Universidad Nacional del Sur

Tesis Doctor en Ingeniería

Javier Iparraguirre

Título:

Sumarización de video en línea basada en detección de características visuales locales

BAHIA BLANCA ARGENTINA

2014

Prefacio

Esta Tesis se presenta como parte de los requisitos para optar al grado Académico de Doctor en Ingeniería, de la Universidad Nacional del Sur y no ha sido presentada previamente para la obtención de otro título en esta Universidad u otra. La misma contiene los resultados obtenidos en investigaciones llevadas a cabo en el Laboratorio de Ciencias de las Imágenes, dependiente del Departamento de Ingeniería Eléctrica y Computadoras durante el período comprendido entre el 4 de marzo de 2008 y el 25 de noviembre de 2014, bajo la dirección del Profesor Claudio Delrieux del Departamento de Ingeniería Eléctrica y Computadoras, Universidad Nacional del Sur.

Javier Iparraguirre
Noviembre 2014

Departamento de Ingeniería Eléctrica y Computadoras
UNIVERSIDAD NACIONAL DEL SUR

Resumen

La extracción de contenido en multimedios ha sido y sigue siendo un tema arduamente investigado durante las últimas décadas. En la literatura se puede encontrar esta rama del conocimiento como MIR (Multimedia Information Retrieval). La sumarización de video se clasifica como una de las ramas dentro de MIR. Hace solo unos pocos años, se han instalado una gran cantidad de cámaras de video en forma masiva. La sumarización de video sigue siendo un problema intensamente investigado y se considera que aún no ha sido resuelto.

Este trabajo presenta un método basado en características visuales de las imágenes. Las principales ventajas son la generalidad de la sumarización, la flexibilidad en la detección y el acotado costo computacional. Estas particularidades lo hacen único y se plantea como una seria solución al problema de extraer contenido de grandes volúmenes de video.

Además del método se presenta un marco de medición cuantitativa de sumarización. Varios autores coinciden en la literatura que el problema de la comparación de sumarizaciones está abierto. El método se basa en trabajos existentes y mejora la metodología usada.

Finalmente se compara cualitativamente y cuantitativamente el algoritmo propuesto con los métodos más referenciados en la literatura. Los resultados cuantitativos muestran puntuaciones superiores a las propuestas existentes. El método propuesto aporta una solución capaz de procesar en línea y es única en su tipo. Las aplicaciones potenciales son muchas y se espera poder lograr un gran impacto en el área de la extracción de contenido en video.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Extracción de contenido en medios audiovisuales

La extracción de contenido en multimedios ha sido y sigue siendo un tema arduamente investigado durante las últimas décadas. Lew et al. [80] definen MIR (Multimedia Information Retrieval) como *la búsqueda de de conocimiento en todas las formas, en todos lados*. Una definición de contornos tan amplios plantea desafíos extraordinarios, cuyos objetivos finales están aún hoy lejos de poderse cumplir. El objetivo final no esta cerca en el horizonte, quizás sea un imposible. Sin embargo, la investigación y el desarrollo asociados a la búsqueda del objetivo planteado tienen un gran auge en nuestros días.

Los orígenes de MIR se pueden encontrar varias décadas atrás cuando comenzaron a aparecer los primeros conceptos de digitalización [80]. En aquellos momentos los fundamentos relacionados con la extracción de contenido no estaban acordados en la comunidad científica. Los lineamientos fundacionales de MIR fueron aportados por otras áreas de la ciencia tales como la inteligencia artificial, la teoría de optimización, la visión de computadoras y el reconocimiento de patrones. Incluso áreas del conocimiento tales como la psicología juegan un rol importante a la hora de proveer los fundamentos necesarios para diseñar una eficiente interacción con el usuario.

Según mencionan Lew et al. [80], los primeros ejemplos de extracción de contenido en multimedia estaban basados en algoritmos de visión de computadoras orientados a la búsqueda de similitudes. Algunos de los ejemplos más influyentes fueron QBIC [43] y Virage [13]. Luego de un tiempo, la misma estrategia de búsqueda de similitudes se extendió a motores de búsqueda de imágenes en Internet tales como Webseek [118] y Webseer [44].

Durante la década de los 90s las imágenes y los videos digitales comenzaron a ser de uso masivo [26]. Comenzó a gestarse dentro de la comunidad de investigadores la idea de

que ya no eran suficientes los conceptos clásicos. Hasta ese momento, la mayoría de los esfuerzos se focalizaban en tecnologías centradas en las computadoras (computer-centic). En contraste, los nuevos desarrollos de ese momento comenzaron a tener una orientación hacia las necesidades de los usuarios humanos (human-centric). La motivación que motorizó el cambio de paradigma se basaba en cruzar la brecha existente entre los usuarios y las tecnologías que le entregan información sobre las aplicaciones multimedia.

Con el arribo de los dispositivos móviles y la gran popularidad de los servicios multimedia, la industria y los investigadores relacionados a MIR se vieron forzados a realizar nuevos desarrollos. Con la nueva realidad fue necesario realizar nuevas técnicas de indexado para organizar el contenido. Además fue necesario comenzar el desarrollo de herramientas que permitan explorar y sumarizar grandes colecciones de contenido multimedia. En 2008 se realizó la primera conferencia internacional sobre MIR organizada por la ACM (Association for Computing Machinery) [26]. A partir de ese momento se puede afirmar que MIR toma cuerpo como entidad dejando de ser una subsección dentro de otra rama del conocimiento tal como visión de computadoras.

1.2. Sumarización de video como caso particular de extracción de contenido

La sumarización de video se clasifica como una de las ramas dentro de MIR. En la literatura puede también encontrarse a la abstracción de video como un equivalente a la sumarización [129]. El marco teórico relacionado con el planteo del problema asociado a la sumarización de video puede encontrarse en una serie de trabajos con alto impacto en el área.

Truong y Venkatesh presentaron un excelente trabajo que clasifica sistemáticamente la abstracción de video [129]. Posteriormente, Money y Agius [97] presentaron un trabajo que separa las técnicas de sumarización de las salidas de las mismas. En otro trabajo reciente realizado por Hu el at. [63] es posible encontrar el estado del arte del indexado de la extracción de contenido en video. El marco conceptual del presente capítulo se basa en los trabajos citados anteriormente.

Una secuencia de video está compuesta por más de una fuente de información. El caso más frecuente es un video compuesto por imágenes y sonido. Además se pueden encontrar fuentes adicionales de información tales como texto, comentarios de humanos que han observado al video o sitios de Interet que hacen referencia al video en cuestión. Este hecho convierten a la sumarización de video en un problea muti-modal y sumamante complejo.

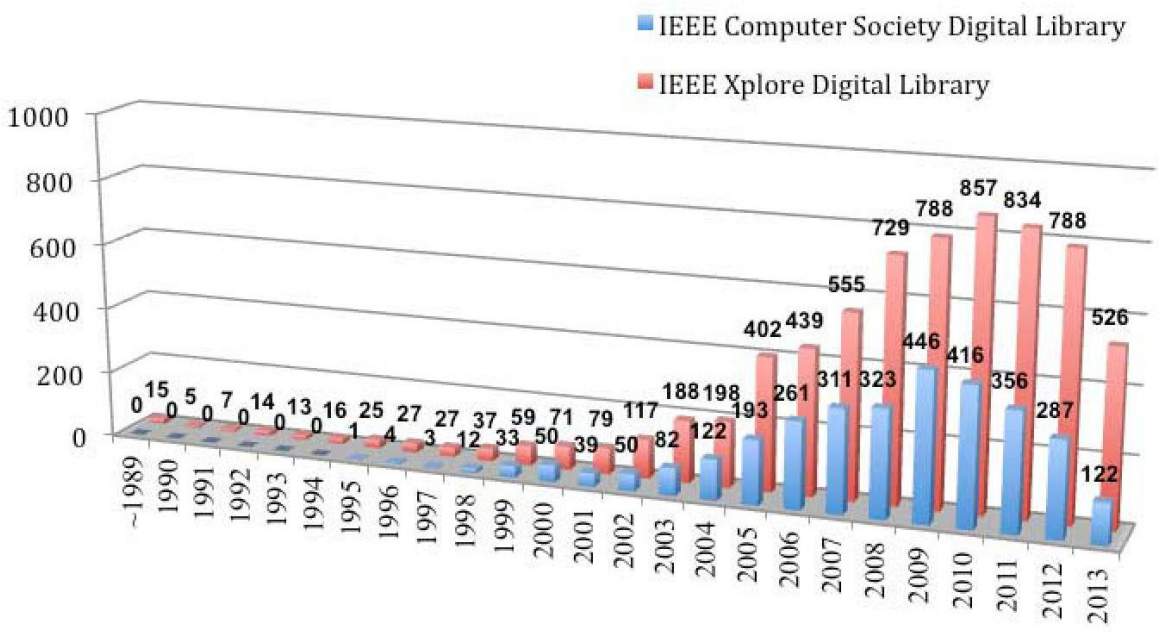

Figura 1.1 Publicaciones en IEEE Computer Society Digital Library y IEEE Xplore que contienen las palabras claves *video* y *surveillance* [64].

Los Grandes Datos, (Big Data), continúan creciendo exponencialmente. Los videos de vigilancia se han transformado en una de las mayores fuentes [64]. En los años recientes se han instalado una cantidad creciente de cámaras de video en forma masiva. Algunos ejemplos cotidianos son ascensores, cajeros de bancos, paredes de los edificios y calles entre otras. Además existen fuentes de generación de video privadas generadas por computadoras portátiles y teléfonos inteligentes.

Ciudades como Pekín o Londres poseen alrededor de un millón de cámaras instaladas. Estas cámaras capturan en una hora mucho más de lo que la BBC (British Broadcat Corporation) o la CCTV (China Central Television) poseen en sus archivos como Programas de TV. De acuerdo con el reporte reciente de la International Data Corporation "El Universo Digital en 2020" la mitad de los Grandes Datos globales son videos de vigilancia. Se espera que el porcentaje se incremente al 65 % para el 2015 [64].

Actualmente la abstracción de video es un área de investigación activa y hay un gran rango de problemas a resolver. En la Figura 1.1 se muestra un histograma de publicaciones en IEEE Computer Society Digital Library e IEEE Xplore que contienen las palabras claves *video* y *surveillance* [64]. Los números correspondientes al año 2013 pueden variar debido a que puede haber publicaciones en proceso de ingreso a las bases de datos. Se puede observar un fuerte crecimiento en los últimos 10 años.

1.3. Aportes del trabajo

El presente trabajo presenta un método de sumarización aplicable a cualquier tipo de contenido. Tiene como principales fortalezas la generalidad de la solución, la posibilidad de trabajar en línea, y un bajo costo computacional. Los resultados producidos tienen una calidad igual o superior a las alternativas publicadas según los trabajos revisados a la fecha. La salida brindada por el algoritmo puede ser un conjunto de cuadros claves o un video resumen. A partir de las características mencionadas, la prouesta se posiciona como una herramienta ideal para procesar grandes volúmenes de video.

Además del método de sumarización propuesto, este trabajo hace el aporte de una metodología de cuantización de la calidad de sumarización. Varios autores mencionan la necesidad de un esquema de evaluación consistente. La metodología planteada es compatible con métodos existentes y supera el estado del arte. Se hace uso del coeficiente Kappa de Cohen a partir de la necesidad de lidiar con la realidad de que la evaluación de una sumarización de video es un problema de clases desbalanceadas. Se define un marco claro de evaluación y se hace disponible públicamente la implementación.

El algortimo de sumarización y la metodología de evaluación cuantitativa de resultados son inéditos y aportan un avance respecto a lo relevando en la literatura. Presentan soluciones a los desafíos de la sumarización de grandes volúmenes de datos y a la necesidad de un marco de comparación consistente. A partir de los conceptos planteados, se puede afirmar que el aporte de este trabajo es sólido y está fundamentado en los avances documentados en la literatura.

1.4. Trabajos Publicados

Antes de la redacción del presente documento se han publicado varios trabajos los cuales fueron revisados por pares [65–67]. Dentro de los más destacados se pueden citar una publicación titulada “Speeded-up Video Summarization Based on Local Features” en el Simposio Internacional de Multimedia (ISM) de IEEE. La conferencia se realizó en diciembre de 2013 en Anaheim, California, Estados Unidos [66].

Otra publicación que merece ser destacada es el trabajo titulado “Online Video Summarization Based on Local Features”. Este trabajo fué publicado en 2014 en la revista científica International Journal of Multimedia Data Engineering and Management (IJMDEM). La revista está editada por IGI Global y tiene sede en Estados Unidos [67].

Capítulo 2

Planteo del problema y contexto dentro de la literatura

Los métodos de sumarización de video pueden ser clasificados por más de un criterio [5, 66]. La clasificación más popular consta en dividir los algoritmos por la salida que producen. En este caso, las sumaziaciones pueden producir usa salida de *cuadros clave* (keyframes en inglés) o un *video condensado* (video skimm en inglés).

Los cuadros claves suelen ser llamados *cuadros representativos* o *sumarización estática*. El conjunto de salida consiste en una colección de imágenes destacadas extraídas de una fuente de video dada. En la Ecuación 2.1 se describe el conjunto de cuadros clave $\mathscr{R}$, donde $A_{cuadosclave}$ representa el procedimiento de extracción de cuadros clave. La notación usada en este trabajo se describe en la Tabla 2.1.

$$\mathscr{R} = A_{cuadrosclave}(\mathbf{V}) = \{f_{r1}, f_{r1}, \ldots, f_{rk}\} \tag{2.1}$$

Los métodos de condensado de video suelen ser llamados *sumarización dinámica* o *resúmen secuencial*. Este tipo de abstracción de video consiste en una colección de segmentos de video unidos en forma abrupta o gradual. Como entidad es un video aunque el largo del mismo es mucho menor que el material original. Una forma muy popular de condensado de video son los avances de las películas en el cine. La definición de un video condensado $\mathscr{K}$ se puede encontrar en la Ecuación 2.2. El procedimiento se representa como $A_{videocondensado}$. E_i es el segmento i contenido en $\mathbf{V}$. Los segmentos se incluyen en la sumarización. El operador $\odot$ representa la operación de integración de dos segmentos (gradual o abrupta entre otras posibles).

$$\mathscr{K} = A_{videocondensado}(\mathbf{V}) = \mathbf{E}_{i1} \odot \mathbf{E}_{i2} \odot \ldots \odot \mathbf{E}_{ik} \tag{2.2}$$

Tabla 2.1 Detalle de la notación usada

Símbolo	Descripción
$\mathbf{V}$	Video o segmento de video (toma o muestra) del cual se extrae un resumen en forma independiente
f_i	El cuadro i de la secuencia de video
$\mathbf{n}$	El número de cuadros en $\mathbf{V}$ donde $\mathbf{n} = \lvert\mathbf{V}\rvert$
$\mathscr{R}$	El conjunto de todos los cuadros clave de $\mathbf{V}$
$\mathbf{k}$	El número de cuadros clave de un video condensado
r_i	El cuadro clave i del video $\mathbf{V}$
$[b_i, b_{i+1})$	Los límites del segmento que es representado por el cuadro clave f_{ri}
$\mathscr{K}$	El video condensado extraído de $\mathbf{V}$
$\odot$	Extracto ensamblado o integración que une tramos individuales para crear un video condensado
$\mathbf{E}_i$	El tramo i de un video condensado
ρ	Perspectiva de la sumarización
$\mathscr{D}(.)$	Función de diferencia simétrica
$\mathscr{C}(f_i, f_j)$	Función de cambio de contenido, no necesariamente simétrica

Una ventaja de la sumarización dinámica sobre la estática es la posibilidad de incluir audio y elementos visuales al resultado. Con los agregados se puede mejorar la expresividad del resumen. Además es más entretenido para el usuario ver un avance de video en contraste con una serie de cuadros estáticos.

El uso de los cuadros clave permite algunas manipulaciones que no son posibles con un condensado de video. La principal diferencia es que no existen restricciones de tiempo o sincronización en un conjunto de cuadros clave. Por lo tanto es posible ver todos los cuadros clave a la vez. Además el uso de la sumarización estática es usada para disminuir la complejidad computacional de algunos métodos de análisis de video tales como extracción de contenido e interpretación semántica.

Otra forma de clasificar a los métodos de sumarización de video es por los requerimientos del material a procesar. Algunos métodos demandan el video original completo para poder comenzar a procesar. En contraste, otras propuestas demandan solo una fracción del material para poder producir un resultado. Esta característica es importante en términos del espacio a demandar por el método sumarización.

Finalmente, existe un tercer tipo de clasificación dependiendo del dominio de operación de los métodos. Algunos algoritmos de sumarización de video trabajan en el dominio comprimido, mientras que otros simplemente operan con datos sin comprimir.

Los procedimientos que trabajan en el dominio comprimido, utilizan las ventajas provistas por los métodos estándar de compresión de video. Las propuestas que trabajan con material sin comprimir solo cuentan con el material en forma directa.

Con el propósito de brindar un mapa de conocimiento relacionado con la problemática de la sumarización de video, se presentan dos secciones descriptivas. En la sección 2.1 se describen las características salientes de los métodos que producen como salida cuadros clave. La sección 2.2 describe los atributos de las técnicas de condensado de video.

2.1. Extracción de cuadros clave

La forma más simple de extracción de cuadros clave es el muestreo uniforme a lo largo de todo el video origen [129]. Aunque desde el punto de vista computacional la estrategia es simple y eficiente, el resultado obtenido carece de riqueza ya que en un video no suceden los hechos relevantes con una separación uniforme entre ellos. La exploración de algoritmos de extracción de cuadros clave tiene en cuenta a las propuestas que analizan la dinámica de la secuencia de video. En la Figura 2.1 se pueden observar las características de los métodos de extracción de cuadros clave. Los atributos descriptos son el tamaño del conjunto de cuadros clave, la unidad de base, el alcance de la representación y el mecanismo computacional.

2.1.1. Tamaño del conjunto de cuadros clave

Existen varias opciones para determinar la cantidad de cuadros clave que se van a producir en un método automático. Dependiendo de la opción elegida, tiene fuertes consecuencias sobre el proceso de búsqueda del conjunto representativo óptimo. La cantidad de cuadros clave puede ser *a priori*, pude ser *a posteriori* o puede ser *determinada* internamente por el proceso de abstracción. La mayoría de las técnicas solo ofrecen una opción en términos del tamaño del conjunto de salida.

