada uno de los archivos.

El departamento de señales digitales del centro GEySED¹ de la Universidad de las Ciencias Informáticas, cuenta entre sus líneas de investigación con el análisis de contenidos audiovisuales. En este centro se desarrollan diversos proyectos en los que se representa información audiovisual contenida en grandes archivos de datos.

En dos de los módulos de desarrollo del proyecto Sistema de Captura y Catalogación de Medias (SCCM), en los que se realiza la visualización de videos, se necesita hacer un resumen de los fotogramas principales de los videos con los que trabaja el usuario para brindar una breve información visual del material. Actualmente se seleccionan los fotogramas dentro del video cada un tiempo estático, este tiempo es escogido por el desarrollador por lo que puede ser cualquiera, por esta razón se desconocen cuales fotogramas fueron extraídos. Debido a esto, al finalizar el proceso, el resumen visual obtenido no tiene la calidad requerida, esta mala calidad está dada fundamentalmente a que se obtienen fotogramas repetidos, oscuros, difusos o de una baja información visual.

Por lo antes expuesto se puede definir el siguiente **problema a resolver**: necesidad de minimizar las falsas detecciones durante el proceso de detección de

¹ Centro de Desarrollo Geoinformática y Señales Digitales.

cambio de toma en video para su representación sintética en el Sistema de Captura y Catalogación de Medias.

Se define como objeto de estudio: la separación automática de tomas de video, y como **campo de acción:** la separación automática de tomas de video a partir de los descriptores visuales en contenidos de videos de noticias.

El objetivo general es: Implementar un componente de software que permita separar de manera automática segmentos de video de noticias, para su representación sintética, integrado con el Sistema de Captura y Catalogación de Medias.

Para dar cumplimiento al objetivo general de la presente investigación se proponen los siguientes **objetivos específicos.**

1. Implementar los algoritmos propuestos para la separación del video en tomas a partir de descriptores visuales que se utilizan como parte de la propuesta.
2. Desarrollar una estrategia para la representación sintética del video haciendo uso de técnicas de agrupamiento.
3. Implementar el componente de software para la separación automática de cambios de tomas en video que se integre al Sistema de Captura y Catalogación de Medias.

La idea a defender establecida en este trabajo es que un componente para la separación automática de videos de noticias basado en cambios de tomas y agrupamiento garantizará una mejor información visual en la representación sintética del video con un mínimo de falsas detecciones.

Para guiar la presente investigación se definieron las siguientes **tareas de la investigación:**

1. Describir el funcionamiento de los algoritmos para detección de cambios de tomas.

2. Caracterizar los métodos y algoritmos de segmentación de videos que permita la selección de las características a utilizar como parte de la segmentación temporal de video.
3. Describir el funcionamiento de los algoritmos de clasificación de video para la identificación de fotogramas claves.
4. Definir la tecnología a usar para el desarrollo de los algoritmos.
5. Desarrollar un componente de software para la detección de cambio de toma en video.
 - 5.1. Seleccionar los algoritmos a utilizar para la separación de segmentos de videos.
 - 5.2. Diseñar el algoritmo de segmentación de videos seleccionado.
 - 5.3. Implementar el algoritmo de segmentación de videos seleccionado.
 - 5.4. Implementar el componente para la separación de segmentos de videos.
6. Validar el funcionamiento del algoritmo propuesto a partir de bases de datos internacionales y métodos estadísticos.

Métodos Científicos.

Para validar la idea a defender planteada, y en correspondencia con el objetivo y las tareas propuestas se aplicaron los siguientes métodos científicos durante el desarrollo de la investigación:

Métodos teóricos:

- **Análisis - Síntesis:** Permiten llegar a conclusiones, a partir del estudio realizado del tema, en diferentes fuentes utilizadas para procesar la información obtenida, para definir la propuesta de algoritmo a partir de la información obtenida de las diferentes fuentes consultadas así como del conocimiento generado durante la presente investigación.

- Inducción – Deducción: Permite establecer la idea a defender que guiará el desarrollo de la investigación, de modo que se puedan generalizar los resultados de la presente sobre la base de los resultados obtenidos con las muestras utilizadas durante las pruebas.
- Histórico – Lógico: Para conocer el comportamiento y evolución de los diferentes enfoques existentes en la separación automática de videos así como los métodos propuestos para su evaluación.

Métodos Empíricos:

- Experimento: Para validar los resultados obtenidos durante la investigación a partir de la selección de un conjunto de métricas que permitan evaluar la eficiencia de los algoritmos.

Principales Aportes.

Como Aporte práctico de la investigación: Se desarrollará un componente de software para la separación automática de segmentos de video que sirva como extensión a algunos de los productos que se desarrollan en la UCI relacionados con los audiovisuales, específicamente con videos.

Estructura del documento.

En el Capítulo 1 se hace una valoración de los diferentes enfoques existentes relacionados con segmentación automática de video. Se describen los algoritmos para identificación de las características que componen un video, además se describen algunos de los algoritmos de clasificación de video existentes.

En el Capítulo 2 se describen las herramientas para implementar el algoritmo propuesto. Se describen cada uno de los métodos que componen el algoritmo propuesto. Finalmente se describen los mecanismos previstos de detección de cambio de toma en video.

En el Capítulo 3 se hace un análisis de los resultados obtenidos a partir de las bases de datos de pruebas definidas y los criterios de medidas que permitan evaluar los resultados obtenidos.

CAPÍTULO 1: Fundamentación teórica de la generación automática de resúmenes de video.

En este capítulo se presentan los principales enfoques de las técnicas de generación automática de resúmenes de video. Se hace un análisis de los conceptos asociados al problema que dio origen a la investigación. Existen diversas tendencias en la generación de resúmenes automáticos de videos, aunque la presente investigación se centra en aquellos que se basan en la extracción de fotogramas claves.

1.1. Conceptos asociados al dominio del problema.

Se hace necesario partir de un conjunto de definiciones que resultan el sustento teórico de la presente investigación.

Video Digital:

Tomando como base la definición de video digital propuesto en (Video Summarization Using Greedy Method in a Constraint Satisfaction Framework) se propone la siguiente definición de video digital.

Sea V una secuencia de fotogramas $I_1 I_2 \dots I_n$ que son indexados a una frecuencia de grabación constante fps dando una sensación de movimiento a los contenidos que se encuentran en cada imagen. En la definición I_i representa el fotograma i -th dentro del video, n indica la cantidad de fotogramas del video, mientras fps la cantidad de fotogramas que componen el video por cada segundo. Cada fotograma representa la unidad básica de información del video.

Resúmenes de video:

Las técnicas de resúmenes automáticas de video tienen como propósito obtener, a partir de un video, una versión reducida del mismo con los contenidos más relevantes. Existen dos modalidades de resúmenes de video que son las que mayor aceptación han tenido por los usuarios de estos sistemas. Estos tipos de resúmenes de video son los *Storyboard* y los *skimming* de video.

En el caso de los *storyboard* consisten en un conjunto de imágenes estáticas (Fotogramas claves) que representan las escenas más relevantes del video. Por lo general se utiliza para navegar por los contenidos del video de forma inmediata.

En el caso de los *skimming* de video consisten en la agrupación de pequeños segmentos de videos, donde cada segmento contiene la información más relevante del video. Estos pequeños segmentos de video se conectan a partir de diferentes tipos de efectos visuales para obtener un resumen del video original.

Para obtener el resumen de video en alguna de las modalidades anteriormente descrita es necesario tener en cuenta la estructura del video. En este sentido existen tres pasos que se deben tener en cuenta para obtener el resumen de video.

- Extracción de los fotogramas claves a partir de la detección de los cambios de tomas.
- El análisis de la información redundante en el video.
- La estrategia para la generación de los resúmenes de video.

La presente investigación se centrará en los resúmenes de videos *storyboard*, debido a que es el tipo de resumen que satisface las necesidades del producto SCCM.

A continuación se describen los principales enfoques consultados en la literatura de cada una de estas etapas. De igual manera se describen algunos conceptos asociados a cada una de ellas.

Detección de cambios de tomas y extracción de fotogramas claves.

Para comprender los diferentes métodos de detección de cambios de tomas se parte de un conjunto de definiciones necesarias.

Toma:

(Wandelmer, y otros, 2006) Define como toma de video a la unidad sintáctica mínima en la que se estructuran los videos. Comprende las imágenes registradas durante una operación de la cámara, esto es, desde que se pulsa el botón de grabación hasta que se detiene la acción. Una misma toma puede contener varios planos.

Sobre esta base se asume de manera formal la siguiente formulación matemática:

Sea S_i la i -th toma de una secuencia de video V cada toma se compone de un conjunto de fotogramas I_k .

1.2. Cambios de tomas.

La transición entre dos tomas, es decir la unión de dos de estas, aparece en el proceso de post-producción mediante técnicas de edición de video. En tal sentido el conjunto de tomas que fueron capturadas por una cámara se editan para eliminar partes de la toma que resultan contenido indeseado (errores de grabación, etiquetas usadas en la grabación, imágenes que quedaron en negro o con el patrón de la cámara, etc.). Por último se unen estas tomas, para conformar el video final, utilizando efectos de edición los cuales son conocidos como transiciones. Esta transición puede ser abrupta o gradual dependiendo del efecto que se quiera lograr.

En el caso de las **transiciones abruptas** es el resultado de un efecto de corte en el video. En este sentido se dice que existe una transición abrupta si los fotogramas I_k e I_{k+1} pertenecen a tomas diferentes. Esta transición se detecta claramente debido a la discontinuidad visible en la secuencia de fotogramas. Este tipo de transiciones son conocidas por CUT.

En el caso de las **transiciones graduales** es el resultado de la aplicación de efectos de edición de video más complejos, en los que se ven involucrados varios fotogramas. Por lo tanto se dice que existe una transición gradual si los fotogramas I_k e I_{k+N} pertenecen a tomas diferente. Donde N representa el número

de fotogramas que conforman la transición. Algunos ejemplos de estas transiciones son los fade, dissolves, etc (Cano, 2001) (Jesús Bescós, APRIL 2005).

1.2.1. Cambios de tomas abruptos.

Se caracterizan porque emplean un solo fotograma para enlazar dos tomas. Son las transiciones más abundantes existentes en los videos y existe únicamente un tipo de efecto de edición representativo de las transiciones abruptas, el corte (Sáes Peña, 2006).

La detección de este tipo de cortes es un problema ampliamente estudiado y muy fácil para hacer frente en la mayoría de los casos. El porcentaje de éxito reportado por los autores se encuentra actualmente por encima del 95% de recall y 95% de precisión, calculado sobre una muestra máxima de alrededor de 300 cortes. Sin embargo, no es fácil comparar los resultados reportados objetivamente (Jesús Bescós, APRIL 2005).

1.2.2. Cambios de tomas graduales.

Estos cambios de tomas no se realizan usando un solo fotograma como lo hacen los abruptos, sino que, utilizan varios fotogramas para cambiar de una toma a otra. Esta unión gradual entre tomas se realiza con la aplicación de varios efectos de edición como pueden ser el dissolve, que es posiblemente el más utilizado, los fades (fade-in y fade-out), wipes, mattes, etc. (Sáes Peña, 2006).

Con frecuencia son utilizados para incorporar información semántica al video, como por ejemplo para indicar un cambio de escena. Su detección es sustancialmente más complicada que la detección de transiciones abruptas, pues al ocurrir de forma gradual quedan habitualmente enmascaradas por el movimiento presente en el video (Lu, y otros, 1999). Además, la gran variedad de transiciones graduales existentes es otro aspecto que dificulta su detección. Por esto la gran complejidad de los algoritmos que se usan para detectarlos.

1.3. Etapas para la detección de cambios de tomas.

Los algoritmos de detección de cambios de tomas se basan en el cálculo de la diferencia entre las características visuales de los fotogramas I_k y I_{k+N} . En el caso particular en que $N=1$ los fotogramas son consecutivos.

Existen tres enfoques fundamentales para detectar cambios de tomas que durante varios años (2000-2007) han sido objeto de prueba en los eventos TRECVID².

1. Análisis de Histograma (15 grupos han utilizado este enfoque).
2. Análisis de los valores de intensidad del píxel (8 grupos han utilizado este enfoque).
3. Análisis de bordes (5 grupos han utilizado este enfoque).

La forma en que se han empleado cada una de estas métricas van desde métodos donde se utilizan cada una de ellas de manera independiente hasta métodos más complejos donde se combinan varias de ellas para obtener una métrica más robusta. De manera particular cuatro de los grupos con mejores resultados en el TRECVID 2005 utilizaron la combinación de estas tres técnicas como parte de sus algoritmos de detección de cambios de tomas (Alan, y otros, 2010). Como salida de estos, se obtiene un vector de distancias por cada métrica empleada con valores cercanos para aquel par de imágenes que pertenecen a una misma toma y ocurre un cambio brusco cuando aparece una nueva toma.

Luego de la extracción de características es necesario desarrollar una estrategia de clasificación que permita separar este vector de características en dos clases (Transición y no transición).

1.4. Tendencias en la detección de cambio de toma en video.

² Del ingles Text REtrieval Conference

Las primeras investigaciones que están registradas con relación a la detección de cambio de toma en video son de antes de la década de los 90³, en la literatura estudiada se encontraron dos [(Coll, D.C. y Choma, G.K, 1976) (Seyler, A, 1965)]. A partir del año 1990 fue creciendo el interés y con este, el número de investigaciones relacionadas con el tema, a la par de estos trabajos se hicieron otros para validar el buen funcionamiento de los algoritmos propuestos, entre estos trabajos se encuentran [(A Survey of Technologies for Parsing". Journal of Visual Communication and Image, 1996) (Brunelli, R.; Mich, O. y Modena, C.M, 1999) (Idris, F. y Panchanathan, S, 1997)].

Existen trabajos más recientes que direccionan su esfuerzo en evaluar diversas métricas de extracción de características tanto en espacios comprimidos como no comprimidos.

En (Jesús Bescós, APRIL 2004) se hace un estudio comparativo de ocho métricas utilizando videos en espacios comprimidos y no comprimidos. Se emplea el criterio de la divergencia aplicado sobre el espacio de color YCrCb. Se diseñaron pruebas de cada métrica individual y la combinación de cada una de ellas en pares para evaluar la efectividad de cada una de estas métricas. Como resultado se evidencia que la combinación de varias métricas en todos los casos arroja mejores resultados para la separación del video en tomas, que el uso de una de ellas de manera independiente. También se plantea que no es posible decidir que métrica es mejor que otra sino que existen unas que tienen mejor comportamiento en espacios comprimidos y otras en espacios no comprimidos. El autor propone la construcción de un espacio 3D de valores utilizando las métricas *diferencia de bloque de píxel* y *LR1* las cuales arrojaron los mejores resultados en espacios comprimidos. Para la construcción de los espacios 3D de valores se hace un análisis estadístico local para obtener los parámetros λ_1 y λ_2 . Por último se utiliza en el proceso de clasificación un método basado en un tetraedro con umbrales por cada métrica.

³ Década a partir de 1990.

En algunos casos las investigaciones se pueden clasificar en dos grupos, unas basadas en el análisis de los videos no comprimidos y otras que investigan el procesamiento de los videos comprimidos. Esta clasificación no puede ser radical, debido a que algunas sirven para el trabajo con ambos tipos de videos, es decir, tanto con los de un dominio comprimido como con los que no tienen compresión alguna.

En la comunidad científica en general, la Detección de cambio de toma en video es actualmente el punto de entrada a cualquier técnica de análisis de video; por ello, existen multitud de trabajos [(Cano, 2001) (Sáes Peña, 2006) (Ziyou , y otros) (Jesús Bescós, APRIL 2005)] que de un modo u otro usan este tipo de técnica.

Por lo general, los algoritmos de detección de cambios de tomas se basan en el cálculo de la diferencia entre los valores de las características de dos fotogramas consecutivos, este valor es posteriormente procesado a través de algún método de clasificación que permita determinar la existencia o no de un cambio de toma (Jesús Bescós, APRIL 2005). Estas técnicas son caracterizadas principalmente por dos aspectos:

1. La función de la distancia diseñada para evaluar la similitud entre dos fotogramas. Esto incluye varios sub-aspectos; las características del fotograma, la métrica o la función de evaluar disparidad entre fotogramas, y la forma de aplicar dicha función (ya sea a los fotogramas generales o subregiones, ya sea para fotogramas consecutivos o fotogramas separados por un paso, ya sea directamente a los fotogramas o filtrada o compensación de movimiento- fotogramas, etc.).
2. Por último las políticas de umbral aplicadas para tomar una decisión.

1.5. Estrategias de resúmenes de video.

Las estrategias de resumen de video han sido abordadas desde diversos enfoques. Algunos autores trabajan los métodos de resumen de video sobre la base de obtener como salida un único resumen del video. Sin embargo trabajos

más actuales se dirigen hacia la generación jerárquica de varios resúmenes para un mismo video de modo que se explote al máximo las estructuras del mismo (Benini, y otros, 2006). Sobre esta base se introduce el concepto de escalabilidad (Zhu, y otros, 2004). con el propósito de obtener múltiples resúmenes de video a partir de un agrupamiento jerárquico de los fotogramas claves. Este concepto resulta de mucha importancia en la navegación por los fotogramas del video y dependiendo del nivel de detalle que se desee por el usuario se obtendrá mayor o menor información.

Existen disímiles trabajos que utilizan el agrupamiento jerárquico de fotogramas para escalar los resúmenes de video (Herranz, y otros, 2010) (Benini, y otros, 2006) (Zhu, y otros, 2004) . La idea general de estos trabajos consiste en construir una jerarquía de fotogramas siguiendo el criterio del vecino más cercano. Se utilizan medidas visuales de distancias similares a las descritas en la sección de cambios de tomas para construir la matriz de distancia entre fotogramas. Con el dendograma construido se aplica algún criterio de umbral para definir las agrupaciones que se generan. A partir de estructurar las tomas en grupos se siguen diversas estrategias para escalar los niveles de resumen de los videos.

En el trabajo (Herranz, y otros, 2010) se utiliza a partir de la construcción de los grupos de tomas una medida estadística, con pesos por cada componente, sobre la base de la duración de cada grupo y la distancia entre los fotogramas que componen el grupo. Esta medida estadística es un índice de relevancia de cada grupo el cual se utiliza para las diferentes salidas que el usuario requiere como parte del resumen del video. De manera similar se presenta una medida de relevancia (Chen, y otros, 2010) con la diferencia que el autor trabaja de manera directa sobre las tomas detectadas.

En (Benini, y otros, 2006) se utiliza como criterio de distancia para el agrupamiento jerárquico de cada fotograma clave extraído, un TSVQ codebook el cual contiene información semántica de cada fotograma clave seleccionado. Con

esto se construye una medida de dispersión la cual se emplea tanto para construir la distancia entre los fotograma clave como para extraer los resúmenes del video.

En (Gong, y otros, 2000) se construye un vector de características por cada fotograma extraído a una frecuencia de 3fps usando el histograma de color en el espacio RGB. Con todos estos vectores se construye una matriz sobre la cual se calcula el SV D. Para obtener las características refinadas de cada fotograma con las cuales se utiliza un método de agrupamiento jerárquico para obtener los grupos que componen el resumen. En el caso de esta propuesta implica un costo computacional aunque no deja de ser interesante.

Otros trabajos están dirigidos a métodos basados en la teoría de grafos (Ngo, y otros, 2003) los cuales son otra forma de construir de manera jerárquica los grupos. En estos enfoques también se utiliza información estadística para obtener los diferentes niveles de resumen.

1.6. Métodos para el cálculo de diferencias.

Los métodos de segmentación de videos se basan en el cálculo de las diferencias entre las características de los fotogramas que los componen. Una característica de video no es más que el parámetro descriptivo que se extrae de una secuencia de imágenes o video. Las características pueden ser utilizadas para interpretar contenidos visuales o como medida de similitud en bases de datos de imagen y video. Las características se pueden describir en tres categorías globales que pueden ser:

Características Estadísticas:

Se extraen de una secuencia de imágenes o de video sin tener en cuenta el contenido. Algunas características pueden ser extraídas directamente de los píxeles de la imagen. Estas características incluyen funciones de análisis, tales como scanges, el flujo de movimiento, y las estructuras de video en el dominio de la imagen (Ziyou , y otros). Más adelante se mostrarán algunos métodos que utilizan las características estadísticas como los métodos basados en píxeles.

Características de dominio comprimido:

Se extraen de una imagen comprimida o secuencia de video sin tener en cuenta el contenido.

Algunos de los archivos multimedia, específicamente las imágenes y el video digital, pueden estar en un formato comprimido. Dado la gran cantidad de archivos de este tipo que se pueden encontrar en el procesamiento de videos, una opción sería descomprimirlos y extraer sus características estadísticas. Haciéndolo de esta forma se pueden tener algunas desventajas como pueden ser:

1. La descompresión implica un gasto extra computacional.
2. El proceso de descompresión y compresión llamado “recodificación”, puede traer como resultado pérdida de las cualidades de las imágenes.
3. Ya que el tamaño de los datos descomprimidos es mucho mayor que el de los de formato comprimido, se necesita mayor capacidad y procesamiento del CPU para codificar y almacenar los datos.

La solución a estos problemas es extraer las características directamente de los datos comprimidos (Ziyou , y otros).

Basadas en contenido:

Son las características que se derivan de la finalidad de describir el contenido real de una secuencia de imágenes o video. Es decir lo que en realidad se puede ver de la imagen.