A priori

En este caso, la cantidad de cuadros clave está decidida antes de comenzar con la abstracción. Puede ser asignada como un número específico o como un porcentaje del total de cuadros en el video. Este método suele usarse en casos donde los recursos computacionales son escasos tales como plataformas móviles. Para esta aplicación puntual la cantidad seleccionada puede depender del ancho de banda de transmisión, de la capacidad de almacenamiento y del tamaño de la pantalla del receptor. El principal

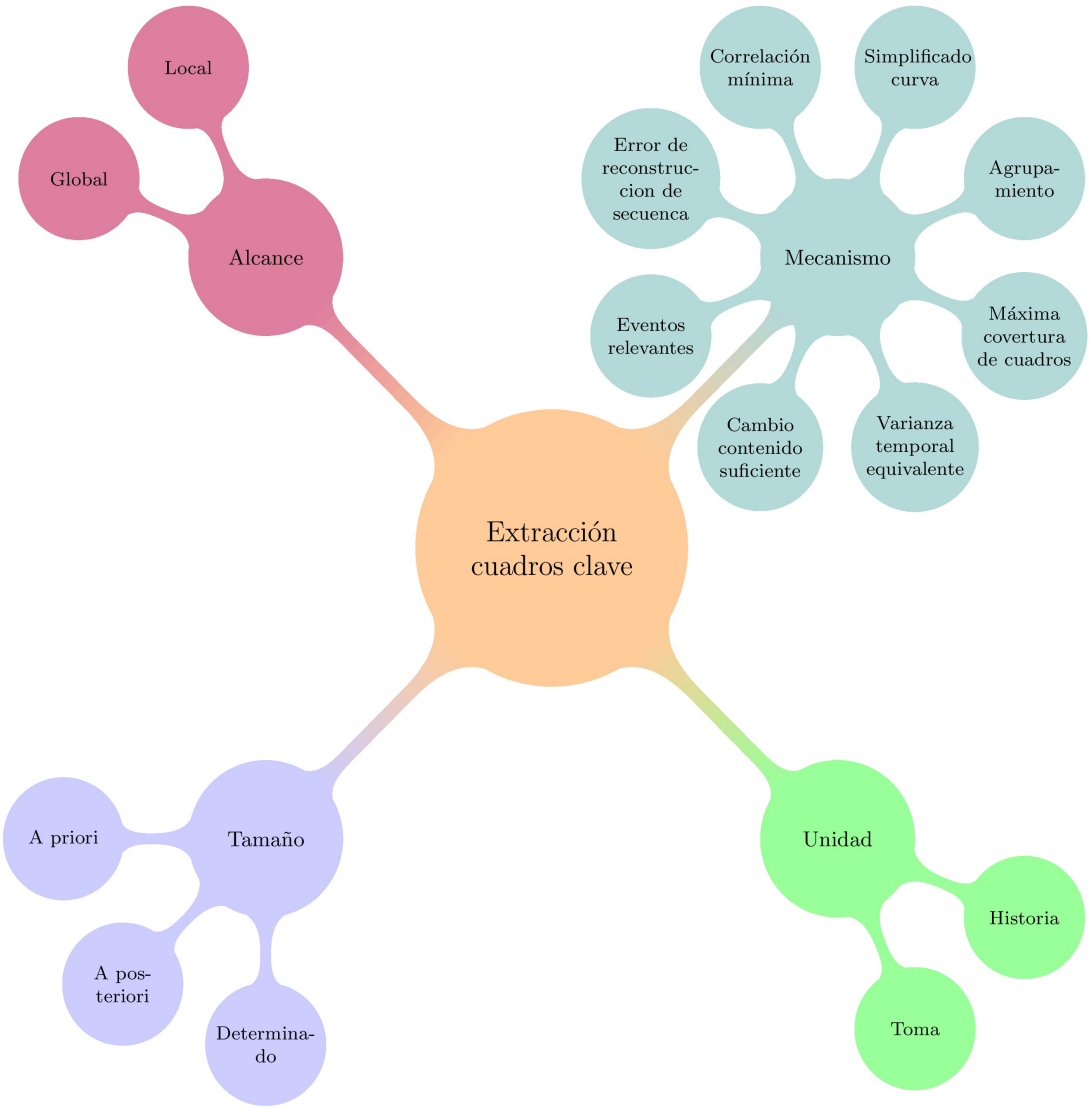

Figura 2.1 Atributos de las técnicas de extracción de cuadros clave.

problema de este criterio resulta en que no se puede asegurar que todos los segmentos importantes pueden ser capturados por el resultado.

En forma ideal, la extracción de cuadros clave de cantidad fija **k** puede ser formulada como un problema de optimización. El desafío consta en encontrar el conjunto de cuadros $\mathscr{R} = \{f_{r1}, f_{r2}, \ldots, f_{rk}\}$ que menos difiere de la secuencia original para una perspectiva dada. La Ecuación 2.3 describe la forma matemática del problema, donde **n** es la cantidad de cuadros en el video, ρ es la perspectiva de la sumarización en la que el usuario esta interesado y $\mathscr{D}$ es la medida de no-similitud.

$$\{r_1, r_2, \ldots, r_k\} = arg\min_{r_i} \{\mathscr{D}(\mathscr{R}, \mathbf{V}, \rho) | 1 \leq r_i \leq \mathbf{n}\} \tag{2.3}$$

La mayoría de los métodos de extracción de cuadros clave toman ρ como medida de *cobertura visual*, donde se intenta representar la mayor parte del contenido de un video con la menor cantidad de cuadros posibles. En la formulación matemática propuesta, ρ puede representar algo más genérico. Algunos ejemplos posibles son asociar ρ con un número de objetos o un número de caras entre varias opciones posibles.

A posteriori

En este caso, la cantidad exacta de cuadros extraídos no se conoce hasta que el proceso concluye. La cantidad de cuadros es usualmente determinado por el nivel de cambio visual. En el caso de videos altamente dinámicos, es posible que este tipo de sumarización produzca excesivos cuadros claves. El caso de extracción de cuadros clave sin límite en el tamaño de salida demanda de un umbral de tolerancia ε también llamado *nivel de fidelidad*. La Ecuación 2.4 describe el modelo asociado.

$$\{r_1, r_2, \ldots, r_k\} = arg\min_{r_i} \{\mathscr{D}(\mathscr{R}, \mathbf{V}, \rho) < \varepsilon, 1 \leq r_i \leq \mathbf{n}\} \tag{2.4}$$

Técnicamete es posible convertir este tipo de resumen en un resumen de cantidad fija a expensas de un alto costo computacional. Dado un número de cuadros fijos a obtener, es posible variar el nivel de ε hasta que se obtenga un resultado deseado.

Cantidad Determindada

En este caso, se sigue esencialmente el concepto del método de cantidad conocida a posteriori con el agregado de un mecanismo de decisión de cantidad óptima de cuadros clave. Por ejemplo, es posible agregar un algoritmo clasificador por grupos (o clusters) con el fin de tomar una decisión respecto a la cantidad de cuadros representativos [42].

2.1.2. Unidades

Es muy importante entender la *unidad de tiempo* que representa un cuadro clave. Es posible encontrar en la literatura al menos dos formas de uso de unidades. En algunos casos, el cuadro clave representa una toma dentro de un video. En otros casos el video por completo se puede representar por un cuadro clave. El primer método se lo conoce como *orientado a tomas* (shot-based). La segunda clasificación se denomina *orientado a la histora* (clip-based).

En el caso de los métodos orientados a la toma, la forma más directa de proceder consta en elegir el primer cuadro para representar toda una toma. También se puede elegir el último o el que se encuentra a la mitad. Sin embargo, la mayoría de los métodos toma el primer cuadro clave de una nueva toma para describir a los cuadros pertenecientes a la misma toma.

Los métodos orientados a la historia solo producen un cuadro clave para un video **V**. La principal ventaja de esta propuesta reside en la simpleza de procesar unos pocos cuadros por toma. En contraste, tiene la limitación de no poder representar la dinámica de la historia.

En el caso de los métodos orientados a tomas, cada una de las tomas se procesa como un video separado del resto. En este tipo de estrategia es posible que se produzcan cuadros claves similares debido a que las tomas dentro de un video pueden parecerse o repetirse. En el caso de tomar uno o dos cuadros clave por toma puede no ser práctico en los casos de videos de gran longitud.

2.1.3. Alcance en la representación del cuadro clave

Otro aspecto a tener en cuenta cuando se extraen cuadros claves es saber si el cuadro representa a los cuadros cercanos localmente o a segmentos no contiguos dentro del video. El segundo caso tiende a producir una menor cantidad de cuadros clave. Sin embargo, el primer caso preserva la temporalidad de los hechos lo cual puede ser valioso para entender el contenido del material.

2.1.4. Mecanismos de cómputo subyacentes

Los mecanismos de cómputo se clasifican según lo descripto en la figura 2.1. En esta subsección se describen cada uno de ellos.

Cambio contenido suficiente

Este método consta en procesar secuencialmente el video computando el nivel de cambio en el contendido de la secuencia. Solo se requiere un conocimiento acotado desde

el comienzo del video hasta el punto en el que se está procesando. Un cuadro clave se selecciona cuando el nivel de cambio acumulado supera un umbral dado. Se puede decir que cuando el *cambio de contenido es suficiente* se selecciona el cuadro clave $f_{r_{i+1}}$ basado en el cuadro clave anterior f_{r_i}. La Ecuación 2.5 describe la lógica asociada al método.

$$r_{i+1} = arg\min_{r_i} \left\{ \mathscr{C}(f_{r_{i+1}}, f_{r_i}) > \varepsilon, i < t \leq \mathbf{n} \right\} \tag{2.5}$$

La función de cambio de contenido es muy importante al momento de determinar el resultado del algoritmo. En la literatura se pueden encontrar varias propuestas. La más popular es la función de cambio basada en diferencias de histogramas [144]. Otras métricas se basan en la energía acumulada del cómputo de desplazamiento de bloques de imágenes [145].

El método de cambio de contenido suficiente se adapta para el uso en aplicaciones de tiempo real o para procesamiento en línea. Tiene la propiedad de poder representar las tomas de diversas longitudes a medida que se presentan en el video. Otra característica del método es que acota el uso de recursos computacionales. Su principal desventaja es que no resulta trivial la tarea de minimizar la cantidad de cuadros clave a elegir para un video dado. No es posible elegir la máxima cobertura a medida que avanza la historia. Tampoco es posible limitar la cantidad de cuadros clave a elegir ya que este parámetro depende del largo total de la entrada.

La Figura 2.2 muestra un ejemplo de cómo funciona un método de cambio de contenido. El eje de las abscisas representa el número de cuadro ordenado cronológicamente, mientras que el eje de las ordenadas representa el nivel de cambio desde el último cuadro clave elegido. Se puede observar que el nivel de cambio se acumula hasta que llega a un umbral determinado. Cuando el valor del cambio supera al umbral, se determina al cuadro como clave y se reinicia el cómputo de varianza de cambio.

Varianza temporal equivalente El método de varianza temporal equivalente es similar al de cambio de contenido suficiente pero con la restricción de fijar la cantidad de cuadros clave *a priori*. Además asume que un correcto conjunto de cuadros clave debe representar segmentos de video $(b_i, b_{i+1} - 1)$ de igual varianza temporal. En este algoritmo b_i y r_i son computados en forma separada. Se define $\mathscr{V}(b_i, b_{i+1} - 1)$ como la varianza de el segmento $(b_i, b_{i+1} - 1)$ representado por el cuadro clave f_{r_i}. La Ecuación 2.6 representa la selección de límites ideal.

$$\mathscr{V}(b_1, b_2) = \mathscr{V}(b_2, b_3) = \cdots = \mathscr{V}(b_k, b_{k+1}) \tag{2.6}$$

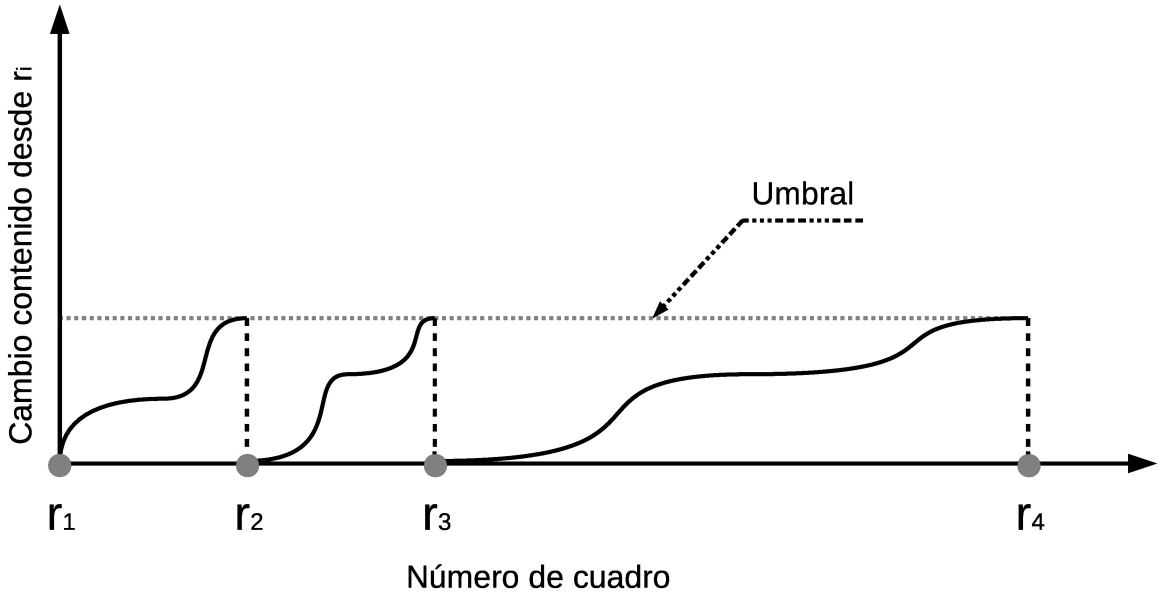

Figura 2.2 Cambio suficiente

La solución ideal es muy difícil de conseguir. Sun y Kankanhalli [121] formularon el problema como una optimización tal como se muestra en la Ecuación 2.7. La función de varianza $\mathscr{V}$ se aproxima por el acumulado de cambio de contenido entre cuadros consecutivos [42]. Además, $\mathscr{V}$ puede ser aproximada por la diferencia entre el primer y el último cuadro de la secuencia. De esa forma, el problema de la Ecuación 2.7 puede ser resuelto borrando el conjunto de cuadros consecutivos con el menor cambio de contenido hasta que el número de cuadros obtenido es el deseado [121]

$$\{b_1, b_2, \ldots, b_{\mathbf{k}+1}\} = arg\min_{b_i} \sum_{i=1}^{\mathbf{k}} \sum_{j=1}^{\mathbf{k}} |\mathscr{V}(b_i, b_{i+1}) - \mathscr{V}(b_j, b_{j+1})| \tag{2.7}$$

Luego que los límites de los segmentos $\{b_i\}$ se han definido, se seleccionan los cuadros clave correspondientes al segmento. Fauvet et al. [42] seleccionan el primero y el del medio. En contraste, Lee y Kim [75] seleccionan el cuadro que minimiza la distancia total del segmento.

En términos generales, el método de varianza temporal equivalente es computacionalmente más demandante que el cambio de contenido suficiente. Sin embargo, la salida producida es globalmente "óptima" según el criterio definido en la Ecuación 2.7. Además, el conjunto de cuadros clave de salida es independiente del orden de procesamiento.

Máxima cobertura de cuadros

Este método busca obtener una lista de cuadros que mejor pueda "cubrir" una secuencia de video. En la literatura es conocido como el método de la fidelidad (fidelity-based) y fue propuesto inicialmente por Chang et al. [21]. Suponiendo que $\mathbf{C}_i(\varepsilon)$ representa el conjunto de cuadros en $\mathbf{V}$ que pueden usar f_i como el cuadro representativo con respecto a un valor de tolerancia ε. En este planteo, el conjunto de cuadros clave óptimo de $\mathbf{V}$ puede ser extraído de la Ecuación 2.8.

$$\{r_1, r_2, \ldots, r_\mathbf{k}\} = arg\,\min_{r_i}\{\mathbf{k} | \mathbf{C}_{r_1}(\varepsilon) \cup \mathbf{C}_{r_2}(\varepsilon) \cup \ldots \cup \mathbf{C}_{r_k}(\varepsilon) = \mathbf{V}\} \tag{2.8}$$

Cuando el número de cuadros clave es especificado como una constante, la formulación del problema cambia. Se puede ver al menos de dos formas. La pimera interpretación se puede ver como la búsqueda del valor de fidelidad mínimo ε de manera que todos los cuadros pueden ser representados al menos por un cuadro clave. Otra forma de formular el problema es encontrar el conjunto de cuadros que representa la mayor cantidad de cuadros posibles. La Ecuación 2.9 representa la primera formulación, mientras que la segunda la expresa la Ecuación 2.10.

$$\{r_1, r_2, \ldots, r_\mathbf{k}\} = arg\,\min_{r_i}\{\varepsilon | \mathbf{C}_{r_1}(\varepsilon) \cup \mathbf{C}_{r_2}(\varepsilon) \cup \ldots \cup \mathbf{C}_{r_k}(\varepsilon) = \mathbf{V}\} \tag{2.9}$$

$$\{r_1, r_2, \ldots, r_\mathbf{k}\} = arg\,\max_{r_i}\{|\mathbf{C}_{r_1}(\varepsilon) \cup \mathbf{C}_{r_2}(\varepsilon) \cup \ldots \cup \mathbf{C}_{r_k}(\varepsilon)|\} \tag{2.10}$$

En este método es de vital importancia determinar la cobertura del cuadro y el procedimiento de optimización. En la literatura se encuentran varias propuestas a este desafío. Chang et al. [21] propuso el uso de un algoritmo codicioso (greedy). También es posible encontrar propuestas que hacen uso de programación dinámica [139]. Cooper and Foote [33] utilizaron el método TF-IDF (frequency–inverse document frequency).

La ventaja del método de máxima cobertura de cuadros es que un cuadro clave no representa a un conjuntos de cuadros contiguos que lo preceden. En consecuencia, el conjunto de cuadros clave es más conciso. En contraste, la similitud de cuadros debe ser computada entre todos los pares. Este tipo de demanda computacional convierte al método en impráctico para sumarizar en tiempo real. Además, la respuesta óptima de este método depende fuertemente de la métrica de similitud visual elegida.

Agrupamiento

Este método considera un cuadro de video como un punto en un espacio de características (features). Las características pueden ser cualesquiera de las conocidas dentro de visión de computadoras tales como histograma de color entre otras. A partir de situarse en el espacio de características, se asume que se pueden encontrar puntos representativos basados en los agrupamientos (clusters). Esos puntos representativos se usan como cuadros clave de una secuencia de video. En esta propuesta no se toma como dado las suposiciones mostradas en las ecuaciones 2.3 y 2.3. Se hace uso de los conocimientos desarrollados para el agrupamiento de datos. El agrupamiento puede ser realizado basado en una toma o en una secuencia. Existen cuatro pasos para llevar a cabo este procedimiento y se denominan *preprocesado*, *agrupamiento*, *filtrado* y *extracción de puntos representativos*.