Ejemplo de esto puede ser la detección de rostros, la detección de objetos o logos en un video determinado.

1.6.1. Método basado en píxel.

Los primeros estudios consideraron el valor de los píxeles como las características principales para el análisis del contenido de los fotogramas, y la

comparación entre dos fotogramas consecutivos, y como la clave para la detección de cambios de tomas abruptos fundamentalmente.

Una medida de la diferencia entre las imágenes sirve como elemento para medir la similitud. Se describen dos métodos fundamentales para la diferencia de imagen: la diferencia absoluta y la diferencia de histograma.

Diferencia absoluta:

La diferencia de dos imágenes se define como la suma de la diferencia absoluta en cada píxel. La primera imagen I_t es analizada con la segunda imagen, I_{t-T} en una distancia temporal T (Ziyou , y otros).

Basados en histogramas.

Otro de los métodos más usados es el basado en histograma. La diferencia entre histogramas es menos sensible al movimiento y además es una medida eficaz para la detección de similitud entre las imágenes. Al detectar cambios significativos en el histograma de color de dos imágenes, se forma una medida más robusta para la correspondencia de la imagen. La diferencia entre histogramas también se puede utilizar en las subregiones para limitar la distorsión que causa el ruido y el movimiento (Ziyou , y otros). Sin embargo, su mayor debilidad se manifiesta cuando dos fotogramas pertenecientes a distintas tomas presentan histogramas similares. En este caso, la transición entre las tomas puede no ser detectada.

1.6.2. Método basado en detección de bordes.

Esta es una técnica que se basa en el cambio de los bordes de las imágenes que se analizan. Para detectar los cambios de tomas se buscan y comparan los bordes que van apareciendo y desapareciendo en la secuencia de fotogramas. Cuando el número de bordes que aparecen y desaparecen de un fotograma a otro es mayor que el umbral que se defina se puede decir que existe un cambio de toma.

1.7. Métodos de clasificación.

Para complementar los métodos de segmentación de video se aplican los algoritmos de clasificación, que en sí, son los que determinan si existe o no un cambio de toma en un lugar determinado del video que se analiza. De aquí la importancia que tiene para la presente investigación el análisis de estos Métodos.

1.7.1. Umbral.

Una vez que mediante los métodos antes expuestos se selecciona la distancia D (diferencia estadística) entre dos fotogramas F_j y F_{j+n} se procede a la selección de un umbral U de clasificación.

Este umbral puede ser estático o dinámico, en función del tipo de material que se analice. En el momento en el que se tiene el umbral en cuestión, se procede a comparar con la distancia entre fotogramas.

- En el caso: $U > D$ se dice que no existe un cambio de toma.
- En cualquier otro caso se dice que si existe un cambio de toma abrupto.

El uso de estos umbrales es muy efectivo. Pero puede traer consigo la detección de falsos negativos cuando se analiza la transacción de una escena estática a una dinámica (Sáes Peña, 2006).

1.7.2. Redes neuronales.

Uno de los métodos más usados para la detección de cambios de toma son las redes neuronales. Una red neuronal es un procesador distribuido masivamente paralelo construido a partir de unidades de procesamiento elementales, denominadas neuronas, que almacenan conocimiento basado en la experiencia y ponen este conocimiento en disposición de ser usado. Recuerda al cerebro en dos aspectos:

1. La red adquiere el conocimiento del entorno a través de un proceso de aprendizaje.
2. La fuerza de las conexiones entre las neuronas, conocida como pesos sinápticos, se emplean para almacenar el conocimiento adquirido.

Las redes neuronales pueden tener varias aplicaciones entre las que más interesan a la presente investigación se pueden citar las siguientes (Sáes Peña, 2006):

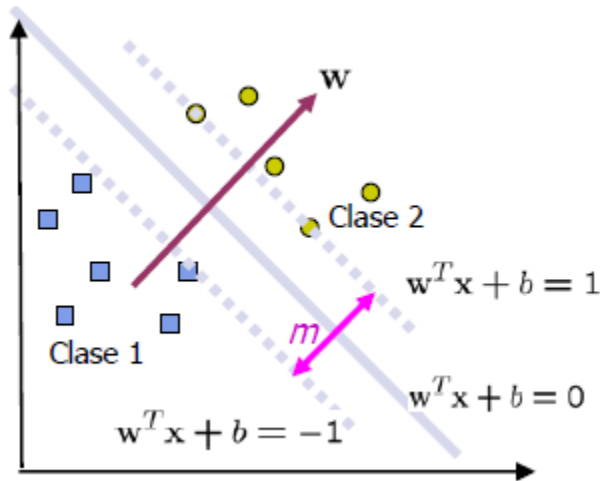
1. Asociación de patrones: memorización de un conjunto de patrones de interés, de forma que sea posible la recuperación posterior de los mismos, bien a partir de una descripción parcial, bien a partir de una descripción completa pero deformada.
2. Reconocimiento de patrones: clasificación de un patrón de interés o una entrada en una clase de un conjunto de categorías predefinidas.
3. Aproximación de funciones: se pretende que la red aproxime una aplicación que existe entre dos conjuntos de datos o espacios y que a priori se desconoce.
4. Construcción de filtros: empleados para la extracción de una característica particular de un conjunto de datos ante la presencia de ruido. En este contexto el ruido puede definirse como las variaciones aleatorias de una o más características de los datos.

1.7.3. Máquinas de soporte vectorial (SVM).

Las Máquinas de Soporte Vectorial (SVM)⁴, es uno de los métodos de clasificación más usados en la actualidad, debido a que han tenido mejor desempeño que algunas de las máquinas de aprendizaje tradicionales como es el caso de las redes neuronales.

⁴ Por su nombre en inglés **S**upport **V**ector **M**achines

Una SVM básicamente, primero mapea los puntos de entrada a un espacio de características de una dimensión mayor y encuentra un hyperplano que los separe y maximice el margen m entre las clases en este espacio (Scientia et Technica, 2005).



Las SVM son básicamente clasificadoras para 2 clases. Se puede cambiar la formulación del algoritmo QP para permitir clasificación multi-clase. Más comúnmente, los datos son divididos “inteligentemente” en dos partes de diferentes formas y una SVM es entrenada para cada forma de división. La clasificación multi-clase es hecha combinando la salida de todos los clasificadores.

Fortalezas

1. El entrenamiento es relativamente fácil.
2. No hay óptimo local, como en las redes neuronales.
3. Se escalan relativamente bien para datos en espacios dimensionales altos.
4. El compromiso entre la complejidad del clasificador y el error puede ser controlado explícitamente.

5. Datos no tradicionales como cadenas de caracteres y árboles pueden ser usados como entrada a la SVM, en vez de vectores de características.

1.7.4. K-means.

El algoritmo de las K-means es otro método de clasificación. Básicamente este algoritmo busca formar clústeres (grupos) los cuales serán representados por K objetos. Cada uno de estos K objetos es el valor medio de los objetos que pertenecen a dicho grupo (Espinoza Villar, 2008).

Inicialmente se seleccionan K objetos del conjunto de entrada. Estos K Objetos serán los centroides iniciales de los K-grupos.

1. Se calculan las distancias euclidianas de los objetos (datos) a cada uno de los centroides. Los datos se asignan a aquellos grupos cuya distancia es mínima con respecto a todos los centroides.
2. Se actualizan los centroides como el valor medio de todos los objetos asignados a ese grupo.
3. Se repite el paso 2 y 3 hasta que se satisface algún criterio de convergencia.

1.8. 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 duda, la buena selección de las tecnologías.

1.8.1. Librerías.

Dentro de las librerías para el trabajo con videos o visión por computadora como se le conoce se pueden encontrar LTI, VXL y Opencv.

LTI-Lib es una librería orientada a objetos con algoritmos y estructuras de datos utilizadas en el procesamiento de imágenes. Se ha desarrollado en la Cátedra de

Técnica Informática (Lehrstuhl fuer Technische Informatik) LTI en la Universidad de Tecnología de Aachen, en el marco de muchos proyectos de investigación en visión por computador y en la robótica en el reconocimiento de objetos (ltilib.sourceforge.net).

LTI-Lib para Windows, necesita al menos el MS Visual C++ 6.0 con Service Pack 5.0 o el MS Visual C++. NET 2003. Por tanto, no se cumple con la condición de implementar un componente multiplataforma con el uso de herramientas totalmente libres.

VXL (Vision-something-Libraries) es una colección de librerías C++ diseñado para la visión por computadora y la investigación informática. VXL está escrito en ANSI / ISO C++ y está diseñado para ser portátil en muchas plataformas (vxl.sourceforge.net). Las bibliotecas del núcleo de VXL son:

- Contenedores y los algoritmos numéricos: (VNL cifras) por ejemplo, matrices, vectores, descomposiciones, optimizadores.
- Vil (imágenes): cargar, guardar y manipular imágenes en muchos formatos de archivo comunes, incluyendo imágenes muy grandes.
- VGL (geometría): Geometría de puntos, curvas y otros objetos elementales, 2 o 3 dimensiones.
- VSL (streaming de E / S), VBL (plantillas básicas), vulnerabilidad (servicios públicos): independiente de la plataforma la funcionalidad de varios.

Como principal desventaja de VXL se puede señalar la falta de algoritmos de clasificación y algoritmo de detección de contornos, que es uno de los que se propone para la confección del componente. Por tales razones no se recomienda el uso de esta librería.

Opencv (Open Source Computer Vision) es una librería de código abierto para la programación de funciones de visión por computadora en tiempo real. Es realizada bajo la licencia BSD, esto quiere decir que es libre tanto para su uso académico como comercial. Está desarrollada en C y C++ y corre bajo Linux, Windows y Mac.

Opencv fue diseñado para la eficiencia computacional y con un fuerte enfoque a aplicaciones en tiempo real. Opencv está escrito en C optimizado y se puede aprovechar el procesamiento de múltiples núcleos.

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 funciones gráficas y de interfaz de usuario.

Uno de los objetivos de Opencv es proporcionar una forma simple de utilizar la infraestructura de la visión por computadora, y ayudar a construir aplicaciones de visión bastante sofisticados rápidamente.

La librería Opencv contiene más de 500 funciones que abarcan muchas áreas en la visión, incluyendo la inspección de productos, imágenes médicas, la seguridad, interfaz de usuario, calibración de la cámara y la robótica.

Opencv también contiene una descripción completa, del uso general de librerías de máquinas de aprendizaje (MLL). Esta es una sub-librería que se centra en el reconocimiento estadístico de patrones. La MLL es de gran utilidad para las tareas de la visión que se encuentran en el núcleo de Opencv, pero es lo suficientemente

general como para ser utilizado para cualquier problema de aprendizaje automático (Kaehler, y otros, 2008).