Durante el *preprocesado* se toma el video original y se lo acondiciona con el fin de facilitar la operación en el espacio de las características. El acondicionamiento ayuda a realizar una clasificación más eficiente y además reduce la complejidad de cómputo. En el caso de Yu et al. [143] cada cuadro es transforamdo en un espacio de autovalores usando el análisis de componentes principales (PCA). Con esta transformación, se reducen las dimensiones del problema minimizando la pérdida de la riqueza de los datos. Otra solución al mismo problema es la propuesta por Xiong et al. [136]. En este caso extraen un conjunto de cuadros usando el método de cambio suficiente de contenido y luego los usan para agruparlos. En el caso de Girgensohn y Boreczky [49], eligen un número predefinido de cuadros bajo los cuales tienen grandes diferencias con el cuadro anterior. Los autores del trabajo asumen que dos cuadros adyacentes que no tienen grandes diferencias entre si tienen grandes probabilidades de ser clasificados dentro del mismo grupo.

El *agrupamiento* de los datos consta en determinar conjuntos que engloben puntos similares dentro del espacio de las características. Xiong et al. [136] emplean una técnica de agrupamiento secuencial donde se asigna un cuadro a un cluster (o grupo) si la similitud excede un valor de umbral dado. En caso de que el grupo no exista, crean un nuevo grupo. En este caso usan como métrica de similitud la norma L_1. Yu et al. [143] utilizan un algoritmo de agrupamiento difuso (c-Means). A diferencia de los anteriores trabajos, Girgensohn y Boreczky [47] utilizan GMMs (Gaussian Mixture of Models) para computar los grupos.

Para realizar un *filtrado* de grupos es necesario descartar los grupos que se consideran "ruidosos" o que no contienen información significativa. En el caso de Zhuan et al. [147] descartan todos los grupos que son de menor tamaño que el promedio. Girgensohn y Boreczky [49] eligen solo los clusters que contienen una secuencia de video de nueve

segundos sin interrupciones. Uchihashi at al. [131] crean una métrica de importancia basada en el tamaño del cluster y la redundancia dentro del grupo.

Finalmente es necesaria la *extracción de puntos representativos*. La solución más intuitiva consta en elegir el punto más cercano a centroide del grupo [143]. En el caso de que los grupos se elijan en función del segemento de video que representen, es posible elegir un cuadro clave que se encuentre en el medio del segmento [131].

El método de agrupamiento es uno de los más populares en la literatura. Sin embargo, presenta gran dificultad en el proceso de selección de clusters. El método no es apto para el procesamiento en línea del video debido a la naturaleza mísma de la formulación. Además, presenta dificultades cuando es necesario preservar la progresión temporal de los cuadros clave elegidos.

Correlación mínima entre cuadros

Este tipo de técnicas asume que la frecuencia de extracción de cuadros clave de un video está limitada y además el conjunto de cuadros clave elegido debe tener mínima correlación ente sus elementos. La Ecuación 2.11 expresa la lógica detrás del método, donde $Corr(.)$ representa la función de métrica de correlación.

$$\{r_1, r_2, \ldots, r_{\mathbf{k}}\} = arg\min_{r_i}\{Corr(f_{r1}, f_{r2}, \ldots, f_{rk})\} \tag{2.11}$$

Doulamis et al. [40] consideró las contribuciones de correlación entre todos los pares de cuadros del video. La Ecuación 2.12 expresa la propuesta, donde $Corr(f_i, f_j)$ es el coeficiente de correlación entre dos vectores de características (f_i, f_j). Es posible usar más de un algoritmo para encontrar una solución cercana a la óptima. Entre los posibles se puede encontrar búsqueda logarítmica [40], búsqueda estocástica [10] o algoritmos genéticos [39].

$$Corr(f_{r1}, f_{r2}, \ldots, f_{rk}) = \left(\sum_{i=1}^{\mathbf{k-1}} \sum_{j=i+1}^{\mathbf{k}} Corr(f_{r_i}, f_{r_{i+1}})^2\right)^{1/2} \tag{2.12}$$

En el caso de que solo se considera las correlaciones entre los elementos sucesivos, la Ecuación 2.11 puede ser expresada como se muestra en la Ecuación 2.13. Sin embargo, esta formulación no tiene como propósito dividir la secuencia en segmentos temporales uniformes. El objetivo de la propuesta consta en maximizar la diferencia total dinámica.

$$\{r_1, r_2, \ldots, r_{\mathbf{k}}\} = arg\min_{r_i}\left\{\sum_{i=1}^{\mathbf{k-1}} Corr(f_{r_i}, f_{r_{i+1}})\right\} \tag{2.13}$$

El método de correlación mínima asegura un bajo nivel de redundancia en los cuadros clave elegidos. En contraste, es significativamente sensible a los casos poco frecuentes. Porter et al. [110] han intentado solucionar este problema tratando a cada cuadro clave como un nodo dentro de un grafo dirigido compuesto por arcos con peso. Este método requiere que el primer y el último cuadro sean tomados como cuadros clave. El conjunto de salida óptimo es el camino más corto entre el primer y último nodo.

Error de reconstrucción de secuencia

Este método de sumarización está basado en una métrica denominada como SRE (Sequence Reconstruction Error) [76, 87, 88]. La métrica SRE cuantifica la capacidad que contiene un conjunto de cuadros dado para reproducir una secuencia de video. Es de gran utilidad cuando el número de cuadros clave esta determinado *a priori* y la progresión temporal es relevante para la aplicación.

La Ecuación 2.14 define la métrica SRE para un conjunto de cuadros clave, donde $\mathscr{D}(.)$ define la diferencia entre dos cuadros. La formulación asume que se cuenta con una función de interpolación de cuadros $\mathscr{I}(t,\mathscr{R})$ que calcula las características de una imagen en un tiempo t en un video dado.

$$\mathscr{E}(\mathbf{V},\mathscr{R}) = \sum_{i=1}^{\mathbf{n}} \mathscr{D}(f_i, \mathscr{I}(i,\mathscr{R})) \tag{2.14}$$

En el caso que el número de cuadros clave está dado por **k**, el conjunto de cuadros óptimo tendrá un SRE mínimo y se define según la Ecuación 2.15. Debido a que la definición ideal es muy difícil de computar, se realizan simplificaciones que simplifican el problema. Los autores citados asumen que un cuadro clave representa un rango temporal $[b_i, b_{i+1}]$ de la secuencia de video. Por lo tanto es necesario definir la Ecuación 2.16.

$$\{r_1, r_2, \ldots, r_k\} = arg\,\min_{r_i}\{\mathscr{E}(\mathbf{V},\mathscr{R}), 1 \leq r_i \leq \mathbf{n}\} \tag{2.15}$$

$$\mathscr{I}(i,\mathscr{R}) = f_{r_t} \iff b_t \leq r_i < b_{t+1} \tag{2.16}$$

Por lo tanto, es posible expresar la formulación del error de reconstrucción de secuencia como se indica en la Ecuación 2.17.

$$\mathscr{E}(\mathbf{V},\mathscr{R}) = \sum_{i=1}^{\mathbf{n}} \mathscr{D}(f_i, \mathscr{I}(i,\mathscr{R})) = \sum_{i=1}^{\mathbf{k}} \sum_{j=b_i}^{b_{i+1}-1} \mathscr{D}(f_j, f_{r_i}) \tag{2.17}$$

La Figura 2.3 describe la lógica detrás del algoritmo. Los cuadros clave elegidos se

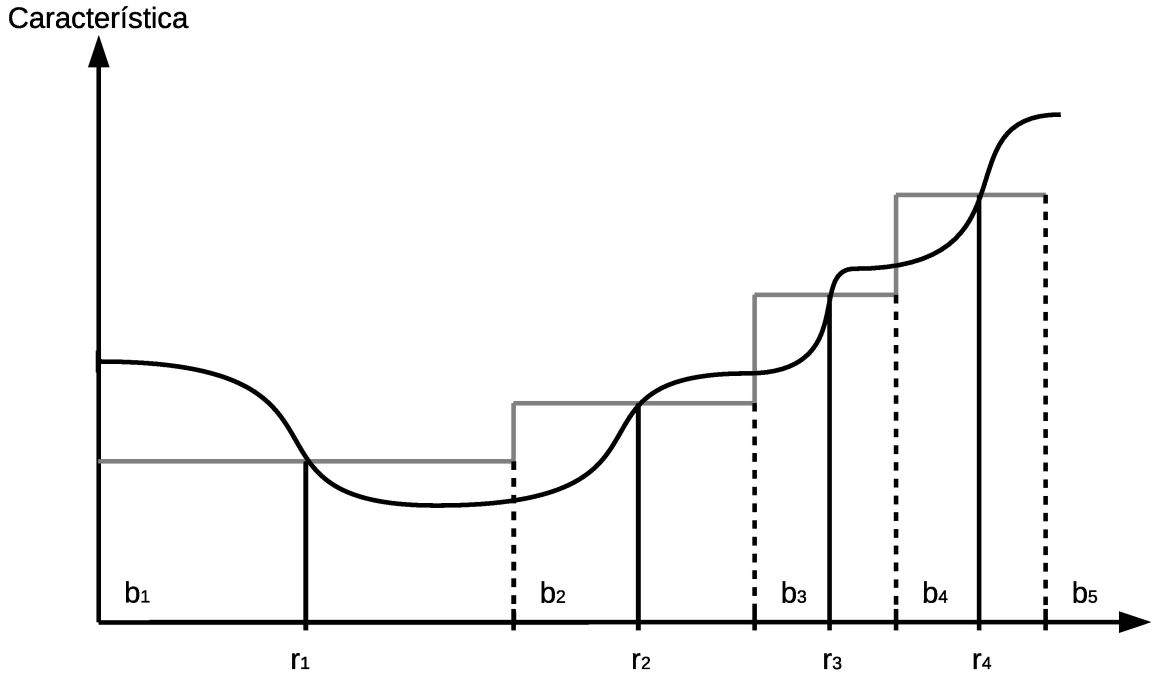

Figura 2.3 Error de reconstrucción de secuencia

indican como r_i y los límites en los cuales se produce la transición entre segmentos se denominan b_i. La curva indica el valor de SRE instantáneo, mientras que los tramos rectos indican el valor promedio del tramo.

Aunque el método selecciona el cuadro clave óptimo localmente, los "empalmes" (o puntos de quiebre) computados nos son óptimos. Para lograr un resultado óptimo (al menos localmente) es necesario integrar el cómputo. Lee y Kim [76] introdujeron un procedimiento iterativo el cual selecciona una cantidad predeterminada de cuadros clave y va reduciendo el error de reconstrucción de la toma y elige la posición de los cuadros clave y de los puntos de quiebre con el fin de obtener un óptimo local.

Li et al. [82] llegaron a una solución óptima usando programación dinámica y asumieron que $b_i = r_i$. Esta propuesta significa que cada cuadro clave representa a los cuadros que lo siguen. Los autores mencionados utilizan un algoritmo de búsqueda por bisección para encontrar el óptimo a partir de un nivel de fidelidad dado.

Simplificación de curva

El método de simplificación de curva se relaciona con la alternativa de agrupamiento (clusters). Cada cuadro en la secuencia de video se trata como un punto en un espacio de las fases multidimensional. En contraste con los métodos de agrupamiento, esta solución

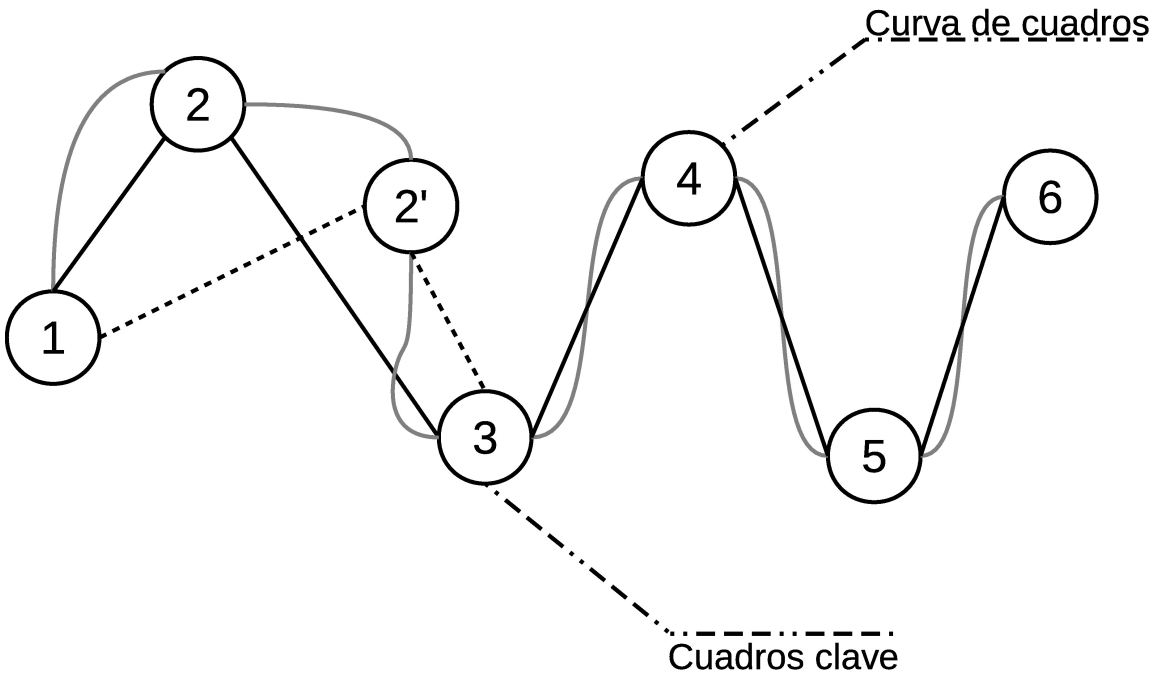

Figura 2.4 Simplificación de curva

busca encontrar un conjunto de puntos que cambien lo menos posible la trayectoria de una curva, y a su vez estén conectados en forma temporal. El resto de los puntos que no son seleccionados pueden removerse. De esa forma se produce la sumarización.

El algoritmo tiene similitudes con el método de error de reconstrucción de secuencia. La principal diferencia es que los puntos en la curva no están separados equitativamente según el orden de cuadros. Además, en este caso no es necesario el modelado explícito del error entre la curva final y la secuencia original. En el caso de simplificación de curva se utilizan algoritmos conocidos tales como la separación binaria de curvas [36] y la evolución discreta del contorno [73].

La Figura 2.4 muestra la lógica que sigue este método. En la figura se pueden ver 6 cuadros que son seleccionados como óptimos. Si el cuadro clave 2 es reemplazado por el 2', el nuevo conjunto de cuadros no aproxima mejor que la primera selección y por lo tanto es menos representativo de la curva original. El método de simplificación de curva divide el video en unidades de cuadros contiguos y presenta orden temporal. Entre las características no deseadas se pueden mencionar el alto costo computacional y además no se garantiza un resultado óptimo.

Eventos relevantes

Los métodos de sumarización descriptos anteriormente se focalizan en maximizar la extensión y el balance de la cobertura visual del conjunto de cuadros resultado. En contraste con los anteriores, los métodos basados en eventos relevantes intentan capturar los cuadros que son importantes desde el punto de vista semántico. La mayoría de los métodos clasificados bajo esta categoría asumen una correspondencia entre la "importancia" del cuadro con los patrones de movimiento alrededor del mismo. También tienen en cuenta las características del contenido tales como rostros humanos o complejidad espacial alta.

Liu et al. [86] propusieron un método que extrae cuadros clave a partir de los patrones de movimiento de la toma. Los autores desarrollaron un método de estimación de energía de movimiento percibida. En esta propuesta los segmentos de video son segmentados en subsegmentos de patrones de movimiento consecutivos en términos de aceleraciones y desaceleraciones. A partir de los segmentos elegidos, los autores seleccionan los cuadros clave cuando se pasa de aceleración a desaceleración o viceversa. Definen el pasaje como punto de cambio. La decisión se basa en que los puntos de cambio en velocidad son los más representativos. Además asumen que el movimiento entre aceleraciones y desaceleraciones pueden ser inferidos.

En un trabajo similar al anterior, Han y Kweon [57] extraen cuadros clave de una curvatura en dos dimensiones. La curvatura involucra el movimiento de la camara. Este trabajo aporta un mayor nivel de sofisticación debido a que incorpora no solo la magnitud sino la dirección del movimiento.

Dirfaux [37] propone un método que selecciona un solo cuadro clave de todo un video. La propuesta consta de dos pasos. El primer paso consiste en la selección de tomas basado en el largo, movimiento, actividad espacial y la probabilidad de inclusión de humanos. Luego que la toma más representativa ha sido seleccionada, se procede a la selección del cuadro clave. El autor sugiere elegir el cuadro cuando hay actividad con bajo movimiento, alta actividad espacial y alta probabilidad de incluir humanos.

Kang y Hua [69] observaron que existen similitudes entre los cuadros claves elegidos por usuarios humanos. En el trabajo propuesto por estos autores definen una serie de descriptores (calidad de cuadro) a partir de los cuales aprenden.

Algunas de las propuestas clasificadas dentro de esta categoría producen resultados aceptables. Sin embrago, no es posible aplicar este tipo de métodos para todos los casos. En general presentan soluciones para casos particulares. Además las soluciones propuestas se constuyen a partit de heurísticas basadas en observaciones empíricas. La complejidad y la irregularidad en los patrones de movimiento hacen que estas soluciones sean inefectivas

para problemas generales.

2.1.5. Métdos de visualización de cuadros clave

Una vez que los cuadros han sido extraídos, es fundamental presentarlos al usuario de una manera amigable. Es muy importante que el usuario pueda entender el contenido de una secuencia de video rápidamente. Este aspecto es crítico en el caso en el que es necesario navegar sobre una colección de videos. En la bibliografía pueden encontrarse dos formas de presentar cuadros claves que predominan sobre el resto [129]. Una de ellas es el **guión gráfico** (storyboard) y la otra es la **presentación dinámica** (dynamic slideshow).

El guión gráfico se construye como un conjunto de cuadros puestos como mosaicos uno al lado del otro. Tiene como principal ventaja el hecho de presentar la totalidad de información simultáneamente. Como factores negativos se puede mencionar la necesidad de espacio para poder visualizarlo y demanda un intenso movimiento en los ojos del usuario. Sin embargo, se ha demostrado usando estudios con usuarios [72] que sigue siendo el método preferido.

En el caso de presentación dinámica, el observador no tiene control sobre la frecuencia de visualización de los cuadros y puede causar problemas para la correcta compresión del significado y en contexto entre los cuadros sucesivos. En contraste, el usuario tiene la ventaja que los cuadros son presentados de a uno a la vez.

Cuando se extraen cuadros con múltiples niveles de fidelidad es posible presentarlos en un modo jerárquico [77, 120]. En el nivel superior cada cuadro representa la secuencia de video completa. En los niveles inferiores los cuadros representan las tomas de las cámaras. Esta forma de representar los cuadros clave asume que las imágenes tienen el mismo peso para la compresión del contenido en el video. Sin embargo no existe entre cuadros relaciones que puedan expresar la vinculación entre los mismos.