Analizando las funcionalidades que brinda la librería se recomienda el uso de la misma para el desarrollo del componente, debido a que se le puede dar solución a todas las necesidades descritas anteriormente.

1.8.2. Lenguaje de programación.

Las computadoras necesitan de un lenguaje propio para poder interpretar las instrucciones que se les dan, en la interacción hombre-computadora. Ese lenguaje que permite esta relación de los usuarios con las computadoras es el lenguaje de programación, de los que se pueden citar varios ejemplos (Basic, C, C++, C#, Java, JavaScript, PHP, etc.).

Un lenguaje de programación está conformado por una serie de reglas sintácticas y semánticas que serán utilizadas por el programador y a través de las cuales creará un programa o subprograma. Por otra parte, las instrucciones que forman dicho programa son conocidas como código fuente (Lanzillotta, 2008).

Como todos los lenguajes, los lenguajes de programación poseen también sus propias clasificaciones. Por ejemplo según su nivel de abstracción se pueden clasificar en tres clases:

- lenguaje de bajo nivel: es el código fuente de la máquina, es decir el que la máquina puede interpretar.
- lenguaje de nivel medio: un término entre el lenguaje de la máquina y el lenguaje natural.
- lenguaje de alto nivel: los que están compuestos por elementos del lenguaje natural, es decir el humano, especialmente el inglés.

Se pueden clasificar también atendiendo la forma en que se ejecutan, donde se pueden encontrar:

- lenguajes compilados: programas que permiten traducir un programa del lenguaje natural al lenguaje de bajo nivel.
- lenguajes interpretados: los que sólo hacen la traducción de los datos que se van a utilizar en ese momento y no los guarda para usar posteriormente.

A continuación se realiza un análisis comparativo de dos lenguajes de programación muy usados en la comunidad de programadores a nivel mundial, por las funcionalidades que brindan, estos son C# y C++.

Visual C#.

Visual C# es un lenguaje de programación diseñado por Microsoft para crear una amplia gama de aplicaciones que se ejecutan en .NET Framework⁵. C# es simple, eficaz, con seguridad de tipos y orientado a objetos. Con sus diversas innovaciones, C# permite desarrollar aplicaciones rápidamente y mantiene la expresividad y elegancia de los lenguajes de tipo C.

Cuando este lenguaje es compilado, el compilador traduce el código fuente a lenguaje intermedio de Microsoft (MSIL), un conjunto de instrucciones independiente de la CPU que puede convertirse a código nativo de manera eficiente (Microsoft, 2005).

C++.

C++ es un lenguaje de programación diseñado por Bjarne Stroustrup a mediados de la década de 1980. 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.

⁵ .NET Framework es un entorno multilenguaje para generar, implementar y ejecutar servicios Web XML y aplicaciones Web (Microsoft, 2005).

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. C++ 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 C++ justificarán de sobra el esfuerzo realizado (Schildt, 04/2001).

Después de un breve recorrido por las principales características de estos lenguajes de programación, teniendo en cuenta que se propone la librería Opencv para el desarrollo y debido a que se desea que el componente se integre con el Sistema de Captura y catalogación de Medias el cual está implementado en C++, se propone que el lenguaje óptimo para el desarrollo del componente es el C++.

1.8.3. Entorno de desarrollo (IDE).

Qt Creator.

Qt Creator 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 siguientes prestaciones:

- 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. (Nokia, 2008).

Por estas razones se recomienda el uso de Qt Creator como IDE para el desarrollo del componente debido a que cumple con todo lo necesario para un desarrollo ágil y óptimo.

1.9. Conclusiones parciales.

En el presente capítulo se cumplieron los objetivos esperados, dando cumplimiento a las tareas que complementaban el mismo. Se pudo apreciar que existen varios métodos para la separación automática de segmentos de video, así como algoritmos para su clasificación. Por tales razones se puede concluir:

1. El algoritmo debe estar enfocado a realizar un resumen de video del tipo Storyboard, para satisfacer las necesidades del SCCM.
2. De los métodos para el cálculo de diferencia entre fotogramas analizados, no existen argumentos suficientes para seleccionar uno en específico. Por lo que se propone implementar todos estos métodos y probar cual de todos se adapta mejor a las necesidades reales del proyecto SCCM.
3. Los métodos de clasificación estudiados pueden ser de una gran ayuda en el momento de determinar en qué posición existe un cambio de toma, de todos el que mejor se adapta a las técnicas de resumen de video es el de umbral por su sencillez y adecuada efectividad el cual se complementa con los agrupamientos de alto nivel para alcanzar elevados índices de identificación.

CAPÍTULO 2: Descripción y diseño de algoritmos y componente para la separación automática de segmentos de video.

En el presente capítulo se realiza la propuesta de un algoritmo para dar solución al problema que dio origen a la presente investigación. Se plantean con argumentos sólidos los diferentes pasos por los que esta constituido el algoritmo de segmentación temporal.

2.1. Selección de requisitos funcionales del componente de software.

Para dar solución al problema planteado, primeramente se debe conocer los requisitos funcionales con los que debe cumplir el componente de software que se proponga, los cuales fueron detectados a lo largo de la investigación.

1. Detectar cambios de tomas en videos.

El componente propuesto debe ser capaz de dada la dirección de un video detectar los cambios de tomas existentes en el mismo.

2. Escoger los fotogramas representativos dentro de los cambios de tomas.

El componente propuesto debe ser capaz de reconocer los fotogramas representativos dentro de las tomas de un video, así como agruparlos por el factor de similitud que posean entre ellos, para reducir el número de los mismos.

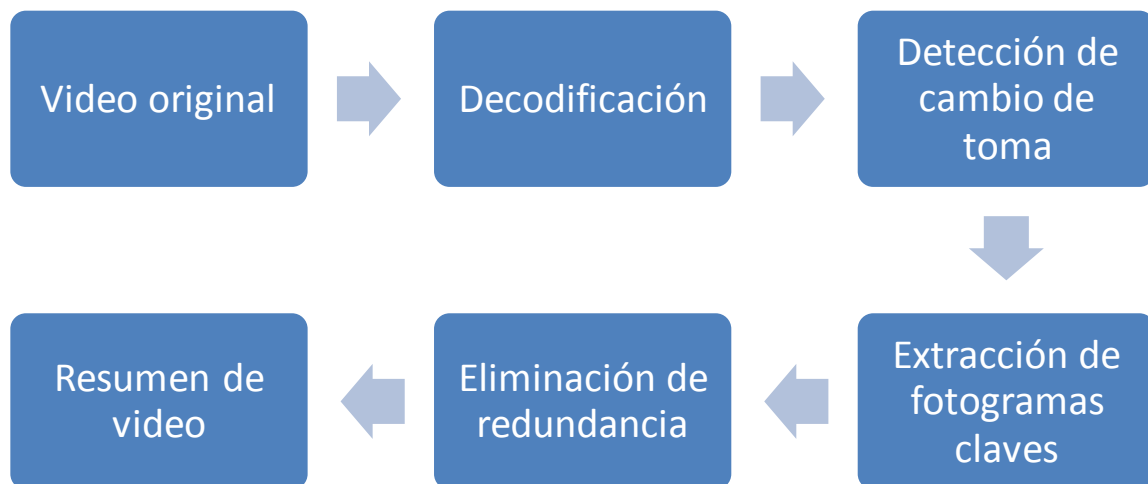
3. Exportar como imagen JPEG fotogramas representativos.

El componente propuesto debe ser capaz de exportar los fotogramas representativos de un video a una dirección local dada. Cumpliendo con lo siguiente:

- Las imágenes se guardarán como (JPEG).
- El nombre de las imágenes coincidirá con la posición en tiempo que poseen dentro del video, cumpliendo con el formato “hh_mm_ss”.

2.2. Descripción general de la propuesta de algoritmo

El diseño del algoritmo se hace siguiendo el orden en que se identifican los requisitos funcionales. El siguiente gráfico servirá de guía para un entendimiento general de la propuesta.



Como se muestra en la figura a partir de un video original se extraen los fotogramas del mismo los cuales son la entrada al método de detección de cambio de toma. Para el proceso de decodificación se utilizan los métodos que posee OpenCV para el acceso a los fotogramas del video y los cuales tienen como base la biblioteca ffmpeg. Luego de la decodificación del video se ejecutan los métodos de detección de cambio de toma los cuales se dividen en dos partes fundamentales. En primer lugar la función de distancia que determina la similitud entre fotogramas consecutivos y luego el proceso de detección de existencia o no de cambio de toma. El método de detección de cambio de toma, luego de procesar todos los fotogramas del video, se obtiene como salida una lista de índices donde se encuentran los cambios de tomas del video. A partir de estos índices se seleccionan aquellos fotogramas que caracterizan a cada toma o lo que se denomina extracción de fotogramas claves. El siguiente paso en el algoritmo permite hacer un agrupamiento de alto nivel para unificar aquellas tomas que son cercanas en el tiempo y contienen coherencia en la información semántica que poseen estos fotogramas claves. Por último a partir de los fotogramas claves de cada agrupamiento se pasa a la construcción del resumen del video. En las siguientes secciones se describen cada uno de los pasos del algoritmo propuesto.

2.3. Decodificación del video

Para la decodificación del video se utilizan los métodos de lectura de video que posee OpenCV. Para ello es necesario crear una estructura de OpenCV de tipo CvCapture y con la función CVQueryFrame se accede de manera consecutiva a los fotogramas que componen el video. Como en el método de detección de cambio de toma es necesaria la información de dos fotogramas consecutivos, es necesario en la decodificación extraer los dos fotogramas. 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 consecutividad 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 cambios de tomas.

El siguiente segmento de código ilustra la manera en que se acceden a los fotogramas del video. Las variables frame1 y frame2 son los fotogramas consecutivos que se extraen mientras que la variable step controla el salto entre fotogramas consecutivos.

```
for(int c=0;;)
{
    for(int j=1;j<step;j++)
        cvQueryFrame(capture);
    frame2 = cvQueryFrame(capture);
    if(frame2 == NULL)
        break;
    if (!is)
    {
```

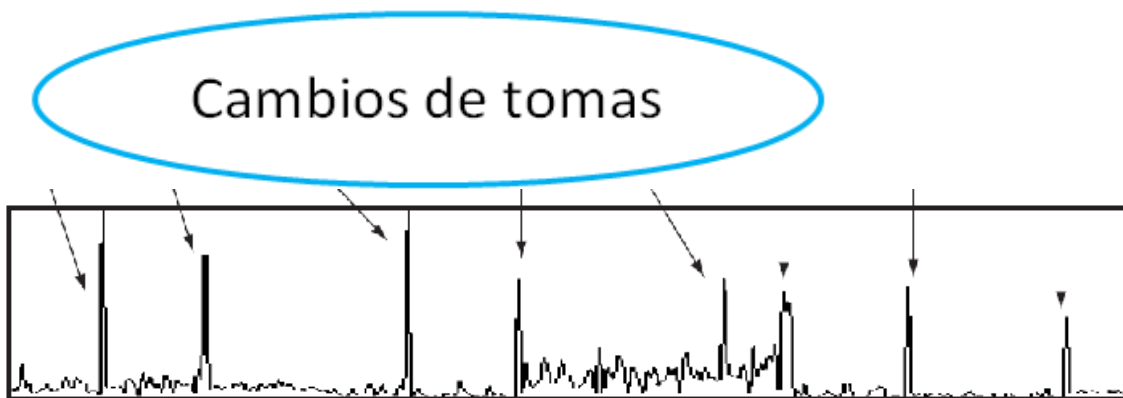
```

double d1 = calc->Distancia_hist(frame1 ,frame2);
punto.x=d1;
L_dist->append(punto);
c+=step;
}
is=0;
cvReleaseImage(&frame1);
frame1 = cvCloneImage(frame2);
}

```

2.4. Detección de cambios de tomas.

El problema de la detección de cambios de tomas se modela a partir de un vector de distancia entre fotogramas consecutivos de un video en el cual aparecen las fronteras de las tomas como picos pronunciados en dicha función. La figura muestra la representación de un vector de distancia con la información visual asociada donde se reflejan los cambios visuales representados por un pico en la función de distancia.