Existen diversas alternativas a las formas de visualización de cuadros las cuales intentan insertar semántica adicional. Para lograr la tarea es necesario un análisis estructural o encontrar las asociaciones semánticas entre los cuadros. Yeung y Leo [141] propusieron representar a cada escena en la secuencia de video por un póster. A partir de la extracción de cuadros clave por agrupamiento, es posible asignar un valor de dominancia y la duración total del grupo. Cada póster se compone combinando los cuadros en distintos tamaños siguiendo un patrón determinado. Los patrones pueden ser elegidos a partir del tipo de escena y los cuadros pueden superponerse dependiendo de los pontajes asignados.

Una visualizacion similar fue propuesta por Uchihashi et al. [131]. En este caso la propuesta es denominada Video Manga. La secuencia de video completa es considerada como una sola escena y no es necesaria la detección de cambio de toma. Video Manga

asigna diferentes tamaños de cuadros dependiendo de la importancia de los mismos. El algoritmo de patrones mantiene el orden temporal de los cuadros y no utiliza patrones predefinidos. En este caso el solapamiento de imágenes no es permitido.

Girgensohn [48] propuso un algoritmo similar al Video Manga con el agregado de un mejor ordenamiento de los cuadros. Los principales aportes consisten en bajar el costo computacional e incrementar la eficiencia en términos de espacio requerido.

Chiu et al. [27] propusieron un método de visualización llamado Vitral (stained glass). El algoritmo consta en detectar las regiones dominantes en términos de actividad para cada toma. En primer cuadro de cada una de las tomas es considerado el más representativo. A partir de la información calculada se construye un diagrama de Voronoi (o polígonos de Thiessen) en el cual cada una de las regiones se escala dependiendo de la relevancia asignada a la toma. El resultado final puede asociarse a un vital debido a las características morfológicas del diagrama. Los resultados de este método son prometedores aunque no es clara la validez de la representación. El principal problema radica en la definición y detección de las regiones dominantes.

Yeung y Leo [141] propusieron un algoritmo que utiliza semántica adicional la cual no se expresa en los guiones gráficos. El método consiste en construir un grafo de transición de escenas. Cada cuadro clave representa un grupo de tomas mientras que los vínculos entre cuadros sugieren la dinámica de la historia. Esta propuesta fue ampliada por Troung et al. [130]. En este caso se hizo foco en el desdoblamiento de los patrones de edición dentro de las escenas.

Cuando se selecciona un número limitado de cuadros clave para representar una secuencia es imposible poder capturar todos los detalles de la acción. Además es imposible poder reconstruir o entender las relaciones espaciales entre los cuadros tales como rotaciones o traslaciones. En los casos en los que es importante mostrar ese tipo de información se pueden encontrar los métodos de visualización tipo mosaico. Una imagen mosaico se sintetiza a partir de una secuencia y muestra todos los detalles estáticos de la escena. Pude decirse que es una visualización de cuadros clave donde todos los cuadros son seleccionados como clave.

En la literatura se puede encontrar la visualización de mosaico como fusión de imágenes (image sprite) [78], estáticos salientes (salient still) [127] o imagen panorámica (panoramic image) [125]. Los resultados de este tipo de método ha tenido éxito aceptable en algunas aplicaciones específicas [7]. Sin embrago, no es posible usarlo en todos los casos debido a que muestra defectos en términos de alineamiento de cuadros.

Los problemas exhibidos por el método de visualización de mosaico fueron solucionados parcialmente por algunos trabajos. Taniguchi et al. [125] solucionaron el

problema para los casos en que se detectan largos tramos de movimiento suave de cámara.

Otro problema que presenta el paradigma es que se focaliza en las tomas y no escala para una secuencia de video larga. Aner y Kender [7] propusieron un algoritmo para agrupar las imágenes mosaico para cada escena. En este caso, el mosaico que mejor mapea al promedio de un grupo se selecciona como el más representativo.

El mosaico por definición descarta los detalles dinámicos y resalta los aspectos estáticos de la escena. Una solución a este problema fue propuesta por Itani et al. [68] quienes construyen un mosaico sinopsis. Método similares fueron publicados por Pope et al. [109] y Pritch et al. [111].

2.2. Extracción de video condensado

Cronológicamente la extracción de video condensado es posterior a la extracción de cuadros clave. En general, las propuestas clasificadas bajo este tipo demandan un análisis de alto nivel de abstracción [129]. La técnica trivial para producir un video condensado es muestrear uniformemente el video original. El resultado es un resumen de largo fijo el cual depende del largo original.

Usando el muestreo del video original, es posible encontrar técnicas de sumarización que hacen muestreo uniforme o adaptativo [38, 101, 105, 106]. Este tipo de propuestas degradan la coherencia del video resultado. En algunos casos, causa desagrado a los usuarios debido a los saltos en el “hilo de la trama”. Además no sigue con la definición de video condensado propuesta en la Ecuación 2.2.

Las propuestas relevadas en este trabajo se focalizan en técnicas que conserven la frecuencia de cuadros original y que se focalicen en detectar los segmentos relevantes del video original. Algunos ejemplos son las propuestas que detectan cuadros clave y agregan los vecinos a los mismos [70, 102]. Sin embargo estas soluciones producen resultados que son difíciles de interpretar.

La Figura 2.5 presenta los atributos de las técnicas de sumarización de video condensado propuestas en la literatura. Como se puede observar, hay seis aspectos por los cuales las técnicas son clasificadas. Las categorías son el largo del resultado, el dominio de los datos, el proceso de generación, la preservación de la perspectiva, los mecanismos subyacentes y las características usadas.

Figura 2.5 Atributos de las técnicas de extracción de video condensado

2.2.1. Largo del condensado

Cuando se extrae un video condensado, el largo del mismo puede ser definido previamente (a priori) o puede ser desconocido antes de comenzar con la sumarización (a posteriori).

A priori

Las técnicas clasificada bajo la categoría *a priori* especifican el largo del video resultado o la relación l entre el largo del video resultado y el original. Se puede formalizar el resultado óptimo como la búsqueda de un conjuto de tramos $\mathbf{E}_1, \ldots, \mathbf{E}_k$ de manera tal que sean lo más cercano al original según la perspectiva ρ.

La Ecuación 2.18 expresa la formulación de una sumarización a priori óptima, donde $\mathbf{E}_i$ es el extracto i incluído en el resumen, $\odot$ es la operación de integración y $\mathscr{D}$ es la métrica de similitud.

En el caso que la perspectiva ρ esté dada y representa la cobertura de información, el problema se reduce a encontrar y combinar segmentos del video original que representen mejor la información fuente. Dos ejemplos que producen sumarizaciones de video a partir de un largo especificado por el usuario son los propuestos por He et al. [60] y Ma et al. [93].

$$\mathscr{K} = arg \min_{\mathbf{E}_i \subset \mathbf{V}} \left\{ \mathscr{D}(\mathbf{E}_{i_1} \odot \mathbf{E}_{i_2} \odot \ldots \odot \mathbf{E}_{i_k}, \mathbf{V}, \rho) | \sum_{i=1}^{\mathbf{k}} |\mathbf{E}_i| = l \right\} \tag{2.18}$$

A posteriori

En este caso el largo del resumen es determinado por el contenido del video original. Formalmente, el video condensado se define como se expresa en la Ecuación 2.19, donde ε es el valor de la tolerancia. Dada una perspectiva ρ, el problema de generar una sumarización se reduce a encontrar el video más corto tal que permite al usuario observar los eventos relevantes. Dos ejemplos de trabajos que se clasifican bajo esta categoría son la propuesta de Babaguchi et al. [12] y la de Sundaram y Chen [122].

$$\mathscr{K} = arg \min_{\mathbf{E}_i} \left\{ \sum_{i=1}^{\mathbf{k}} |\mathbf{E}_i| | \mathscr{D}(\mathbf{E}_{i_1} \odot \mathbf{E}_{i_2} \odot \ldots \odot \mathbf{E}_{i_k}, \mathbf{V}, \rho) < \varepsilon \right\} \tag{2.19}$$

2.2.2. Dominios específicos de videos

La mayoría de las técnicas de generación de video condensado propuestas solo son válidas para casos de uso particulares. Dependiendo del tipo de contenido del video, hay

casos en los cuales es más simple sumarizar la información. Uno de los ejemplos de dominios específicos son los deportes [8, 11, 12, 30, 31, 58, 137].

Otro dominio sobre el cual se pueden encontrar propuestas de sumarización son las noticias [28, 119]. También hay algorimos de sumarización para documentales [142], películas [59, 85, 108, 122, 123], videos caseros [84, 142, 146] y presentaciones de oradores [60].

Se define una técnica de sumarización como genérica si es posible ser aplicada en cualquier dominio de video. Por ejemplo, si un algoritmo utiliza los efectos del audio tales como los aplausos, los gritos y las risas para detectar eventos destacados en deportes [20], puede ser usado casos tales como videos de entretenimiento. Algunas propuesta basadas en eliminación de redundancia tienen características de sumarización genérica [32, 50].

2.2.3. Proceso de generación de video

A partir de los trabajos publicados, es posible identificar 5 pasos que engloban los métodos de generación de video condensado. Las etapas son segmentación de tomas, selección de tomas, acortado de tomas, ensamble de tomas e integración multimodal. El término toma se refiere a un segmento del video el cual es independiente de la edición o escenas presentes en la trama del video.

Segmentación de tomas

En este caso la generación del video condensado se basa en usar segmentos de video extraídos de la fuente original. Generalmente, los segmentos se consideran independientes entre sí. En los casos de sumarizaciones basadas en fuentes de texto tales como subtítulos, una forma de detectar la transición de segmento es usar las pausas entre las conversaciones [126].

En el caso de las técnicas basadas en eventos relevantes [8, 114], la detección de segmentos se reduce en marcar el video original en segmentos con evento y los que no los contienen. De esta forma se produce una división natural de los datos de entrada.

En otros casos, la regla general es la detección de cambios en el movimiento dominante de la imagen [107]. El concepto es similar a la factorización de la matriz de auto-similitud propuesto por Cooper y Foote [32].

Selección de tomas

El paso siguiente consiste en seleccionar los segmentos a ser incluídos en el resumen. Es importante aclarar que a esta altura no se tiene en cuenta el ordenamiento de los extractos.

La cobertura del video condensado va a depender de los segmentos seleccionados y además va a impactar en la coherencia del resultado.

Una opción usada es agrupar las tomas y luego seleccionar la toma que se extiende por mayor tiempo dentro del grupo [50]. En el caso de los métodos orientados a eventos, se seleccionan los segmentos en los cuales se producen los mayores cambios [8, 107]. Es posible tener más de un factor a la vez. En el caso de Babaguchi et al. [12], se crea una tabla que vincula la importancia de la toma y el tipo de segmento. Luego se eligen los segmentos a partir del recorrido de la tabla.

Ngo et al. [100] proponen una método de selección iterativo. Los segmentos se dividen en distintos grupos. Para cada grupo se computa un índice de movimiento y probabilidad de ocurrencia. Basado en estos dos parámetros, se eliminan los elementos de los grupos.

Acortado de tomas

Un correcto acortado de tomas consiste en producir un resultado conciso y sin pérdida considerable de la información. La tarea no es simple debido a que se pueden crear puntos de corte inapropiado o se pueden acelerar eventos que pueden confundir al espectador.

La forma más simple de resolver el problema es seleccionar un porcentaje definido de la toma. Algunas propuestas definen un largo de segmento fijo [50, 51, 79]. He et al. toman un tiempo fijo luego de detectar un cambio de transparencia en el caso de sumarización de presentaciones [60].

Existen otras propuestas más elaboradas. Sundaram y Chang [122] modelan la distribución de tiempo de comprensión del espectador a partir de la complejidad visual de los segmentos. De esa forma una toma puede ser acortada hasta el límite de la compresión. Ma et al. [93] modelan una curva de atención de la cual extraen los cuadros clave. Luego acortan la toma usando los segmentos vecinos a los cuadros.

En el trabajo de Zhao et al. [146] proponen clasificar los segmentos por importancia semántica del audio. Algunas de las clases pueden ser conversaciones, risas o música. Dependiendo de la importancia que se desee para cada una de las categorías, es posible acortar los segmentos.

Otra forma de acortar un segmento es eliminar cuadros. El trabajo de Li et al. sigue esta lógica [81]. En este caso se modifica el audio para que pueda ser coherente con las imágenes. Físicamente hay limitaciones para esta estrategia debido a que el máximo acortamiento presenta dificultades a los observadores.

Integración multimodal y ensamble de tomas

Los fragmentos a los cuales se ha hecho referencia hasta ahora son usualmente monomodales debido a que hacen foco a un componente del video tal como audio, imagen o texto. En este paso de generación del video condensado se produce la combinación de los diversos modos, se acondicionan los bordes de los segmentos y se ensambla el resultado final. Si la integración multimodal se realiza correctamente, es posible mejorar la coherencia, la cobertura y el contexto del condensado final.

Dependiendo de la forma en que los videos condensados son construídos es posible clasificarlos en dos grupos. El resultado puede ser sincrónico o asincrónico. En el primer caso, el audio y el video se sincronizan siguiendo el orden del video original. Este tipo de sumarización es usada para los casos de películas o programas de televisión debido a que lo que se escucha está relacionado con lo que se ve. En los casos en los que se toma como referencia para la sincronización al audio original, se dice que el resumen está basado en el audio (centrado en audio). El mismo concepto puede ser aplicado con el video (centrado en video) o el texto (centrado en texto).

Cuando el video condensado se genera a partir de un solo modo, se pueden utilizar operaciones lógicas estándar (OR, AND) para producir la compilación [41]. En el caso de Agnihotri et al. [1] se utiliza la operación OR para suavizar el resultado final. La sincronización se asegura seleccionando segmentos de los distintos modos siguiendo una linea de tiempo común.

En los casos en que el video original es un documental o noticias, se puede usar una integración asincrónica. El compilado final también puede ser centrado en video, centrado en audio o centrado en texto. A diferencia del caso sincrónico, el condensado final se construye seleccionando los segmentos que tienen mayor importancia semántica. Por ejemplo, el trabajo de Smith y Kanade [119] parte de un resumen basado en audio y le agrega información a partir de elementos visuales tales como movimientos de cámara y rostros humanos. En los casos de noticias, es recomendable mantener la coherencia del audio para preservar la comprensión y componer el resumen final con el canal visual acorde [83]. En contraste, Gong y Liu [50] utilizan un algoritmo de alineamiento bipartito basado en grafos para integrar segmentos centrados audio y en video.

La forma más directa de integrar fragmentos en un video condensado es unir las piezas en orden temporal. De hecho esta solución es la más usada en la literatura. Existen algunas excepciones que dividen los fragmentos en clases y compilan un video que no sigue la cronología original [41, 84]. Sin embargo, la mayoría de las propuestas de compilación siguen la cronología original. Los aportes se focalizan en suavizar las transiciones entre fragmentos [104, 115].

2.2.4. Preservación de la perspectiva

Cuando se genera un video condensado se debe decidir qué perspectiva se va a preservar. Dependiendo de la perspectiva elegida, se producen distintos tipos de sumarización. Una perspectiva en particular no siempre es la mejor elección para todos los casos de sumarización posibles. En la literatura se pueden distinguir tres clases de perspectivas. Las categorías son cobertura de la información, eventos interesantes y consulta de contexto y personalización.

Cobertura de la información

La perspectiva orientada a la cobertura de la información se focaliza en proveer una impresión de la información contenida en el video original eliminando la redundancia. Además es importante priorizar la comprensión del usuario de la idea general del video.

La aplicación principal de esta perspectia es en videos que son diseñados para comunicar información. Algunos ejemplos son videos de noticias o videos instructivos. En estos casos los usuarios están interesados en comprender el contenido general sin prestarle atención particular a un tramo específico. Algunos ejemplos relevantes en la literatura son Lienhart [85], He et al. [60], Hanjalic et al. [59], Sundaram y Chang [122], Ma et al. [93], Gong et al. [50, 51]

Eventos interesantes

El concepto de la perspectiva orientada a eventos interesantes consiste en capturar los momentos de mayor interés en un video. En la bibliografía se lo puede encontrar como “momentos destacados” (highlights) [59]. La aplicación de este concepto se usa principalmente en videos deportivos debido a que es simple definir cuándo un evento es sobresaliente. A partir de tener una definición clara de un evento interesante, es posible proceder a realizar el resumen.

Los conceptos usados como relevantes se pueden definir de varias formas. En el caso de deportes, Assfalg et al. usan la conversión de un gol en los partidos de fútbol [9]. Algunos casos consideran relevantes las acciones realizadas cerca del área cercana a un arco de fútbol [134]. Xiong et al. detectan reacciones anormales de la audiencia tales como aplausos o gritos [137]. También han sido usados como eventos relevantes las acciones del productor del video, por ejemplo la frecuencia de cambio de cámara y el uso de secuencia repetida [104].

Exiten ejemplos de detección de eventos destacado en dominios más generales. Bagga et al. [14] ponen atención sobre la repetición de títulos en programas de noticias. Ma et al.

[93] detectan tomas con altos niveles de movimiento y con cantidad de colores por encima de lo normal. También hay ejemplos de uso de patrones inusuales o poco frecuentes [3, 94, 113, 137, 142].

Consulta de contexto y personalización

En este tipo de perspectiva los usuarios especifican los detalles de interés. En el caso de Christel et al. [29] los usuarios ingresan una consulta y a partir de ese punto se genera el video condensado. En este caso el sistema se implementa teniendo un cómputo previo de los pesos de las preferencias del usuario sobre un conjunto de características disponibles para la perspectiva.

En el trabajo de Zhao et al. [146] se proponen pesos para eventos temporales tales como conversaciones, risas, música, aplausos, gritos, movimiento y ruido general. En los casos de videos deportivos, se pueden encontrar propuestas que tienen en cuenta jugadores populares o equipos predilectos a partir de los cuales se puede crear resúmenes [11]. En el caso de Agnihotri et al. [2] se intenta producir sumarizaciones que tienen en cuenta el perfil del usuario. Algunos de los factores usados son el sexo de la persona o la edad.

2.2.5. Mecanismos subyacentes

Los mecanismos para generar el video condensado es altamente dependiente de la perspectiva que se desea preservar. Por ejemplo, los métodos basados en eventos relevantes no van a ser eficientes en términos de la cobertura máxima. En la literatura se pueden encontrar tres grupos de mecanismos para la generación del video condensado. Los nombres son eliminación de redundancia, detección de evento y curva de formulación de condensado.

Eliminación de redundancia

En este caso se entiende por redundancia a las partes del video que contienen información conocida. El acortado de segmentos es fundamental para lograr resultados satisfactorios [122, 123]. Cooper y Foote [32] remueven redundancia en los segmentos seleccionando solo cuadros contiguos que maximizan la similaridad promedio de la secuencia entera. Otra forma de elimiar la redundancia es seleccionar los segmentos luego de agruparlos [50, 51, 59]. Lu et al. [90, 91] remueven redundancia usando un método de correlación mínima de cuadros clave. Esta propuesta fue explicada en 2.1.4.

Detección de evento

En los casos en que el video condensado se focaliza en la detección de eventos relevantes, es importante detectar el evento y además los límites en donde ocurre. Chan et at. [22] utilizan Modelo oculto de Markov para detectar eventos en partidos de baseball. Dagtas y Abdel-Mottaleb [34] detectan eventos destacados en videos deportivos usando dos estrategias. Una de ellas es detectar en el audio palabras claves dentro de segmentos de alta energía de sonido. La segunda estrategia es detectar transiciones de cuadros entre sectores centrales de juego y sectores de marcado de puntos o goles. Babaguchi [12] detecta eventos de marcado de puntos en fútbol americano analizando el subtitulado de los partidos. Peyrad y Bouthemy [107] detectan eventos en videos deportivos usando solo características de movimiento. En contraste, Assfalg et al., [9] detectan eventos en partídos de fútbol usando modelos lógico-temporales. En este caso se propone un conjunto de modelos que relacionan la zona de juego, el movimiento del balón y la posición de los jugadores con el evento detectado.

Una alternativa distinta a detectar eventos específicos en eventos deportivos, es la detección de actividad infrecuente en el audio, el video o el movimiento. En el caso de una repetición de secuencia, es posible observarla en cámara lenta. La mayoría de los trabajos que detectan la repetición de secuencia se basan en las secuencias de cámara lenta [52, 52, 102, 104, 128]. Xiong et al. hacen uso del audio para detectar eventos relevantes [137, 138]. Rui et al. [114] detectan la voz del relator con el fin de capturar los “supuestos” momentos destacados.

Curva de formulación de condensado

En el caso de la curva de formulación de condensado se procede al cómputo de un puntaje que se asocia directamente con la base temporal. La métrica representa la probabilidad de inlcuir el tramo de video en el condensado con respecto a la prespectiva ρ. Por ejemplo, si el usuario quiere ver eventos destacados, la métrica debe reflejar el nivel de interés en función de la base temporal. Usualmente la base temporal tiene una unidad base y depende de un solo modo. En el caso de sumarizaciones visuales, la base temporal son los cuadros. En los casos en los que el video condensado se construye a partir de los textos, la base temporal suele ser la palabra [126].

La Figura 2.6 muestra un ejemplo de construcción de un video condensado a partir de un cómputo de la curva de perspectiva. En este caso los segementos se determinan a partir de un valor de umbral en la curva de la perspectiva. Una vez que los segmentos han sido determinados, el resumen final se construye uniendo los extractos seleccionados. En el caso

en el que la unidad de base temporal es el cuadro o un segmento de largo fijo, es simple construir un video condensado de tiempo fijo eligiendo los tramos de mayor valor de ρ hasta llegar a satisfacer la condición definida [93, 137].

Si la unidad de tiempo está basada en tiempo, pueden usarse los cuadros o los segmentos con un tiempo fijo como unidad. En el caso de tener la restricción de un resumen con tiempo limitado, se puede crear un resultado con los tramos de mayor puntuación en la curva hasta que se completa el tiempo esperado. El mismo problema es más complejo de resolver en el caso de que las unidades de base tienen duración temporal variable. Una solución propuesta consiste en ordenar los segmentos teniendo en cuenta la probabilidad de puntuación [81, 135]. Luego se usa un algoritmo voraz (greedy) para selccionar los segmentos [126].

En este tipo de método de sumarización, la selección de la unidad de base juega un rol fundamental. Es posible seleccionar unidades para resolver problemas específicos. En el caso de videos deportivos, la puntuación puede ser asociada al tipo de evento, el volumen del audio, la frecuencia del relato o los aplausos. En el caso de que la unidad se base en texto, el planteo cambia completamente y se aplican técnicas de análisis de texto.

En el caso de la creación de la curva de puntuación, existen varias propuestas interesantes. Hanjalic [58] modela el nivel de interés en el video usando el nivel de movimiento, frecuencia de corte de escenas y nivel de energía en el audio. En el caso de Lu et al. [92], se le asigna a cada toma un valor de importancia el cual se computa a partir de la descripción basada en anotaciones y el principio de reafirmación mutua. Este principio propone que un término importante debe contener descripciones importantes y viceversa. Otra forma de determinar los momentos relevantes es observar la interacción del usuario con la secuencia [3, 94, 99, 142].

La figura 2.6 muestra un ejemplo de curva de formulación de condensado. En la parte superior se puede observar la curva de perspectiva. La línea recta representa el umbral elegido. Cuando la curva de perspectiva supera el umbral, el tramo de video original es agregado al resumen. La salida del algoritmo se compone de la sumatoria de tramos seleccionados. En la figura se indican los tramos elegidos con los números 1, 2 y 3.

En los casos en que la unidad base es el cuadro, un segmento de largo fijo o una toma, la generación de un video condensado tal como se muestra en la Figura 2.6 no asegura coherencia. Tampoco se puede asegurar una covertura del contenido balanceada. Además puede contener segmentos muy cortos y segmentos similares en términos del contenido. Lu et al. [92] proponen una solución encontrando el conjunto mínimo de segmentos que maximizan la puntuación de importancia general. De una forma similar, Mei et al. [95] proponen una solución para videos caseros estimando un curva a partir de un conjunto de métricas las cuales incluyen la inestabilidad, el brillo, la orientación y el esfumado entre

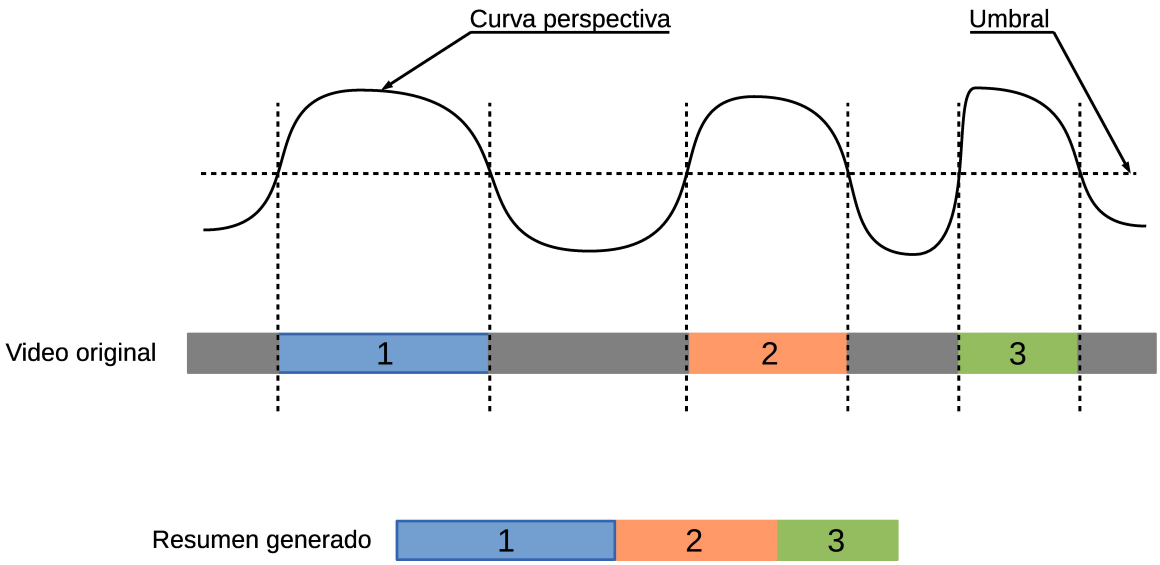

Figura 2.6 Curva resumen

otras. En vez de usar un umbral para la curva de perspectiva, seleccionan tomas uniformemente distribuídas formulando el problema como una optimización.

2.2.6. Características usadas

Otra forma de clasificar los métodos de sumarización es a partir de las característica (features) usadas. Las clasificaciones de las cacterísticas usadas consisten en visuales, de texto, audio, moviminento, operaciones de cámara y semánticas.

Visuales

Puede encontrarse una gran diversidad de características visuales usadas en la bibliografía. En el caso de videos deportivos se usan colores dominantes, bordes y texturas ya que son determinantes para identificar las vistas de las cámaras y las posiciones dentro del juego. De hecho son usadas par inferir los momentos destacados del juego [8, 9, 22, 24, 56, 116].

En los casos en que es necesario detectar similitud de cuadros o tomas, son usados los histogramas de colores. En particular son muy usados en los métodos de condensación de video que eliminan redundancia [50, 51, 100].

Existen casos particular en el uso de características visuales. Ma et al. [93] hacen uso de contraste de colores, contraste de intensidades y contraste de orientaciones para modelar la atención del humano a una imagen. Pfeiffer et al. [108] extraen escenas de alto contraste

para ser incluídas en el resumen de una película.

Texto

En los casos en los que se dispone de texto, se puede hacer uso del mismo para sumarizar un video. Esta característica puede tener un gran poder de abstracción. La importancia semántica puede ser más fácil de extraer que en el caso de usar solo el aspecto visual. El texto puede provenir de varias fuentes. Entre las más conocidas pueden citarse las que vienen dentro del video [84, 85, 108], los subtítulos que acompañanan a las imágenes [34, 56], el reconocimiento de voz [8, 50, 119] y comentarios externos [11].

Los datos de texto se usan generalmetne detectando palabras clave. En el caso de deportes puede usarse la parabra "gol" como clave [12]. Las palabras pueden ser especificadas manualmente o a partir de un conjunto de datos de entrenamiento [11, 12, 34].

Audio

Este tipo de características se refiere a el uso de primitivas tales como la energía, la envolvente de la señal o la detección de cierto patrón en el espectro. Debido a los beneficios computacionales que presenta, se hace uso de características brindadas por los compresores de audio tales como factor de escala [20, 34, 56, 93].

Movimiento

El nivel de movimiento en el video es asociado normalmente con el interés del segmento. Puede ser usado para detectar escenas de acción en películas [108] o para detectar una jugada interesante en un envento deportivo [58]. También ha sido usado para detectar acciones relevante en videos de cocina.

Otro aspecto importante en el cual ha sido usado es en detectar los momentos en los cuales la atención del usuario es máxima [93, 100]. Un patrón de movimiento que se extiende por una secuencia de video es útil para detectar eventos relevantes desde el punto de vista semántico. Un ejemplo de eso es la detección del video en cámara lenta [104].

Operaciones de cámara

En algunos dominios en particular tales como los eventos deportivos, los movimientos de cámara están relacionados con la "trama de la histora". Dependiendo del deporte que se esté analizando se pueden encontrar patrones que permitan sumarizar un video [31, 56].

Incluso se pueden encontrar propuestas en las cuales el movimiento de la cámara se convina con un modelo de atención del usuario [93].

Semánticas

Las características semánticas hacen referencia a particularidades del video que son extraídas de forma independiente. Un ejemplo de esto es el propuesto por Lienhart [85]. En su trabajo detecta rostros humanos para clasificar escenas en películas que son de diálogos. Ma et al. [93] modela la atención de un humano a partir de rostros y los tamaños relativos dentro de las imágenes. Otros ejemplos de características semánticas incluyen la información de acceso a un video en particular por parte de los usuarios [142].

2.3. Formato de video usado

El video puede venir en más de un formato. Debido a la gran candidad de infrormación que contiene el video, usualmente se lo almacena en formato comprimido. La comprensión de video permite reducir significativamente el espacio utilizado aunque obviamente demanda una compresión previa. Los métodos de sumarización de video pueden clasificarse por el formato en el cual trabajan. En esta sección se clasifican los métodos formato crudo o comprimido.

La mayoría de los métodos de sumarización propuestos en la literatura trabajan con video sin comprimir. El video sin comprimir permite acceder a toda la información de cada uno de los cuadros. En consecuencia, permite manipular la totalidad de la información disponible en la secuencia. Esta es la principal ventaja del enfoque.

En contraste, tabajar con el video comprimido solo permite manipular los parámetros definidos por el método de compresión. En este caso ya no se dispone de la totalidad de la información y se reduce el espacio para aplicar alternativas de procesamiento. Sin embargo presenta la gran ventaja que permite la operación en línea del video. Por lo tanto, se puede decir como regla general que trabajar con video comprimido demanda menos recursos de cómputo y permite resúmenes en línea.

2.3.1. Sin comprimir o crudo

Una de las propuestas más relevantes que operan con video sin comprimir es la de Kleban et al. [71]. En este trabajo se propone el uso de características generales para realizar una fusión de las mismas. Inicialmente, se produce un submuestreo del video para reducir la redundancia de la información. Del subconjunto resultante, se extraen cinco

características globales de cada uno de los cuadros. Luego se selecciona el peso óptimo de las características usando un algoritmo de descenso de gradiente que maximiza la covertura de los datos de entrenamiento. Finalmente, la sumarización se obtiene usando un algoritmo de agrupamiento k-means para identificar la información más relevante.

Pan et al. [103] presentaron un método de sumarización basado en la detección de cambio de toma. La propuesta consiste en la ejecución de tres pasos. Inicialmente se detecta el cambio de toma comparando los histogramas de colores. El segundo paso consiste en remover las redundacias usando un algoritmo de agrupamiento para clasificar las tomas. Luegro del agrupamiento, seleccionan una toma por grupo. El resultado final se construye agregando las tomas seleccionadas en el resumen basandose en la importancia de cada una de ellas.

Le y Satoh [74] popusieron un método de sumarización basado en agrupamiento. Incialmente agrupan cuadros consecutivos que son similares en fragmentos usando un algoritmo de detección de toma. Luego se agrupan los fragmentos en grupos usando el algoritmo GreedyRSC [62]. Finalmente construyen la sumarización seleccionando los fragmentos más largos.

Putpuek et al. [112] presentaron una propuesta similar a la de Le y Satoh [74]. La entrada de video es dividida en tomas usando histogramas de colores. En este caso también se usa el algoritmo GreedyRSC para la detección de toma. El prinicial aporte del trabajo es la fusión de tomas adyacentes y por consiguiente se logra reducir la redundancia. La sumarización final se logra eligiendo las tomas con mayor movimiento en cada uno de los grupos.

Bredin et al. [19] usaron PCA (Principal Component Analysis) para sumarizar video. La propuesta se compone de tres pasos. Inicialmente se detecta el cambio de toma usando un umbral adaptativo para la distancia de histogramas de colores en cuandros consecutivos. Luego se aplica PCA sobre una matriz de características usando los histogramas de colores. Finalmente se crea un mapa de contenido usando los componentes más representativos. Cada toma se representa con una “firma” dentro del mapa de contenido. El resumen final se compone de las tomas más relevantes.

Chasanis et al. [23] hicieron uso del agrupado espectral y del alineamiento de secuencia. Primero se segmenta la secuencia de video en tomas comparando histogramas de cuadros consecutivos. Luego se usa un algoritmo de agrupamiento espectral para seleccionar el cuadro más relevante en la toma. Una vez seleccionados los cuadros, se agrupan las tomas similares usando un algoritmo de alineamiento de secuencia. La toma más larga del grupo se selecciona como la más significativa. El resultado final se produce concatenando cuadros alrededor de los cuadros claves de cada una de las tomas.

Besiris et al. [18] propusieron un método de sumarización basado en la conectividad de

un grafo y el agrupamiento. El método se compone de tres pasos. Primero se submuestrea la secuencia y se extrae el histograma de color HSV como característica. El segundo paso consiste en agrupar los cuadros usando un algoritmo basado en grafos. Finalmente se extraen los cuadros clave usando los centroides de cada una de las clasificaciones.

Guan et al. [53] usaron características globales y locales para crear un método de sumarización. La propuesta consiste en tres etapas. Inicialmente representan cada cuadro con una característica global llamada CEED (Color and Edge Directivity Descriptor). Luego de la representación, realizan un agrupamiento usando el algoritmo k-means. La clasificación les permite dividir el video completo en escenas individuales. Como segunda etapa, se seleccionan cuadros clave dentro de las clases usando la característica local SIFT (Scale-invariant feature transform) [89]. Crean un espacio global de puntos característicos detectados en los cuadros. Luego de la construcción del espacio, seleccionan un conjunto de cuadros clave que cubre el espacio a un porcentaje dado. Finalmente el resumen se crea reordenando los cuadros clave en orden cronológico.

En un trabajo posterior, Guan et al. [54] presentan un nuevo método de sumarización. En este caso también proponen tres pasos pero solo usan características locales. Primero computan puntos característicos para cada cuadro. Luego crean el espacio de puntos. El tercer paso es la selección de cuadros clave basados en los criterios de covertura y redundancia.

Murdur et al. [98] desarrollaron un método basado en la Triangulación de Delaunay (DT). La triangulación se aplica a agrupar los cuadros de video. Inicialmente se sub-muestrean los cuadros del video con el fin de disminuir la cantidad de información a procesar. Cada cuadro se reprenta como un histograma de color en el espacio de colores HSV. El histograma se representa por un vector fila y los vectores se concatenan en una matríz. Luego, se aplica PCA (Principal Component Analysis) para reducir el tamaño de la matríz. A partir de este punto, se construye un diagrama Delaunay. Con los bordes del diagrama, se obtienen los grupos (o cluster). Finalmente por cada cluster se selecciona un cuadro clave el cual es el más cercano al centroide del grupo.

Furini et al. [45] presentaron VISTO (Visual STOryboard for web video browsing). En este trabajo se compone de tres pasos. Inicialmente representa cada cuadro por un histograma de colores en el espacio HSV. Luego seleccionan los cuadros más significativos usando un algoritmo de clasificación rápida. Finalmete se hace un procesamiento final al conjunto de cuadros clave con el fin de eliminar redundancias.

En un trabajo posterior, Furini et al. presentarion SITMO (STIll and MOving Video Storyboard for the Web Scenario) [46]. La propuesta es similar a VISTO pero con algunas mejoras. En SITMO extraen un vector de características de color de 256 elementos.

Además el algoritmo de clasificación hace uso de una desigualdad triangular para eliminar redundancias.

Fontes de Avila et al. presentaton VSUMM [35]. El método genera cuadros clave y consta de cinco pasos. Inicialmente se sub muestrea la entrada de video para bajar la complejidad del procesamiento. Luego se representa cada cuadro con un histograma de color de 16 clases en el espacio HSV. El tercer paso consiste en clasificar los cuadros usando k-means. Por cada clase, se extrae un cuadro clave elimnando redundancia. Finalmente se compone el resultado final ordenando los cuadros en orden cronológico.

2.3.2. Comprimido

Chew y Kankanhalli [25] presentaron un método de sumarización que extiende otro ya existente [121]. Inicialmente convierten cada cuadro del video en un vector de 64 dimensiones el cual contiene características globales. Luego, la secuencia de video es segmentada uniformemente en trozos de tamaño fijo. Para cada segmento se define una unidad de cambio basada en la similitud de los cuadros del comienzo y del fin del segmento. A partir de la unidad de cambio, usan un valor de umbral predefinido para ordenar en dos grupos a los segmentos. Los grupos se denominan de poco cambio y de mucho cambio. Se consideran relevantes todos los cuadros de los segmentos de mucho cambio. Además se toman solo el priero y el último cuadro de los segmentos de poco cambio. Finalmente los cuadros se organizan en un nuevo video usando un algoritmo que itera recursivamente hasta llegar al largo deseado para el resultado.