Función de distancia

Como se describe en el capítulo 1 haciendo referencia al trabajo de (Alan, y otros, 2010) (Herranz, y otros, 2010) donde se hace un análisis de los principales enfoques expuestos en la competencia TRECVID 2005 los métodos de distancias

más utilizados por los principales grupos de investigación son los métodos basados en el histograma de color, estadísticas, bloque de píxel y características por lo que en la propuesta se implementan cada uno de estos métodos y se hace un análisis según la estrategia de cual de dichos métodos arroja mejores resultados.

2.4.1. Cálculo de distancia por bloque de píxeles.

Dentro de los métodos que se proponen para usar en el algoritmo, como parte del cálculo de distancia, se encuentra la distancia euclidiana entre bloque de píxeles de modo que la diferencia entre dos imágenes se determina según las expresiones (2.1) y (2.2) (Ziyou , y otros).

El método basado en bloque de píxel parte de dividir de manera homogénea cada fotograma en K bloques de resolución (m,n), la distancia entre cada bloque de los fotogramas consecutivos se determina por la expresión (2.1) donde $f_k^a(i, j)$ es el píxel en la posición (i,j) del bloque k del fotograma f^a .

$$d_k = \sum_{i=0}^m \sum_{j=0}^n \left(f_k^a(i, j) - f_k^b(i, j) \right)^2 \quad (2.1)$$

A partir de la distancia entre cada bloque se hace un análisis local de aquellos bloques que poseen una distancia superior a la media de las distancias de los bloques, los cuales no se tienen en cuenta en la expresión (2.2), este análisis local nos permite minimizar las falsas detecciones ante pequeños movimientos que pueden aparecer en el contenido de la escena, o simplemente problemas propios en la codificación del video, que pueden introducir pequeños cambios visuales en algún lugar de la imagen aunque estos no sean perceptibles. Luego de esta discriminación se determina la suma de todos los bloques cuyo valor esté por debajo de la media según la expresión (2.2)

$$d = \sum_{k=1}^K d_k \quad (2.2)$$

Con relación a la implementación se creó un método que recibe como entrada dos fotogramas y devuelve la distancia utilizando el criterio antes mencionado.

Pasos

1. Se divide la imagen en bloques de manera homogénea.
2. Se calcula para cada bloque la distancia según la expresión (2.1) por cada canal de color de la imagen y determina la media de cada bloque.
3. Se determina la media de todas las distancia entre los bloques
4. Se discriminan los bloques cuya distancia esté por encima de la media.
5. Se determina la distancia entre los fotogramas según la expresión 2.2.

2.4.2. Cálculo de distancia por histograma de color.

Los métodos basados en histogramas de color están reportados por múltiples autores como los más eficientes a la hora de modelar el problema de la detección de cambios de tomas. Desde el punto de vista teórico el histograma de color de una imagen es una característica global de la imagen que identifica la distribución de colores en la misma. Por ser una medida global de la distribución de color resulta muy difícil que esta cambie de manera significativa en las imágenes que componen una misma toma, pues de ser así la calidad visual de esta toma se puede ver afectada. No obstante este método es sensible a cambios de iluminación y movimientos bruscos de los contenidos de la toma aunque su efectividad es muy alta en la detección de cambios de tomas. La construcción del vector histograma de color de una imagen cuenta con 3 vectores de 256 valores para el caso de una imagen en colores de 8 bit. Para la determinación de la distancia entre histograma de colores existen tres métodos que se han impuesto: intercepción de histograma, método de χ^2 y correlación de histograma.

En la propuesta presentada se utilizó el método de intercepción de histograma sobre los 3 canales del espacio de color HSV. Se propone este espacio de color

ya que a diferencia del RGB, este es un espacio de color perceptualmente uniforme. La siguiente expresión muestra la forma de determinar la distancia basada en este enfoque y aplicado sobre cada canal. El valor de la distancia final se determina como la media de las distancias de todos los canales.

$$d_c = \sum_{k=0}^K \min(H_c^a(k), H_c^b(k)) \quad (2.3)$$

Para la implementación de este enfoque basado en la intercepción de histograma se desarrolló un método que recibe como entrada dos fotogramas de un video y devuelve la distancia entre estos. A continuación se describen de manera general los pasos seguidos en la implementación.

Pasos:

1. Convierte cada fotograma del espacio de color RGB a HSV con la función de OpenCV `CvCvtColor`.
2. Calcula el histograma de cada canal para cada fotograma.
3. Determina el valor de distancia usando el criterio de intercepción de histograma.

2.4.3. Cálculo de distancia a partir de las características (bordes).

Los métodos basados en características o bordes como también se conocen se utilizan, en general, para compensar los movimientos de los objetos que componen las escenas y los cambios de iluminación. Por esta razón su uso en métodos híbridos, sobre todo de conjunto con métodos de histograma, es bastante utilizado. Los bordes en una imagen se determinan a partir de la discretización de funciones diferenciables de segundo orden. Con ello existen diversos filtros que son capaces de determinar los bordes en una imagen. Uno de los más conocidos son los filtros Sobel y Canny. El principio de funcionamiento de estos métodos es bastante similar, en todos los casos se convoluciona la matriz de la imagen con un kernel que tiene en cuenta para cada píxel de una imagen los valores de la

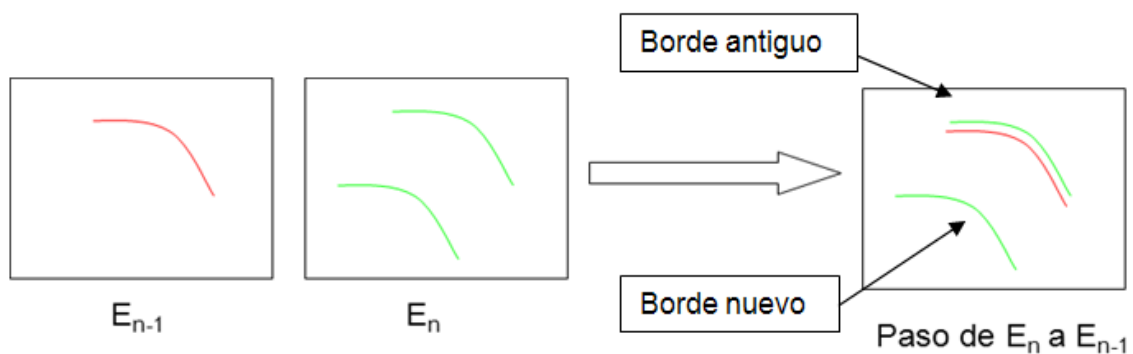
vecindad de este píxel. La diferencia esencial en estos métodos es el kernel que utilizan; que unos le dan más pesos a los vecinos diagonales y otros a los horizontales y verticales o cualquier combinación de estos.

La detección de cambios de tomas utilizando el criterio de bordes se basa en conocer los bordes que aparecen y desaparecen de un fotograma a otro. Si este cambio es significativamente grande significa que los objetos han cambiado y por tanto se ha pasado de una toma a otra (ver fig. 4).

La propuesta que se utiliza tiene en cuenta además pequeños movimientos que pueden aparecer entre objetos de una misma imagen por lo que los bordes no solo tienen que estar en la misma posición sino que pueden aparecer en el nuevo fotograma en una vecindad de su posición anterior.

Para resolver este problema se dilatan los bordes de la imagen destino y se determinan los bordes que se mantienen en la región dilatada. Para la dilatación se utiliza una estructura circular de 5X5 con lo cual se garantiza ser estable ante movimientos leves de los objetos de la toma.

Por último se cuentan los bordes que aparecen y los que desaparecen y se aplica una expresión para calcular la distancia.



De manera general se definen los siguientes pasos en la implementación del método de distancia basado en bordes.

Pasos:

1. Se extraen los bordes de cada fotograma utilizando el método de Canny que proporciona OpenCV.
2. Se invierte cada imagen y se asignan a variables temporales para luego comparar con las imágenes de los bordes.
3. Se dilatan las imágenes invertidas usando una estructura morfológica de 5X5.
4. Se determinan que bordes del primer fotograma aparecen en la imagen dilatada del fotograma 2 y lo mismo entre los bordes de la imagen 2 con la imagen 1 dilatada.
5. Se aplica la expresión para el cálculo de distancia.

2.4.4. Cálculo de distancia a partir de una medida estadística LR1.

La construcción de medidas estadísticas de segundo grado resulta frecuente para abordar el problema de la detección de cambios de tomas. Para ello se utilizan las combinaciones de las varianzas y del cuadrado de la media de los valores de píxel de dos imágenes consecutivas. La expresión (2.5) permite determinar la distancia entre dos fotogramas utilizando una medida estadística denominada LR1.

$$d_{LR1} = 1 - \min\left(LR1, \frac{1}{LR1}\right) \quad (2.5)$$

Donde:

$$LR1 = \frac{\left[\frac{s_a^2 + s_b^2}{2} + \left(\frac{m_a - m_b}{2}\right)^2\right]^2}{s_a^2 * s_b^2} \quad (2.6)$$

Se propone utilizar esta medida para determinar la distancia entre fotogramas con un enfoque estadístico. Los parámetros s_a y m_a son la varianza y la media respectivamente de los píxeles de uno de los fotogramas.

2.5. Ajuste de la función de distancia.

Antes de aplicar el proceso de detección de cambios de tomas se propone utilizar un método local de ajuste de la distancia de los fotogramas del video. Esta función de ajuste se propone el uso de una ventana deslizante de tamaño $2w+1$. Para realizar el ajuste por cada valor de la lista de distancias se toman los valores

de distancia de los w vecinos hacia la izquierda y hacia la derecha del valor que se analiza. Se determina la media de todos estos valores para finalmente calcular el valor absoluto de la distancia que se analiza restado a la media de los valores de la ventana.