Peker y Divakaran [105] propusieron un método de sumarización basado en adaptar la velocidad del video relativa al movimiento. El método acelera el video si el nivel de movimiento es bajo y baja la velocidad en caso de que la cantidad de información visual se incrementa. Todo el método tiene como componente princial las características espacio-temporal de la transformada discreta coseno (DCT) usada en los métodos de compresión.

Benini et al. [17] publicaron un método de sumarización que usa la información derivada de la caracterización de la dinámica de video en las tomas. Usando las características de los decodificadores de video, computan descriptores de movimiento que estiman la contribución de cada toma. Luego, hacen uso de un modelo oculto de Markov (Hidden Markov Models) para modelar la secuencia. Finalmente, la sumarización se genera como una scuencia de observaciones, donde la probabilidad más alta se asigna a las tomas con más movimiento.

Herranz y Martínez [61] presentaron un método basado en agrupamiento y posicionamiento. Inicialmente se extrae un descriptor de color de la imagen de cada cuadro tipo I usado en compresores tipo MPEG. Luego particionan la secuencia de entrada en tomas comparando las diferencias entre los vectores de características. Una vez que tienen

las tomas, hacen uso de un algoritmo de agrupamiento. La sumarización final se construye usando un procedimiento de posicionamiento interactivo, en el cual los grupos son posicionados y seleccionados incrementalmente.

Almeida et al. [4–6] presentaron una técnia de sumarización de video que opera adecuadamente en línea. La prouesta se basa en el estándar de compresión MPEG. Siguen tres pasos para producir la sumarización, los cuales son extracción de características, selección de contanido y filtrado de ruido. Por cada cuadro se obtienen caracterísicas construyendo un histograma de colores a partir del espacio de colores HSV. Usando la métrica ZNCC (Zero-mean Normalized Cross Correlation) determinan la distancia de dos cuadros. Por cada uno de los cuadros se computa el nivel de diferencias y en caso de que el valor supere un umbral dado, se determina el cuadro como relevante. Luego de la selección del contenido, computan el histograma de colores y el histograma de orientación de gradiente por cada cuadro relevante. Finalmente, el ruido es eliminado teniendo en cuenta el caso en que los dos histogramas tienen una varianza normalizada que supera un valor de umbral predeterminado.

2.4. Demandas computacionales de la sumarización

Además de los resutados producidos por un método de sumarización, es importante conocer las demandas computacionales. Solo en unos pocos trabajos se publica este tipo de información. Se ha relevado en la literatura algunos trabajos que mencionan el costo computacional y los requerimientos de espacio de cada algoritmo popuesto.

Entre los trabajos que publican las demandas computacioanes se pueden mencionar el presentado por Fontes de Avila et al. [35] y el de Almeida et al. [4–6]. En la sección 5.1 se detalla la información relevada y se compara con la propuesta presentada en el presente trabajo.

2.5. Métodos de evaluación

Los métodos de evaluación de la sumarización de video son fundamentales para poder comparar el avance en la materia. Es fundamental poder tener un “banco de pruebas” común. Sin embargo, la mayoría de los autores en la literatura coinciden que no se cuenta con un marco de comparación consistente y correcto desde el punto de vista metodológico [80, 129]. La mayoría de los trabajos proponen su propio método de evaluación. Además, en la mayoría de los trabajos tampoco se menciona la complejidad computacional de la propuesta o el desempeño.

Una de las razones por la cual no se cuenta con un criterio de evaluación común se debe a la dificultad de la tarea. A diferencia de otras áreas de conocimiento no es simple definir conceptos uniformes respecto a el reconocimiento de un objeto o a la abstracción del una escena. De hecho es difícil para los humanos cuantificar la caliad de una sumarización de video.

Otro factor que complica un poco más el escenario es la subjetividad subyacente. Un evento puede ser relevante dependiendo del contexto. Por ejemplo, una pelota de fútbol puede ser común dento de un video deportivo pero totalmente atípico en una película romántica.

De la literatura se puede clasificar a los métodos de evaluación en tres categorías. Las clases son descripción de resultado, métricas objetivas y estudios con usuarios. A continuación se describe cada una de ellas.

2.5.1. Descripción del resultado

La descripción de resultado es la forma más simple y además es la más usada en la literatura. No requiere ninguna comparación con otras técnicas. Usualmente, la técnica propuesta es aplicada a unas pocas secuencias de video y la sumarización generada es descripta por los autores. Como regla general las conclusiones se restringen a evaluar si es aceptable o no el resultado. Es común encontrar una discusión de cómo los parámetros o la dinámica visual afectan a los cuadros clave extraídos [53, 59, 145]. En algunos trabajos se intenta describir las ventajas del método propuesto con respecto a otros existentes [132].

Varios autores concluyen que este tipo de método de evaluación no es adecuado [129]. Existen varios aspectos que son criticables. El primero es la gran subjetividad implícita. Otro detalle muy importante es que solo se usan unos pocos videos para la comparación. Además no permite tener fundamentos metodológicos claros para usarlo como referencia para futuros trabajos o comparar con otros métodos.

2.5.2. Métricas objetivas

En el caso de los métodos de extracción de cuadros clave, puede encontrarse una función de fidelidad la cual es computada a partir de los cuadros extraídos y la secuencia de video general. Esta métrica es usada para comparar los cuadros extraídos por distintos métodos de sumarización. También es usada para evaluar cómo los parámetros de un método en particular afectan el resultado.

Auque esta métrica es llamada “objetiva”, tiene una tendencia hacia una perspectiva particular o hacia alguna técnica de sumarización. No existe una justifucación general que

afirme que la métrica se correspende con el criterio de un observador humano para un conjunto de datos dado. Matemáticamente, este tipo de métrica tiene un formato similar a los conceptos de el conjuto de cuadros clave óptimos prouestos en las Ecuaciones 2.3 y 2.4. Un ejemplo de uso puede encontrarse en el trabajo de [88], donde comparan usando la métrica SRE con Zhang et al. [144, 145].

En los casos de métodos de sumarización que producen videos condensados, se puede observar la aplicación de las métricas de precisión y sensibilidad (precision/recall). En particular se aplica a los casos en que se basan en la detección de eventos relevantes. Este tipo de sumarización permite reconocer cuáles son las partes del video que son verdaderos positivos y por lo tanto es directa la creación de métricas. Algunos ejemplos son los de Chang et al. [22], Xiong et al. [137] y Ariki et al. [8].

2.5.3. Estudios con usuarios

Estos estudio involucran usuarios independientes para evaluar el resultado de una sumarización. Probablemente son los más útiles y realísticos para la evaluación. Tienen la gran ventaja que es natural la aplicación en los casos en que la sumarización tiene como fin facilitarle a los usuarios la navegación y la búsqueda de contenido en videos.

Los estudios con usuarios son empleados en métodos de extracción de cuadros clave en varios trabajos [35, 86]. En el caso de Liu et al. [86] por cada toma los usuarios califican la totalida de los cuadros extaídos como las etiquetas “malo”, “aceptable” o “bueno”. La puntuación obtenida es usada para evaluar las técnicas de evaluación sobre una colección de videos amplia. Avila et al. [35] seleccionan una colección de 50 videos. Por cada video tienen como información válida cuadros seleccionadso por cinco usuarios. A partir de los cuadros de los usuarios y el resultado del método crean métricas cuantitavitas para evaluar el desempeño de un método dado.

En el caso de video condensado también es posible encontrar estudios con usuarios para evaluar resultados. Algunos trabajos exponen a los usuarios a los resultados y les piden que los califiquen [81, 93, 100, 122]. En los casos en que el largo del condensado es elegido a priori, también se evalúa a partir de los juicios de los usuarios distintos tipos de condensados [100]. La mayoría de los métodos reportan buena aceptación de los usuarios, sin embrargo tienen problemas de continuidad o suavidad a la hora de presentar el resulado a los usuarios.

2.6. Problemas a resolver y desafíos

De lo observado en los trabajos publicados, se pueden observar varios desafíos a resolver en el problema de la sumarización de video. En general, las propuestas que reportan buenos resultados solo son válidas para casos particulares. Son muy pocas las propuestas que intentan una solución general al problema de extraer un resumen del video.

La mayoría de los algoritmos de sumarización propuestos generan un espacio donde cada punto es un cuadro. A partir del espacio agrupan con un algoritmo de clasificación. Luego del agrupamiento extraen el resultado. Esa solución no permite la evaluación en línea y es muy costosa desde el punto de vista computacional.

Por otro lado, las que intentan reducir el costo de procesamiento hacen uso de las herramientas brindadadas por los métodos de compresión de video. Debido a que los métodos basados en el estándar MPEG dividen los cuadros en bloques, es muy dificultoso detectar cambios menores en los videos. Un ejemplo de este escenario son los videos de vigilancia.

Otro gran desafío pendiente es el desarrollo de un método de evaluación metodológicamente correcto. Algunos avances se reportaron en la competencia TRECVID (TREC Video Retrieval Evaluation), organizada por NIST (National Institute of Standards and Technology) [117]. Sin embargo, la mayoría de los datos a procesar solo están disponible para aquellos que participan en la competencia y dificulta la libre comparación de resultados. En general, el problema de la evaluación tampoco ha sido resuelto.

Considerando el escenario planteado, es necesario seguir investigando en el área. Existe la necesidad de un método de sumarización que sea general, produzca resultados en línea y tenga un costo computacional que permita operar sobre grandes volúmenes de datos. Además del método, es necesario un correcto marco de evaluación para poder determinar si se ha llegado al objetivo.

Capítulo 3

Nueva propuesta a la sumarización de video

Luego de explorar el estado del arte en la sumarización de video, se pueden delinear los requerimiento de un método de sumarización "ideal". Aunque puede haber diversidad en los enfoques, se puede llegar al menos a un acuerdo mínimo en requerimientos básicos a satisfacer. Una lista de requerimientos inicales puede ser que el método sea general, que detecte toda la información relevante y que sea computacionalmente viable para procesar grandes volúmenes de datos.

El primer requerimiento es que se sea general. Como se ha mencionado en la Sección 2.2.2, se pueden encontrar varias soluciones para dominos específicos. Aunque el resultado sea excelente en un domínio en particular, no es significativo para que aporte algo nuevo a los métodos existentes. Un método general debe concentrarse fundamentalmente en los cambios tratando de minimzar las suposiciones sobre el contenido potencial del video.

El segundo requerimiento es que detecte hasta el menor cambio en el video. Existen aplicaciones como la vigilancia por video que demandan este tipo de requerimientos. Los métodos que utlizan caracterísitcas globales tales como histogramas de colores, no sirven para estos casos. En los casos en que las tomas se realizan sobre grandes espacios y los cambios suceden en una fracción del cuadro, generalmente son invisibles a las caracterísitcas globales.

Finalmente es necesario que el método de sumarización sea aplicable a grandes volúmnes de datos. Constantenmente se está capturando video. En las grandes ciudades el problema se agraba a medida que se agregran más cámaras. Además la adopción acelerada de teléfonos inteligentes genera videos en alta definicón y de manera masiva. Un método de sumarización que no pueda procesar en línea no es apto para poder sacar conclusiones en tiempos razonables. De hecho, sería muy deseable que la velocidad de sumarización sea

más rápida que la velocidad de captura de la información.

La propuesta presentada en este trabajo se adapta a los requerimientos mencionados. Propone el uso de caracterísitcas locales como solución a la detección de cambios mínimos. El problema de la generalización se resuelve evaluando solo los cambios *relativos* de las carácterísticas locales. De esa forma se tiene gran parte del problema resuelto.

El principal obstáculo en el uso de características locales es el costo computacinal. Por cada cuadro hay que detectar los puntos de interés y además describirlos para poder indentificalos. En 2006 [15] se propuso el algoritmo SURF (Speeded-Up Robust Features) como una posible solución al problema de cálculo demandante para características locales. El algoritmo se publicó con algunas revisiones en 2008 [16]. Unos años más tarde se publicaron implementaciones aceleradas sobre arquitecturas de computadoras paralelas que permitían tener velocidades cercadanas a el procesamiento en línea.

A partir de los avances desde el punto de vista algorítmico y desde el punto de vista computacional, se desarrolló el método propuesto en este trabajo. En la próxima Sección se hace una breve reseña a SURF solo con fines informativos. La propuesta teórica del método de sumarización se documenta en la Sección 3.2. Una gran ventaja del algoritmo de sumarización presentado, es que permite obtener cuadros clave o video condensado. Este es otro punto saliente. En la Sección 3.3 se presenta la metodología la generación del video condensado.

3.1. Breve Reseña de SURF

SURF [15, 16] es un algoritmo que provee detección y descripción de características visuales de las imágenes. Las características locales de las imágenes son invariantes a la rotación y a la escala. Este hecho provee robustez a la hora de detectar **puntos de interés** (ipoint) en la imagen. Dada una imagen cualquiera se puede extraer una lista de ipoints. Es posible buscar correspondencia entre puntos de interés y de esa forma sacar conclusiones con mayor nivel de abstracción.

Cada ipoint contiene cuatro datos asociados, los cuales lo hace único. El primer dato consta de la ubicación en la imagen en coordenadas cartesianas. El segundo dato describe la orientación local del punto. La escala a la cual fue detectado el punto es el tercer dato. Finalmente, el cuarto dato consiste en un vector de descripción el cual consta de 64 o 128 elementos dependiendo de lo desado por el usuario.

La detección de los puntos relevantes se realiza usando el detector Fast-Hessian (Hessiano rápido). El detector obtiene un conjunto de matrices Hessianas usando una serie de filtros Gaussianos a escalas múltiples. Luego se usa una imagen integral para escalar el

Figura 3.1 Imagen de una caja y sus características locales

filtro. La ubicación del punto de interés se calcula detectando el máximo o mínimo local en la imagen con el uso de las matrices Hessianas.

La detección de la orientación local del ipoint permite invariancia respecto a la rotación. Usando un coeficiente de wavelets (onditas) se calcula la orientación en la vecindad del punto detectado. El ángulo de rotación se discretiza con un paso de 60 grados.

La construcción del descriptor se produce a partir del fraccionamiento de la vecindad del punto en una región regular. En cada división se aplica la wavelet de Haar (onditas de Haar). Se obtienen 4 coeficientes de cada región. Como resultado de dividir el vecindario en 16 sectores, es posible construir un vector de 64 elementos.

En la Figura 3.1 se muestra una imagen antes y después de ser procesada por SURF. En el caso de la imagen procesada, es posible ver cada una de las caractetisticas encontradas. Cada uno de los círculos mostrados está centrado sobre un ipoint con coordenadas x e y. La escala a la que fue detectada es indicada por el radio del cíclulo. La línea que une el centro con el círculo, indica la orientación del ipoint.

En la Figura 3.2 se puede ver a la caja anteriormente mencionada, acompañada de otros ítems dentro de una escena. Sobre la derecha de la figura, se puede ver la salida provista por SURF. En este caso también se puede distiguir las características locales detectadas.

En la Figura 3.3 se puede ver el resultado de la comparación y asociación de los ipoints. La mayoría de las asociaciones son correctas y se puede ver que la caja es detectada en la escena. Sin embargo, existen comparaciones que no son correctas. Este resultado es esperado y es parte de las limitaciones de la metodología usada.

Figura 3.2 Imagen de la caja en una escena y sus características locales

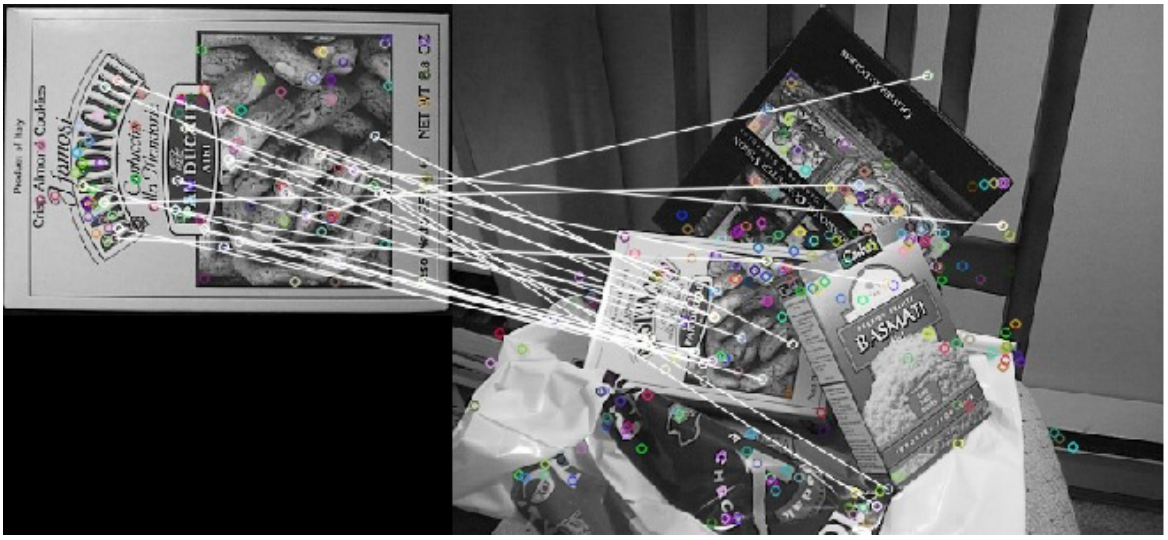

Figura 3.3 Resultado de la comparación de características de las dos imágenes

3.2. Nuestra propuesta

Nuestra propuesta para la sumarización de video tiene dos grandes ventajas. La primera de ellas es que toma en cuenta las características locales de las imágenes. Este hecho permite tener la capacidad de detectar el mínimo cambio entre dos cuadros. En el caso de usar características globales, un cambio menor dentro de la escena pasa inadvertido a la detección. Un típico caso son los videos de vigilancia. En estos casos es importante detectar un cambio de forma o una pequeña rotación de un objeto en la escena.

La segunda ventaja que posee nuestro método es la de poder procesar el video en línea (es decir, sobre la secuencia de video en tiempo real). No es necesario contar con el total del contenido del video para detectar cuadros clave. En nuestra propuesta, basta solo con unos pocos cuadros para poder producir resultados. El uso de SURF permite desempeños aceptables para procesar video a velocidades del orden de tiempo real. En el próximo capítulo se discutirán los aspectos relacionados al tiempo de procesamiento de nuestro método.

En términos generales, el método propuesto puede describirse como una serie de pasos. Todos los pasos que se describirán a continuación se aplican a cada uno de los cuadros de la secuencia del video. La Figura 3.4 detalla el diagrama de flujo del algoritmo propuesto. El primer paso consta en aplicar SURF a cada uno de los cuadros. Por cada cuadro se obtiene un conjunto de puntos característicos (ipoints) con una serie de descriptores asociados a cada uno de los puntos.

Luego de la detección de los puntos característicos, se buscan las coincidencias con los puntos característicos del cuadro anterior. El resultado de este proceso produce un número de coincidencias de puntos característicos. Este valor es un escalar que se define como el número común de puntos característicos (NCPC). Una vez que se obtiene el valor de NCPC, se calcula el promedio de los N últimos cuadros procesados. Luego de una serie de pruebas empíricas, se determinó que se necesitan guardar los valores de los 10 últimos cuadros procesados. Luego del cálculo del promedio, se procede a la detección de cambio.

Para la detección de un potencial cuadro clave, se analiza el valor absoluto del cambio porcentual entre el valor actual de NCPC y el promedio calculado. El cómputo del cambio porcentual provee un valor escalar que relaciona el “nivel de cambio” en el video. Seleccionando un nivel de umbral se puede determinar cuando el cambio es lo suficientemente significativo como para determinar si el cuadro es candidato a ser clave. Se normaliza el valor de umbral entre 0 y 1. En el caso de que el umbral se elija en 0, significa que cualquier cambio en el cuadro lo convierte en cuadro clave. En cotraste, en el caso que el umbral se fije en 1 ningún cuadro en el video se convertirá en cuadro clave. Este umbral permite el control de la **sensibilidad** del sistema.

Luego que un cuadro se ha determinado como potencial cuadro clave, se procede al paso de filtrado. Se realiza una nueva búsqueda de coincidencias entre el potencial cuadro clave y el último cuadro clave válido. De esta forma se puede evaluar "la cantidad de ruido" computando el cambio porcentual en valor absoluto luego de la comparación. De una forma similar a lo hecho con el umbral de sensibilidad, se define el **umbral de ruido**. Donde un valor 0 para el umbral de rudo significa que todos los candidatos se convierten en cuadros clave. En contraste, ningún cuadro se convierte en cuadro clave si el umbral de ruido toma valor 1.

En la Figura 3.5 se pude ver el funcionamiento del algoritmo en una secuencia de 20 cuadros. El gráfico superior muestra la cantidad actual de coincidencias y el promedio. Se puede observar que la curva promedio sigue a la cantidad instantánea de coincidencias. En el gráfico central se muestra el nivel de cambio porcentual y el valor de umbral elegido.

Sobre la parte inferior de la Figura 3.5 se muestran tres secuencias de video. La primer secuencia son los cuadros del video original. La segunda secuencia son los cuadros elegidos como candidatos los cuales muestran un nivel de cambio superior al umbral elegido. Finalmente se muestran los cuadros clave elegidos luego de ser comparados teniendo en cuenta el umbral de ruido seleccionado.

3.3. Creación de videos de resumen

A partir del algoritmo descripto para la detección de cuadros clave es posible generar un nuevo video que sea el resumen del original. Debido a la naturaleza del esquema planteado se puede generar un resumen de manera directa. Luego de la detección de un cuadro clave es posible agregar la secuencia de cuadros que le sigue al cuadro detectado. Solo es necesario agregar una pequeña cantidad de cuadros. De esta forma se puede obtener una versión comprimida del video original la cual es inteligible por un observador humano.

Luego de una serie de pruebas con usuarios, se pudo observar que una escena que dure menos de medio segundo causa una percepción de "salto" en el video resultado. En contraste, escenas de mayor largo generan un resumen más extenso. Se ha encontrado que una buena solución de compromiso es darle al usuario la cantidad de cuadros equivalente a un segundo luego de que se detecta el cuadro clave. Como resultado, se ha optado por agregar 30 cuadros consecutivos luego de la detección del cuadro clave.

En la Figura 3.6 se puede observar la descripción del proceso. En la parte superior se muestra una línea temporal donde se marcan los cuadros clave detectados. En este caso los cuadros se identifican como A, B y C. La parte inferior de la figura muestra cómo se

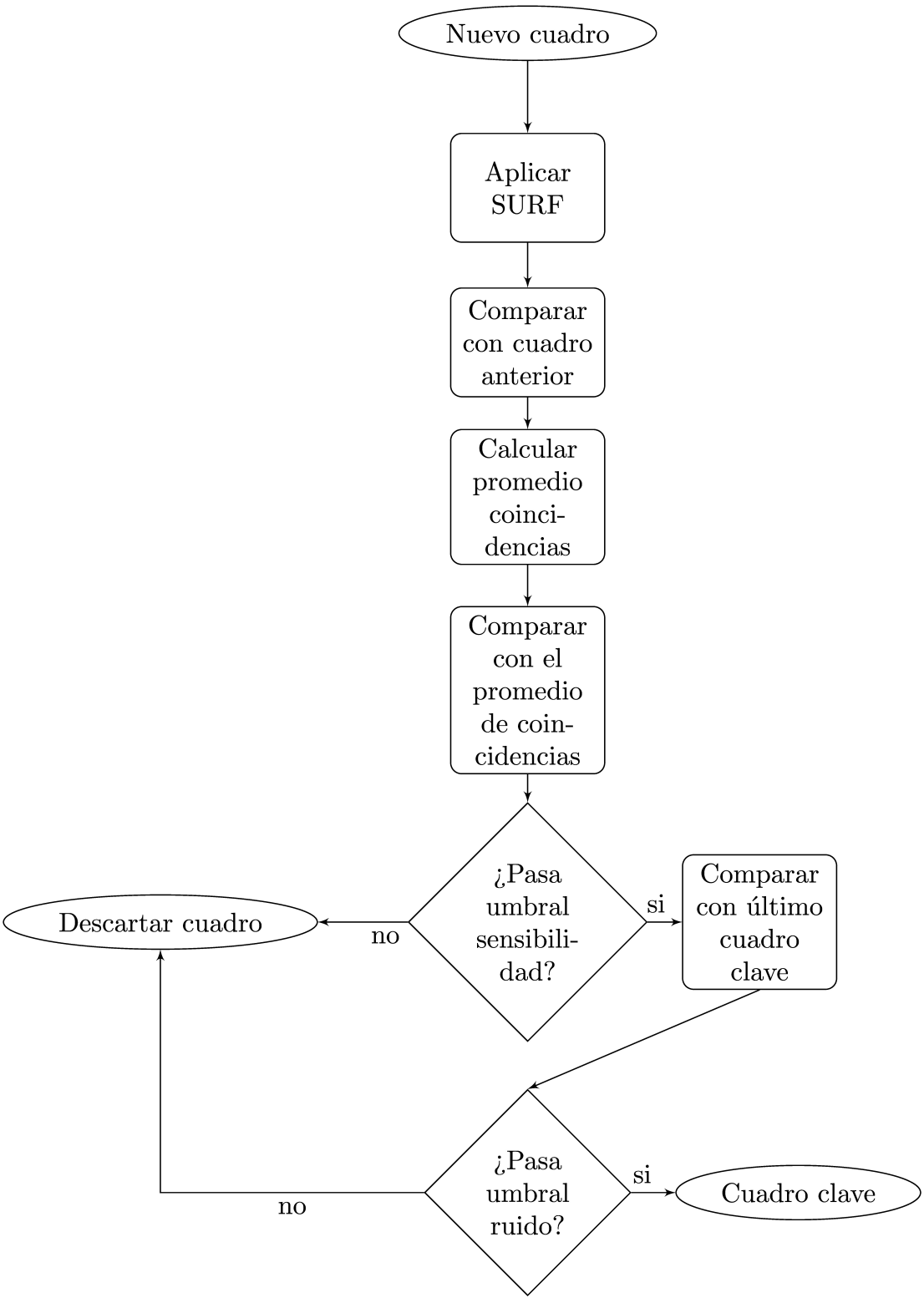

Figura 3.4 Diagrama de flujo del algoritmo propuesto.

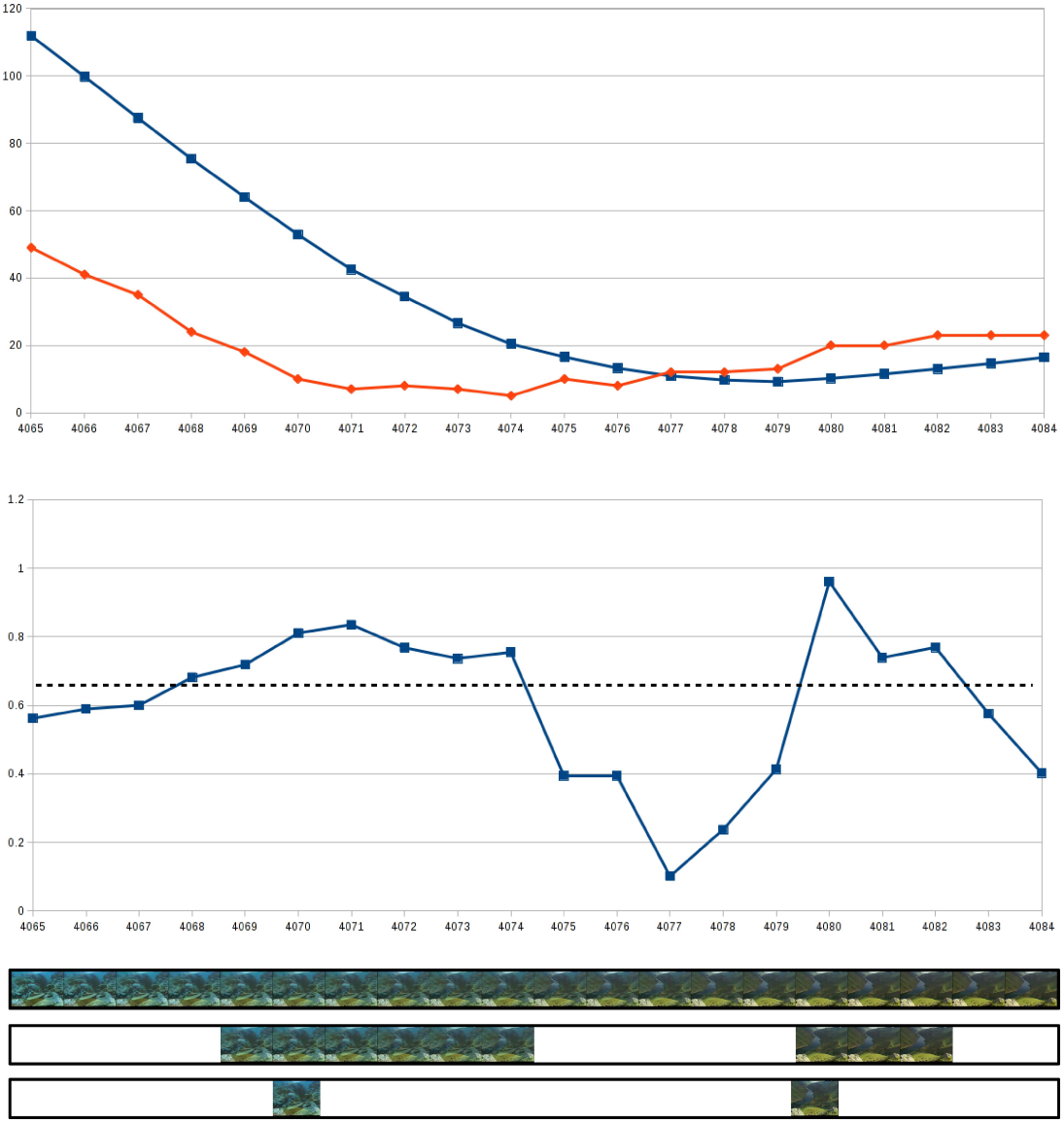

Figura 3.5 Visualización dinámica de la respuesta del algoritmo propuesto.

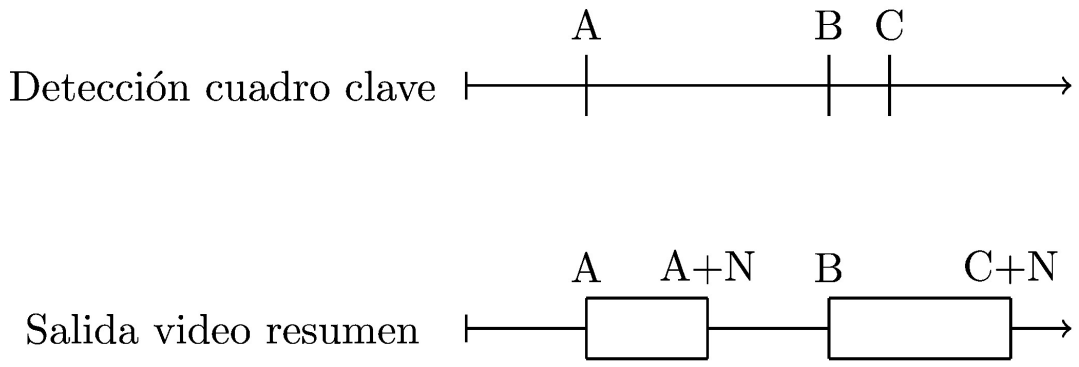

Figura 3.6 Generación de un video de resumen a partir de la detección de cuadros clave.

genera el video de resumen. En este caso N equivale a 30 cuadros. Se puede observar que la distancia entre B y C es menor a N por lo tanto se agregan N cuadros luego de C.

Es importante destacar dos caracterísitcas beneficiosas del proceso propuesto. La primera ventaja es que el video resumen es generado en línea directamente sobre la secuencia de video. Esto permite que la fuente del video puede ser una cámara o algún dispositivo generador de video en tiempo real. Como resultado es posible generar a elección los cuadros claves o el resumen de video de forma inmediata.

El segundo beneficio del algoritmo propuesto es que el usuario puede seguir los eventos dentro del video resumen. Esto es posible porque el procesamiento elimina la información redundante y conserva los momentos importantes sin alterar el tiempo original. El hecho de agregar cuadros contiguos le permite al observador entender naturalmente los eventos en el video. El resultado final es muy similar a como luce un video editado por un operador humano.

Un tercer beneficio es la cantidad reducida de parámetros de operación. El usuario del algoritmo tiene solo dos parámetros para variar y el signicado de cada uno de ellos es claramente intuitivo. La configuración propuesta facilita la operación y convierte al método sumamente flexible para explorar resultados.

En el siguiente capítulo evaluaremos extensivamente la propuesta presentada y la compararemos con los métodos más destacados que se presentaron en el capítulo anterior. Se realiza una evaluación cualitativa y una evaluación cuantitativa. Además se propone un nuevo método de evaluación que mejora lo relevado en la literatura.

Capítulo 4

Resultados

4.1. Evaluación de la sumarización de video y sus problemas asociados

En cualquier área del conocimiento es necesario poder comparar soluciones que intentan resolver el mismo problema. Por esa razón es muy importante poder comparar resultados en el contexto de la sumarización de video. Como se ha mencionado en la Sección 2.5, varios investigadores concuerdan que no existe un marco de evaluación consistente para la sumarización de video [26, 35, 80, 129]. De hecho es muy frecuente encontrar que cada una de las soluciones propuestas cuenta con su propia metodología de trabajo. Además generalmente no es posible encontrar ninguna comparación en términos de desempeño.

En esta sección se presentan los resultados del método de sumarización propuesto. Siguiendo la clasificación descripta en la Sección 2.5, se presentan dos tipos de evaluaciones. El primer caso es una evaluación cualitativa la cual se limita a describir los resultados. La segunda evaluación es cuantitativa y es del tipo de estudios con usuarios. Ambas evaluaciones toman como referencia metodologías y datos usados previamente en la literatura.

Además de realizar una evaluación cuantitativa de los resultados, se propone un nuevo método de medición. La creación de una nueva propuesta de medición se basa en falencias detectadas en la metodología existente. La nueva metodología y los resultados obtenidos se documentan en la Sección 4.4.

4.2. Evaluación cualitativa

La evaluación cualitativa o descripción del resultado es la primera forma de comparar métodos de sumarización. Cronológicamente esta metodología fue la primera en ser aplicada. En este caso hemos seguido las mismas comparaciones realizadas por Guan et. al. [53]. Aunque no es la mejor metodología posible, aporta una visión cualitativa de la sumarización.

Para realizar la comparación cuantitativa se ejecutó el método de sumarización propuesto usando videos estándar disponibles en Open Video Project (OV)[1]. OVP provee los videos y además brinda una serie de cuadros clave asociados. Luego de una onptimización simple, se llegó a la conclusión de utilizar 0.25 para el coeficiente de sensibilidad y 0.5 para el coeficiente de filtrado. Los resultados presentados en esta sección están disponibles públicamente en un sitio de Internet con acceso público [2].

La Figura 4.1 muestra los cuadros clave del video *NASA 25th Anniversary Show Segment 1*. En la parte superior de la figura se pueden observar los cuadros seleccionados por Guan et. al. [53]. En la parte intermedia se muestran los cuadros provisto por Open Video Project (OV). Los cuadros seleccionados por el método propuesto en estre trabajo se ven en la parte inferior (SURF-SUMM).

A partir de los resultados presentados en la Figura 4.1, se puede decir que SURF-SUMM produce resultados que son comparables con métodos existentes. De los cuadros seleccionados por los tres métodos se puede inferir que coinciden en la relevancia del satélite, del astronauta, del cohete, del planeta y del trasbordador espacial. El segmento que muestra al planeta Saturno se puede ver que produce repeticiones de cuadro para el caso de SURF-SUMM. Este comportamiento se debe a que las caracterísitas locales cambian en el caso en que se realiza el escalado del objeto. Las implicancias de este efecto son tratadas con mayor detalle en la Sección 5.2.

La Figura 4.2 muestra los cuadros clave de el video *NASA 25th Anniversary Show Segment 3*. En este caso, se comparan los resultados de SURF-SUMM con los publicados por OV, Besiris et al. [18], and Guan et al. [53]. Se puede concluir que la mayoría de los cuadros clave propuestos por los trabajos relacionados son capturados por SURF-SUMM. La sumarización produce resultados comparables con los trabajos equivalentes. De manera similar a instancias anteriores, en este caso es posible observar repeticiones de cuadros en dos secuencias. La primera toma muestra una cabina movida por un brazo, la segunda toma muestra una cabina fija que rota sobre un eje. En ambas tomas se puede observar que SURF-SUMM es sensible a el cambio de las caracterísitas locales. De forma similar al

[1] http://www.open-video.org/
[2] http://www.javieriparraguirre.net/video-summarization/

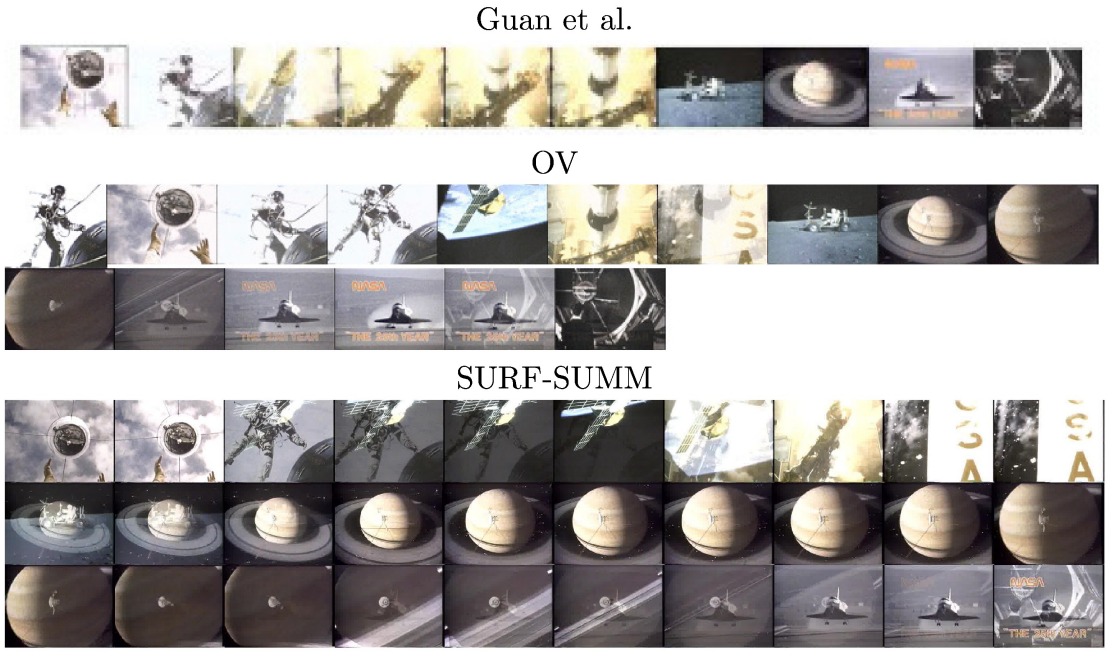

Figura 4.1 Cuadros clave del video NASA 25th Anniversary Show, Segmento 01

caso de la Figura 4.1, es un comportamiento típico del algoritmo propuesto.