La siguiente expresión describe de manera formal la estrategia antes mencionada.

$$d_1(i) = \left| d(i) - \frac{1}{2w + 1} \sum_{k=i-w}^{k=i+w} d(k) \right|$$

Con esta estrategia se permite reducir el ruido que pueda aparecer en algún segmento de la señal de distancia cuyos valores pueden estar afectados por algún movimiento en el contenido del video y que realmente no estamos en presencia de un cambio de toma.

2.6. Detección de cambios de tomas.

Los enfoques de clasificación de cambios de tomas son muy diversos como se expone en el capítulo 1. Existen trabajos [(Ziyou , y otros)] sobre resumen de video que la detección de cambios de tomas se ataca desde métodos simples tales como estrategias de umbrales automáticos. Esto es algo que se encuentra muy relacionado con el hecho de que en resumen de video no se requiere de gran exactitud en los métodos de detección de cambios de tomas, pues en todos los trabajos existe un enfoque de agrupamiento de alto nivel que hace que la estructura inicial de separación en tomas se pierda pasando a una estructura de escenas. Por tal motivo se propone emplear un método de umbral global automático.

Este método de umbral se aplica sobre la función de distancia ajustada. El cálculo automático del umbral se basa en determinar la media (m) y la varianza (s) de todos los valores de la función de distancia ajustada y con ello construir un umbral $T = m + s$ que nos permita determinar aquellos valores que sean superiores a este umbral como posibles cambios de tomas.

En el enfoque se habla de posibles cambios de tomas pues luego de aplicar el umbral se hace un pos-procesamiento que se basa en el principio de que no

existen dos cambios de tomas consecutivos cuya diferencia de tiempo sea inferior a un valor predeterminado. Por lo que se recorren todos los posibles cambios de tomas se define una ventana de tiempo fija y se verifica que no existan dos o más tomas consecutivas dentro de esta ventana. De cumplirse esta condición se selecciona como cambio de toma al mayor de todos los posibles cambios de tomas dentro de la ventana y de no cumplirse la condición simplemente se etiqueta como cambio de toma.

2.7. Selección de fotogramas claves.

Los métodos de detección de fotogramas permiten a partir de una imagen fija representar el contenido de una toma. Para ello se utilizan con frecuencias métodos simples que emplean el centro de la toma o los extremos de la toma como fotograma clave. Esto a pesar de ser simple no se considera para nada inefectivo pues dentro de una misma toma los fotogramas que la componen no presentan diferencias significativas. El método de selección de fotogramas claves propuesto utiliza el centro de la toma.

2.8. Agrupamiento de alto nivel

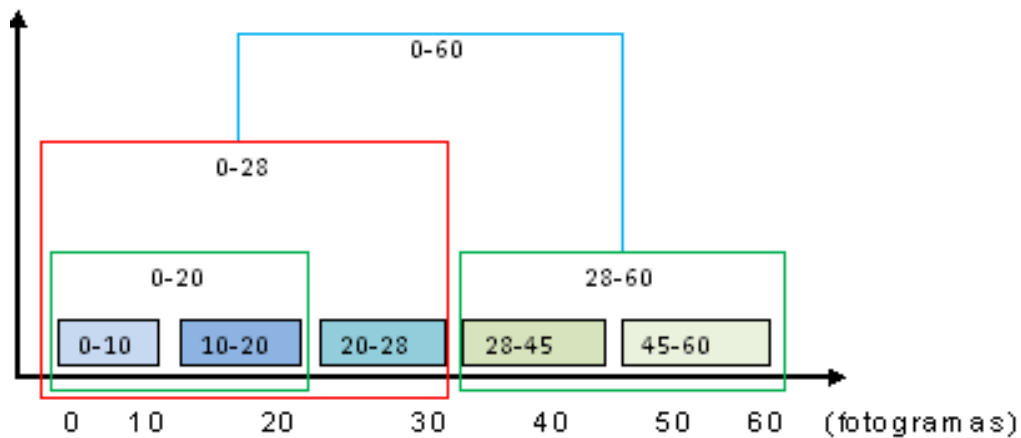
Para la selección de los fotogramas representativos se propone emplear un método de agrupamiento jerárquico basado en el criterio de la distancia mínima entre los fotogramas claves.

Los pasos que se siguen para el agrupamiento son fundamentalmente los siguientes:

1. Calcular la distancia entre los fotogramas claves consecutivos de un video.
2. Agrupar el par de fotogramas claves de menor distancia en un clúster.
3. Crear un nuevo clúster y asignarle como sus hijos los clústeres seleccionados en el paso 2 recalculando la distancia con el nuevo clúster a partir de los valores medios de las distancias anteriores.
4. Poner el nuevo clúster dentro del listado general y eliminar sus hijos de este listado.

5. Repetir desde el paso 2 hasta que se tenga un solo clúster padre de todos.

Luego de concluir estos pasos, se contará con un clúster que dependiendo el nivel de profundidad que se solicite, será capaz de dar los fotogramas representativos del video. En la imagen que se muestra a continuación se puede observar como queda el clúster de agrupamiento.



2.9. Exportación de fotogramas como imágenes.

Al concluir el proceso de extracción de fotogramas representativos, estos se deben guardar en una dirección local, estos fotogramas están identificados por su posición dentro del video por lo que se debe calcular el tiempo exacto en el que se encuentran en milisegundos, para poder exportarlo en el formato "hh_mm_ss".

$$MSC = \frac{1000 * PF}{Fps}$$

En la fórmula anterior se calcula el milisegundo (*MSC*) donde se encuentra un fotograma, teniendo la posición del fotograma (*PF*) y la cantidad de fotogramas por segundo con que se debe reproducir el video.

Después de tener los milisegundos exactos en que se encuentran los fotogramas representativos, se debe salvar la imagen haciendo una conversión de tiempos, es decir de milisegundos a horas, minutos y segundos, para darle origen al nombre de la imagen.

2.10. Integración.

La implementación del presente componente tiene como objetivo su futura integración con el producto SCCM, de aquí la necesidad de definir un conjunto de parámetros de entradas y salidas para su correcto funcionamiento e integración.

Como parámetros de entrada se definieron los siguientes:

- Id del proceso que se realiza: Esto representa el id del proceso que se ejecuta en la plataforma de codificación en ese momento, con el objetivo de tener un control estricto de los mismos.
- Dirección de entrada: Es la dirección física de la media a la que se le desea extraer los fotogramas claves.
- Dirección de salida: Lugar de la PC en que se desean salvar los fotogramas claves.
- Sensibilidad: Representa el porcentaje de resumen que se desea obtener luego de aplicar el proceso de segmentación al video.

Como parámetro de salida se definió el siguiente:

- Porcentaje de ejecución en que se encuentra el proceso. El componente debe informar a la plataforma en cada momento en qué porcentaje del proceso de resumen se encuentra.

2.11. Conclusiones parciales.

En el presente capítulo tras la selección de un grupo de requisitos funcionales y describir paso a paso un componente para la separación automática de segmentos de videos, con el objetivo de realizar un resumen del tipo Storyboard, se puede decir que se está en presencia de un algoritmo diseñado para cumplir con la necesidad de resumir videos, tanto en el producto SCCM como en otro que requiera esta funcionalidad. El mismo está compuesto por los métodos que han arrojado mejores resultados en la segmentación temporal de videos y además los

más usados a nivel internacional. Para confirmar que el componente realmente cumple con las expectativas requeridas, solo se necesita llevar a cabo un grupo de pruebas que validen la propuesta realizada.

CAPÍTULO 3: Validación del componente propuesto.

En el presente capítulo se documentan las diferentes pruebas que se le realizan al componente para validar su correcto funcionamiento. Primeramente se realizan pruebas para comprobar los formatos de videos que soporta. También se le realizarán pruebas para analizar las detecciones de cambios de tomas.

3.1. Pruebas de soporte de formato.

Las pruebas de soporte de formato son un pequeño test al que es sometido el algoritmo, el cual tiene el objetivo de analizar y documentar cuales de los formatos de videos, son soportados por el mismo.

Para realizar estas pruebas se escogieron un total de 4 formatos de videos, entre los que se encuentren los empleados en el producto SCCM. Como resultado estas pruebas arrojaron los siguientes resultados:

Nombre del video:	72060fps.mov
Códec:	DV Video (dvhp)
Resolución:	960x720
Fotogramas por segundo:	59.9400
Valoración:	procesamiento total del video sin inconvenientes.

Nombre del video:	1080 30fps.mov
Códec:	DV Video (dvh6)
Resolución:	1280x1080
Fotogramas por segundo:	29.9700

Valoración: procesamiento total del video sin inconvenientes.

Nombre del video: drama.mpg

Códec: MPEG-1/2 Video(mpgv)

Resolución: 352x288

Fotogramas por segundo: 25

Valoración: procesamiento total del video sin inconvenientes.

Nombre del video: MicrosoftSustainability.avi

Códec: MPEG-4 video (XVID)

Resolución: 480x270

Fotogramas por segundo: 29.970029

Valoración: procesamiento total del video sin inconvenientes.

Nombre del video: google.avi

Códec: H264

Resolución: 176x144

Fotogramas por segundo: 29.970029

Valoración: procesamiento total del video sin inconvenientes.

Nombre del video:	sample.ogg
Códec:	Theora Video (Theo)
Resolución:	320x240
Fotogramas por segundo:	29.970029

Valoración: procesamiento total del video sin inconvenientes.

Luego de las pruebas realizadas, se puede plantear que de una muestra de 6 videos, con diferentes formatos, los seis fueron soportados por el componente y se logró extraer de forma satisfactoria los fotogramas representativos, lo que representa el 100% de la muestra utilizada.

3.2. Pruebas a algoritmos de detección de cambios de tomas.

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 de los algoritmos de detección de cambios de tomas consiste en calcular su *recall* (R) y su *precisión* (P). Para probar los algoritmos 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 a partir de las corridas de los métodos utilizados en el componente contra la base de datos estadística utilizada en el TREC Vid de 2005. Para su documentación se tuvo en cuenta: nombre de la media que se analiza, cantidad de fotogramas, cantidad

de cambios abruptos, falsos positivos, falsos negativos, recall y precisión de los algoritmos.

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

Bloque de Píxel

	# fotogramas	abruptos	GT	Fp	Fn	Recall	Precisión
News11	42828	252	249	42	3	98,81	85,57
News12	27400	97	97	7	0	100,00	93,27
movie	23820	116	108	14	4	96,43	88,52
Drama	23390	141	141	2	0	100,00	98,60

LR1

	# fotogramas	abruptos	GT	Fp	Fn	Recall	Precisión
News11	42828	252	248	45	4	98,41	84,64
News12	27400	97	97	7	0	100,00	93,27
movie	23820	116	111	8	1	99,11	93,28
Drama	23390	141	141	21	0	100,00	87,04

Correlación de Histogramas (HSV)

	# fotogramas	abruptos	GT	Fp	Fn	Recall	Precisión
News11	42828	252	231	60	6	97,47	79,38

News12	27400	97	93	9	2	97,89	91,18
movie	23820	116	110	12	2	98,21	90,16
Drama	23390	141	141	39	0	100,00	78,33

Analizando los resultados anteriores se puede observar que el algoritmo funciona con un recall por encima de 97, esto se debe a que existe un ínfimo número de falsos negativos, siempre menor que 7.

En el caso de la precisión del algoritmo con números superiores a 79, si se analiza de forma general, es menor que su recall, esto se debe a su alta sensibilidad lo que proporciona la detección de un número mayor de falsos positivos. Esto no es un problema preocupante ya que con la estrategia de agrupamiento mediante un clúster jerárquico se elimina un gran número de redundancias existentes en el resumen.

3.3. Conclusiones parciales.

Tras las pruebas realizadas se puede asegurar que el componente cumple con las necesidades de resumen del producto SCCM. Siendo capaz de realizar resumen Storyboard a videos OGG y alta definición, que son los dos formatos con los que trabaja el SCCM. Se puede asegurar además que se está en presencia de un componente confiable, ya que se tiene una detección valida en el 97% de los casos y se logra eliminar la redundancia con el uso del agrupamiento jerárquico.

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 mejorar la información visual en la representación sintética de videos en el Sistema de Captura y Catalogación de Medias, con la obtención de un mínimo de falsas detecciones.

El uso de un agrupamiento jerárquico garantiza un componente con una alta escalabilidad, capaz de extraer varios resúmenes de un mismo video a petición del usuario.

Con el uso del framework 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 realizar resúmenes de videos representados mediante Storyboard.

Recomendaciones.

- Continuar el desarrollo de esta investigación con el fin de incorporar elementos que permitan la detección de cambios de tomas graduales en videos.
- Explotar la información semántica de los videos para realizar los resúmenes visuales.

Bibliografía y referencias bibliográficas.

A Survey of Technologies for Parsing. *Journal of Visual Communication and Image*.
Ahanger, G. y Little, T.D.C. 1996. 1996.

Alan F. Smeaton, Over, Paul and Doherty, Aiden R. 2010. *Video shot boundary detection: Seven years of TRECVID activity*. s.l. : Computer Vision and Image Understanding, 2010.

Benini, S., et al. 2006. *Extraction of significant video summaries by dendrogram analysis*. s.l. : IEEE International, 2006.

Brunelli, R.; Mich, O. y Modena, C.M. 1999. "A Survey on the Automatic Indexing of Video Data" *Journal of Visual Communication and Image Representation*. 1999. pp. 78-112.

Cano, Jesús Bescos. 2001. *Segmentación Temporal De Secuencias De Videos*. Maderid : Universidad Politécnica De Madrid, 2001.

Chen, Y., et al. 2010. *A temporal video segmentation and summary generation method based on shots' abrupt and gradual transition boundary detecting*. s.l. : Communication Software and Networks, International Conference on, 2010.

Coll, D.C. y Choma, G.K. 1976. *Image Activity Characteristics in Broadcast Television*. s.l. : IEEE Transactions on Communications, 1976. pp. 1201-1206.

Comparison of video shot. **Rowe, J. S. Boreczsky and L. A. Jan. 1996.** Jan. 1996, Vol. 2670.

Espinoza Villar, Raúl. 2008. *DETERMINACIÓN DE LOA SÓLIDOS EN SUSPENSIÓN EN LOS RÍOS DE LA AMAZONIA PERUANA MEDIANTE IMÁGENESSATELITALES MODIS*. Lima : Universidad Nacional Agraria La Molina, 2008.

Gong, Y. and Liu, X. 2000. *Generating optimal video summaries*. s.l. : in Multimedia and Expo 2000. ICME ,IEEE International Conference on, vol. 3, 2000.

Hernandez León, Rolando Alfredo and Coello Gonzalez, Sayda. 2011. *El Proceso de Investigación Científica* . Habana, Cuba : s.n., 2011. 978-959-16-1307-3.

Herranz, L. and J. Mart. 2010. *A framework for scalable summarization of video*. IEEE Transactions. s.l. : Circuits and Systems for Video Technology, 2010.

Hrenández Peñalver, Gregorio. 2008. dma.fi.upm.es. *Universidad Politécnica de Madrid*. [Online] Departamento de Matemáticas Aplicadas., 2008. [Cited: febrero 11, 2011.] <http://www.dma.fi.upm.es/gregorio/JavaGC/Cconvexo/>.

Idris, F. y Panchanathan, S. 1997. *Review of Image and Video Indexing Techniques.* s.l. : Journal of Visual Communication and Image Representation, 1997. pp. 146-166.

Jesús Bescós. APRIL 2004. *Real-Time Shot Change Detection OverOnline MPEG-2 Video.* s.l. : IEEE, APRIL 2004.

Jesús Bescós, Guillermo Cisneros, José M. Martínez, José M. Menéndez, and Julián Cabrera. APRIL 2005. *IEEE TRANSACTIONS ON MULTIMEDIA, A Unified Model for Techniques on Video-Shot Transition Detection.* APRIL 2005.

Kaehler, Adrian and Bradski, Gary. 2008. Learning OpenCV. [book auth.] Adrian Kaehler Gary Bradski. *Learning OpenCV.* s.l. : O'Reilly Media, 2008.

Lanzillotta, Analía. 2008. mastermagazine.info. *Definición de Leguajes de Programación.* [Online] Master Magazine, 2008. [Cited: febrero 11, 2011.] <http://www.mastermagazine.info/termino/5560.php>.

ltilib.sourceforge.net. sourceforge.net. *sourceforge.net.* [Online] [Cited: 5 19, 2011.] <http://ltilib.sourceforge.net>.

Lu, HB, Zhang, YJ and Yao, YR. 1999. *Robust Gradual Scene Change Detection.* Beijing : Department of Electronic Engineering, Tsinghua Universiv, 1999. 100084.

Microsoft. 2005. msdn. *msdn.microsoft.com.* [Online] Microsoft, 2005. [Cited: febrero 14, 2011.] <http://msdn.microsoft.com/es-us/library/kx37x362%28v=vs.80%29.aspx>.

Ngo, C.-W., Ma, Y.-F. and Zhang, H.-J. 2003. *Automatic video summarization by graph modeling.* s.l. : Computer Vision, IEEE International Conference , 2003.

Ngo, Chong Wah, Ma, Yu Fei and Jiang, Hong Zhang. 2005. Video Summarization and Scene Detection by Graph Modeling. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.* 2005, Vol. 15, 2.

Nokia. 2008. QT home. [Online] Nokia, 2008. [Cited: febrero 17, 2011.] <http://qt.nokia.com/products/developer-tools?currentflipperobject=821c7594d32e33932297b1e065a976b8>.

Rafael C. Gonzalez, Richard E. Woods. 2002. *Digital Image Processing.* New Jersey : Prentice-Hall, 2002.

Robust Gradual Scene Change Detection. **Lu, H.B. and Zhang, Y.J. y Yao, Y.R.:** 1999. 10.1109, Kobe, Japon : Proc. IEEE International Conference on Image Processing, 1999, Vol. 3. ICIP.1999.817122..

Sáes Peña, Edmundo. 2006. *Segmentación Automática de Video.* Málaga : Universidad de Málaga, 2006.

Schildt, Herber. 04/2001. C ++. Guía de autoenseñanza. C ++. *Guía de autoenseñanza*. España : s.n., 04/2001.

Scientia et Technica. **Betancourt, Gustavo A. 2005.** No 27, Pereira : Scientia et Technica, 2005. LAS MÁQUINAS DE SOPORTE VECTORIAL (SVMs). UTP. ISSN 0122-1701.

Seyler, A. 1965. *Probability Distributions of Television Frame Di.* s.l. : Proceedings of the Institution of Radio and Electronics Engineers (IREE), 1965. pp. 355-366.

vxl.sourceforge.net. sourceforge.net. *sourceforge.net*. [Online] [Cited: 5 19, 2011.] <http://vxl.sourceforge.net>.

Wandelmer and Pedro, José San. 2006. *Arquitectura Paralela Para el Procesamiento y Análisis de Video Digital Utilizando Anotación MPEG-21 Aplicaciones Implantadas*. Madrid : Universidad Politécnica de Madrid, 2006.

Zhu, X., et al. 2004. *Exploring video content structure for hierarchical summarization*. s.l. : Multimedia Systems, 2004.

Ziyou , Xiong, Radhakrishnan, Regunathan and Rui, Yong . A Unified Framework for Video Indexing, Summarization, Browsing, and Retrieval. Illinois : s.n.

Anexos.

Anexo 1. Figuras.



Fig.1 Cambio de toma abrupto.



Fig.2 Cambio de toma gradual.

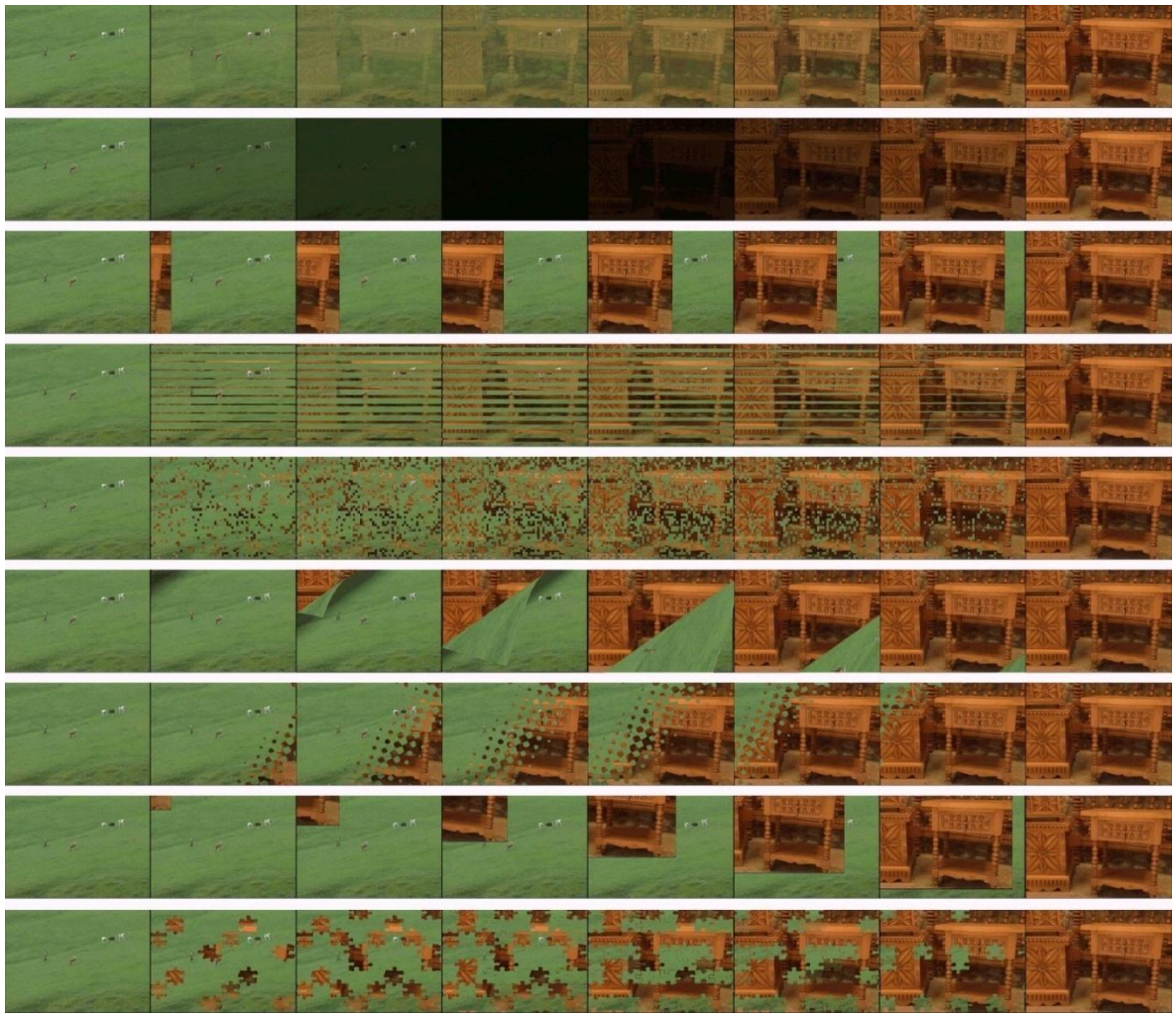


Fig.3 Cambios de tomas graduales con efectos de edición (Sáes Peña, 2006).

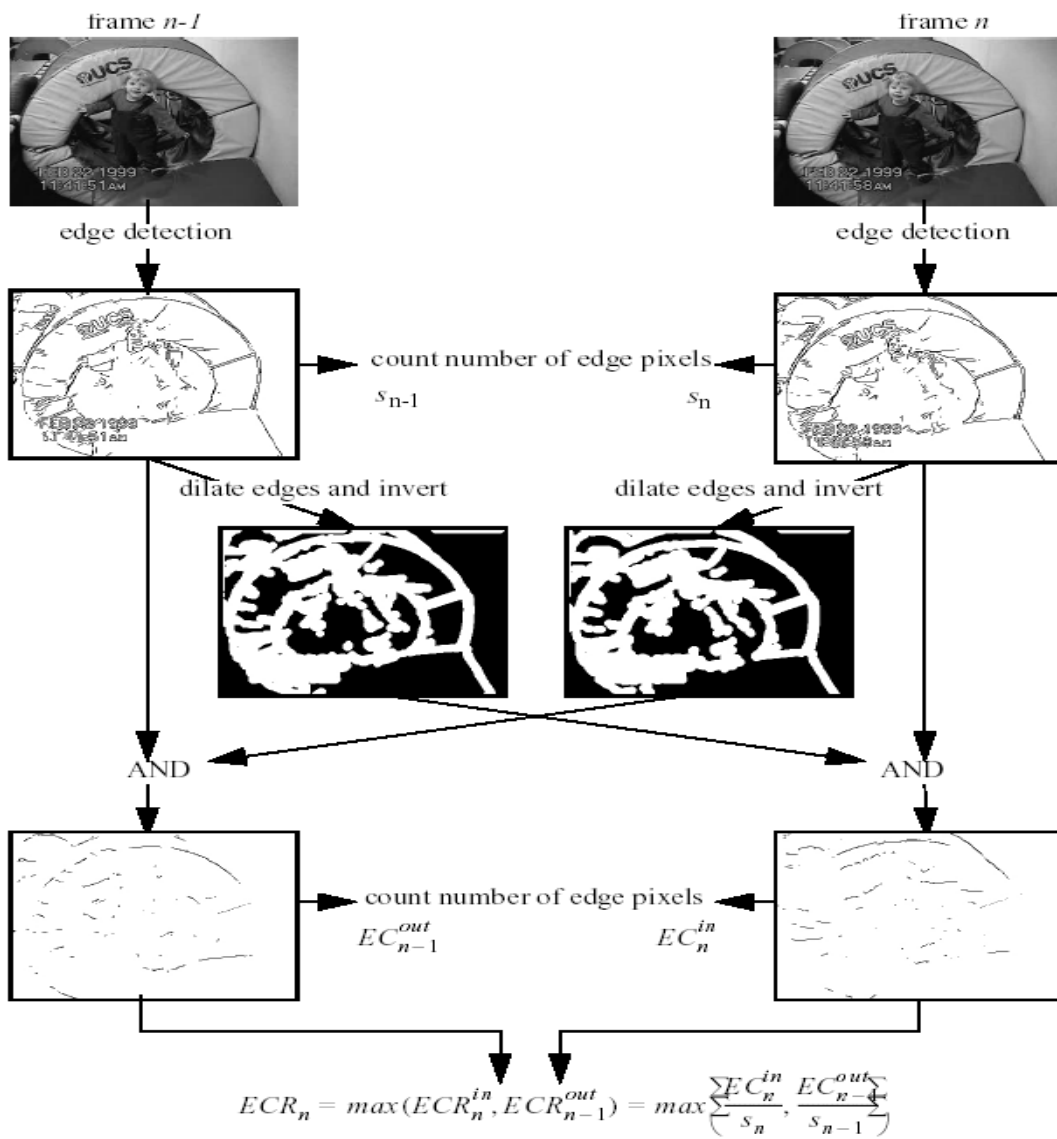


Fig.4 Esquema del algoritmo de detección basado en bordes.

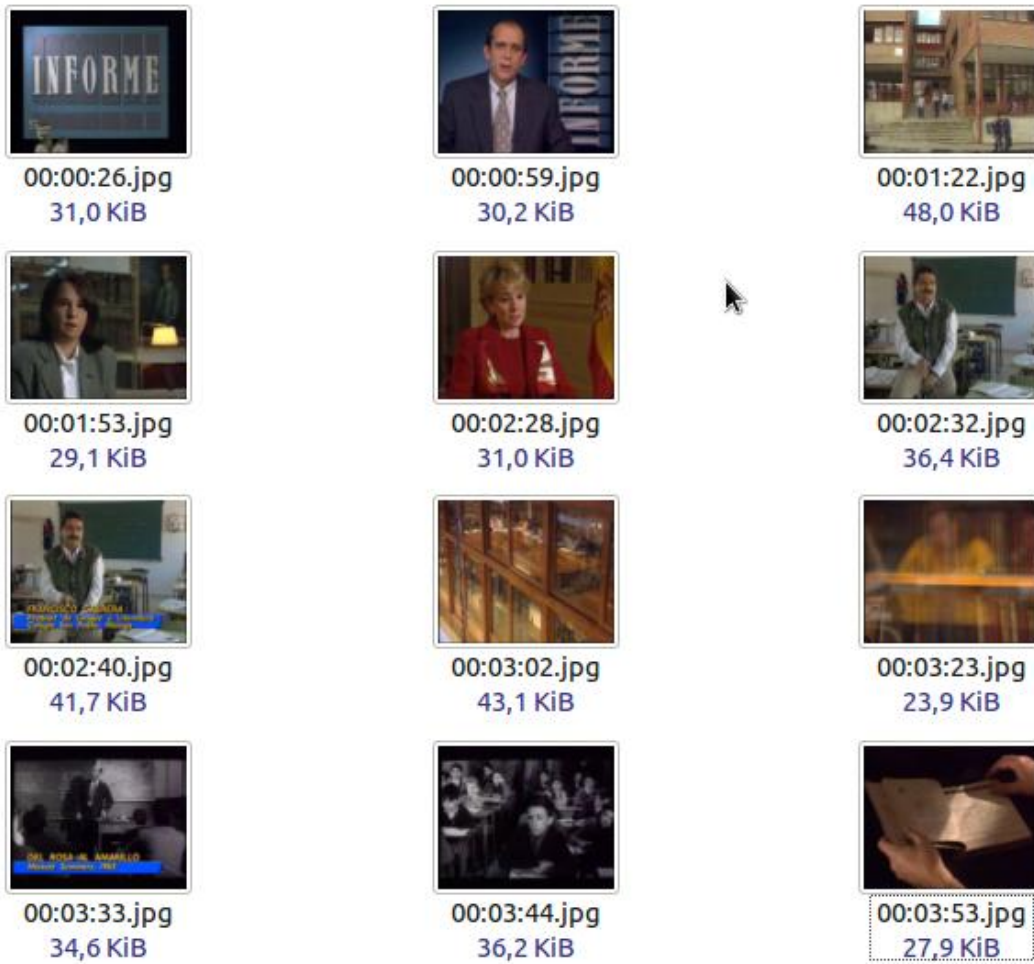


Fig.5 Resultado de la ejecución del algoritmo.

Glosario.

Archivo: 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.

Bit: del inglés **B**inary **d**igit (dígito binario), un bit es un dígito del sistema de numeración binario.

Dendograma: representación gráfica en forma de árbol que ilustra el proceso de agrupación en un análisis de clúster.

Edición: es el mecanismo mediante el cual se enlazan dos tomas consecutivas en una secuencia de video.

Escalabilidad: la medida de la capacidad de crecimiento de un servicio o de una aplicación para satisfacer demandas de rendimiento cada vez mayores.

Grafos: un grafo es un conjunto, no vacío, de objetos llamados vértices (o nodos) y una selección de pares de vértices, llamados aristas que pueden ser orientados o no.

HSV: el modelo HSV (del inglés Hue, Saturation, Value – Matiz, Saturación, Valor), también llamado HSB (Hue, Saturation, Brightness – Matiz, Saturación, Brillo), define un modelo de color en términos de sus componentes.

Hyperplano: concepto de geometría, es una generalización del concepto de plano.

Multimedia: archivo que utiliza conjunta y simultáneamente diversos medios, como imágenes, sonidos y texto, en la transmisión de una información.

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

Plano: superficie plana paralela al horizonte, colocada en la parte inferior del cuadro, donde se proyectan los objetos, para construir después, según ciertas reglas, su perspectiva.

Vector: un conjunto de variables del mismo tipo cuyo acceso se realiza por índices.

Varianza: dada una variable aleatoria X con media $\mu = E(X)$, se define su varianza, $\text{Var}(X)$ como: $\text{Var}(x) = E[(X - \mu)^2]$