Es relevante destacar que las Figuras 4.1 y 4.2 comparan resultados de métodos que utilizan la totalidad del video antes de producir resultados. Este hecho no permite la sumarización en línea. Además en el caso de Guan et al. [53], el método de sumarización es sumamente complejo. El resultado surje a partir de una clasificación y luego selección de cuadros a partir de un criterio de cobertura de la información.

La Figura 4.3 muestra los cuadros clave para el video *A New Horizon Segment 2*. En este caso se compara la salida de SURF-SUMM con los resultados publicados por OV y por Almeida et al. [4]. A diferencia de las comparaciones cualitativas anteriores, en este caso se compara con un método de sumarización en línea. De la información mostrada, se puede concluir que SURF-SUMM detecta más información que OV y que Almeida et al. [4]. Un aspecto destacado de SURF-SUMM es que en los primeros cuadros de la secuencia es capaz de detectar las divisiones dentro del mapa. Este aspecto es desultado de detectar las caracterísitas locales en contraste a usar histograma de colores (o caracterísitas globales). Se puede ver este caso como un beneficio del uso de caracterísitas locales.

De las comparaciones cualitativas realizadas se puede inferir que los resultados producidos por SUMM-SURF son comparables con métodos que representan al estado del arte en el área. Los resultados son comparables a métodos que usan el video completo

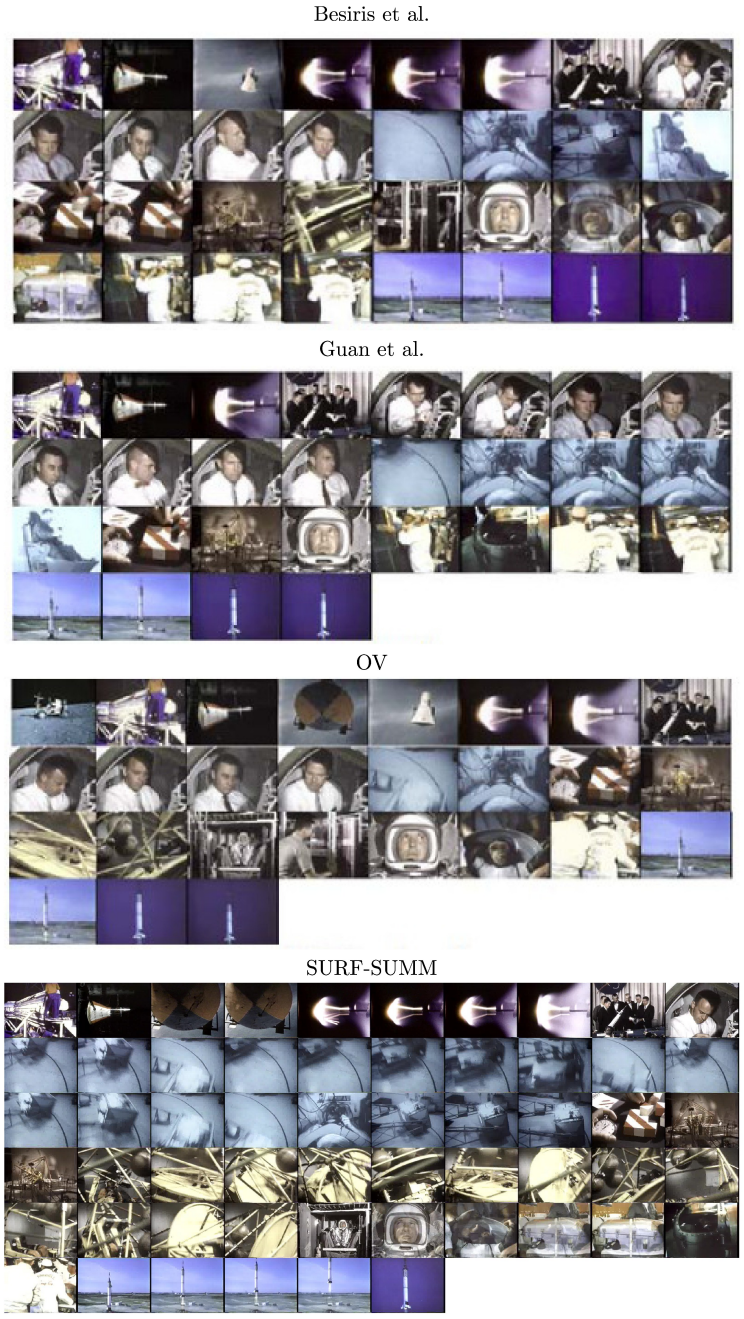

Figura 4.2 Cuadros clave del video NASA 25th Anniversary Show, Segmento 03

Figura 4.3 Cuadros clave del video A New Horizon, Segmento 02

antes de producir resultados. Además, se muestran beneficios respecto a métodos de sumarización en línea exitentes. Desde un punto de vista cualitativo se puede concluir que los resultados son aceptables. De todas formas, es necesario una metodología de evaluación que minimice la subjetividad implícita. Por esa razón en la próxima sección se aborda una evaluación numérica.

4.3. Evaluación cuantitativa

La evaluación cuantitativa aborda el problema de cuantificar el resultado de un método de sumarización. A partit de la problemática planteada respecto a los métodos de evaluación, se decidió seguir un método que involucre estudios con usuarios. En la literatura es posible encontrar varias alternativas. Sin embargo se optó por seguir método CUS (Comparison of Summaries) que fue propuesto por De Avila et al. [35]. La decisión se basa en que CUS está claramente planteado, provee un conjuntos de datos libre de uso, provee usuarios de referencia y es reconocido en la literatura.

4.3.1. Método CUS

El método de medición CUS tiene como motivación tres objetivos. El primero es reducir la subjetividad en la evaluación de la sumarización. El segundo es cuantificar la calidad del resultado. Finalmente, el tercer objetivo es proponer métricas objetivas que sean usadas por las distintas propuestas de abstracción de video.

La metodología de evaluación consiste en tres pasos. Inicialmente, usuarios humanos son invitados a mirar videos y elegir los cuadros que sumarizan el video según sus propios criterios (paso 1). Los usuarios pueden elegir la cantidad de cuadros que consideren necesaria. El segundo paso consiste en comparar los cuadros seleccionados por los usuarios con los elegidos por el método de sumarización. El tercer paso consiste en calcular las métricas que cuantifican la sumarización. La Figura 4.4 muestra los tres pasos propuestos por CUS.

Es importante destacar que en el caso de CUS los usuarios seleccionan los cuadros de un conjunto el cual no contiene el total de los cuadros en el video. Los creadores del método de medición muestrean los videos con una frecuencia de un cuadro por segundo y generan un subconjunto de cuadros. Los usuarios pueden seleccinar cuadros clave del subconjunto provisto. La principal ventaja de esta decisión es facilitar la tarea de los usuarios debido a la significativa reducción de la cantidad de cuadros. Por ejemplo, para un video de 2 minutos de largo con una velocidad de 30 cuadros por segundo, se muestrean 120 cuadros sobre un

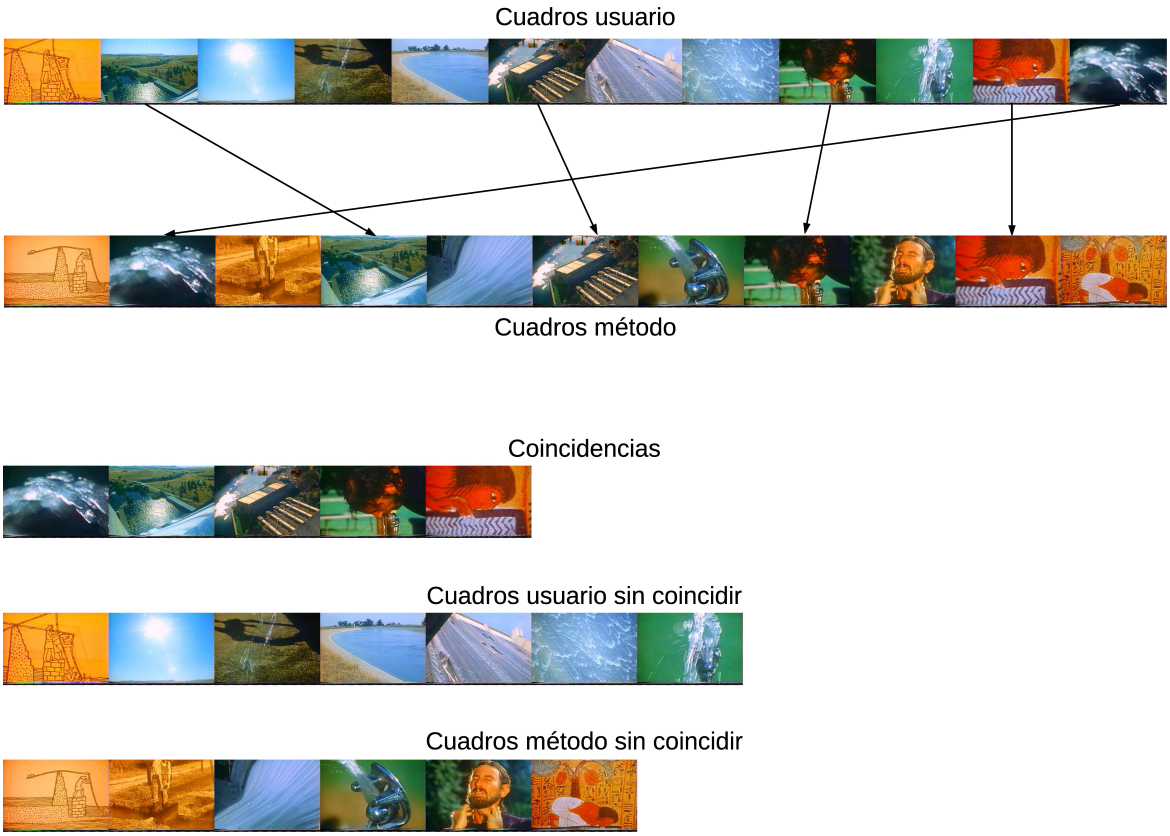

Figura 4.4 Método de medición CUS propuesto por De Avila et al. [35]

total de 3600. Sin embargo, esta decisión tiene implicacias las cuales son abordadas en la Sección 4.4.

Respecto al criterio de decisión sobre la igualdad de dos cuadros, CUS propone el cómputo de los histogramas de colores y luego el cálculo de la distancia Manhattan entre los mismos. En caso de que la distancia sea menor que un umbral δ dado, se consideran los dos cuadros como iguales. Luego que los dos cuadros son marcados como coincidentes, se eliminan de la lista y se procede con el resto de la información disponible (Figura 4.4, paso 2). En todos los experimentos en los cuales se ha usado CUS, se ha tomado un valor de δ constante igual a 0.5.

CUS define dos métricas para cuantificar la calidad de una sumarización. La primera es CUS_A la cual mide la precisión (accuracy). La segunda es CUS_E y cuantifica el nivel de error. Las ecuaciones 4.1 y 4.2 defienen las méticas, donde n_{mAS} es la cantidad de cuadros coincidentes de la sumarización automática, $n_{\overline{m}AS}$ es la cantidad de cuadros sin coincidencias de la sumarización automática y n_{US} es la cantidad de cuadros clave seleccionados por el usuario.

$$CUS_A = \frac{n_{mAS}}{n_{US}} \tag{4.1}$$

$$CUS_E = \frac{n_{\overline{m}AS}}{n_{US}} \tag{4.2}$$

Los valores de CUS_A varían entre 0 y 1. El peor caso es 0 y significa que ninguno de los cuadros elegidos por la sumarización automática (AS) ha coincidido con los elegidos por el usuario (US). Es importante remarcar que cuando CUS_A es 1 no significa que todos los cuadros han coincidido. Puede darse el caso en que la cantidad de cuadros elegidos por la sumarización automática n_{AS} sea mayor a la cantidad elegida por el usuario (n_{US}). En ese escenario, quedan cuadros de AS sin coincidir.

CUS_E varía entre 0 y $\frac{n_{AS}}{n_{US}}$. El mejor caso es cuando todos los cuadros en AS coinciden con los cuadros elegidos por el usuario y CUS_E vale 0. Cuando no se produce coincidencias, estamos en el peor caso. Las dos métricas propuestas por CUS son complementarias. Es decir, la sumarización de mayor calidad produce $CUS_A = 1$ y $CUS_E = 0$. Ese es el caso en que todos los cuadros claves en AS y US encuentran un par.

CUS provee dos conjuntos de videos como información de referencia. Cada conjunto contiene 50 videos. El primier conjunto se compone de 50 videos seleccionados de OV. El segundo conjunto contien 50 videos obtenidos de YouTube.

En un sitio web de acceso público [3] es posible acceder a los binarios que implementan

[3]https://sites.google.com/site/vsummsite/

las métricas propuestas por CUS, los dos conjuntos de videos, y resultados de varios trabajos relacionados. Además de lo mencionado, CUS provee 5 elecciones de cuadros clave realizdas por usuarios para cada video. Este escenario hace que CUS sea una buena referenica para realizar experimentación cuantitativa de sumarización de video.

Los trabajos relacionados que aportan cuadros claves para comparar son los De Avila et al. [35] (VSUMM), los cuadros clave de Open Video Project (OV), Murdur et al. [98] (DT) y Furini et al. presentaron SITMO [46]. El caso de SITMO está relacionado con un trabajo similar denominado VISTO realizado por el mismo autor [45]. En los experimentos presentados en el presente trabajo, se consideran SITMO y VISTO como equivalentes debido a que solo se cuenta con un conjunto de cuadros clave para las dos versiones.

Almeida et al. [6] presentaron un trabajo titulado VISON (VIdeo Summarization for ONline applications). En ese trabajo, los autores obtienen resultados cuantitativos de el métdodo de sumarización propuesto por ellos usando los datos provistos por CUS. A diferencia de lo realizado por De Avila et al. [35], en VSION se usan nuevas métricas y se descartan videos del conjunto original. Esto hace dificultoso reproducir los resultados numéricos presentados en VSION. Sin embargo, los autores publican los cuadros producidos por VISON en un sitio de Internet público [4]. Esta información permite incluir a VISON en los resultados cuantitativos usando métricas disponibles.

Sintonización de SUMM-SURF

Como se documentó en la Sección 3.2, el algoritmo de sumarización propuesto en este trabajo (SUMM-SURF) solo requiere dos parámetros para producir resultados. El primer parámetro es el *nivel de sensibilidad* y el segundo es el *nivel de filtrado*. Ambos están normalizados entre 0 y 1 (también se puede decir entre 0 % y 100 %).

Debido a que el resultado del resumen depende de la elección de sensibilidad y filtrado, es importante lograr encontrar los valores óptimos para maximizar el resultado. Siguiendo un concepto similar al propuesto por Almeida et al. [6], se utilizó Valor-F (también conocida como F-measure o F-score) como criterio de optimización. La Ecuación 4.3 define Valor-F.

$$Valor - F = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4.3}$$

En las Ecuaciones 4.4 y 4.5 se definen los componentes de Valor-F a partir de los parámetros usandos por CUS. El coeficiente Valor-F presenta una evaluación unidimensional de la sumarización. Debido a que CUS_A y CUS_E son complementarias, no es simple encontrar un óptimo con estas dos métricas. Usando Valor-F es posible optimizar

[4] http://www.liv.ic.unicamp.br/~jurandy/vison/

los parámetro de sintonía de un método de sumarización para un conjunto de datos dado.

$$Precision = \frac{n_{mAS}}{n_{AS}} = \frac{CUS_A}{CUS_A + CUS_E} \tag{4.4}$$

$$Recall = \frac{n_{mAS}}{n_{US}} = CUS_A \tag{4.5}$$

Otro beneficio de tener Valor-F como métrica de evaluación es la simplicidad para comparar propuestas de sumarización distintas. A mayor valor de F, mejor es la calidad de la sumarización. Los valor de CUS_A y CUS_E no son descartados, se usan para clacular Valor-F.

A partir de los usuarios y los dos conjuntos de videos provistos por CUS, se realizaron sumarizaciones usando SUMM-SURF. Se variaron los valores de sensibilidad ente 5 % y 95 % con un paso de 5 %. Por cada valor elegido de sensibilidad, se fue variando el nivel de filtrado entre 60 % y 95 % con un paso de 5 %. Para cada convinación de los parámetros sensibilidad y filtrado se obtuvieron los valores de CUS_A y CUS_E usando el programa provisto por CUS con un valor de δ de 0.5.

La Figura 4.5 muestra cómo varían los valores de CUS_A en función de la variación de los parámetros sensibilidad y filtrado. De forma similar, la Figura 4.6 muestra los valores obtenidos de CUS_E. La variación de Valor-F se muestran en la Figura 4.7.

En las Figuras 4.5 y 4.6 se puede observar la relación que existe entre CUS_A y CUS_E. Cuando el error es bajo, la exactitud (accuracy) también toma valores bajos. También se observa que para valores altos de exactitud, el error es alto. En la Figura 4.7 se puede ver la variación de Valor-F. En este caso se obtuvo un Valor-F máximo de 0.343 para una sensibilidad de 70 % y un filtrado de 80 %.

Es importante destacar que siguiendo esta metodolía es fácil encontrar el valor óptimio de los parámetros. Dado un conjunto de datos, rápidamente se puede configurar la respuesta. La “simpleza” de variar sensibilidad y filtrado permite una adaptación intuitiva de SUMM-SURF.

Resultados Cuantitativos Usando Medición CUS

Luego de obtener el Valor-F máximo para SUMM-SURF se procedió a calcluar la misma métrica para el resto de los métodos de los cuales se dispone cuadros clave. Esta decisión se basó en que la publicación en la cual se presentó CUS solo se presentan valores de CUS_A y CUS_E. Además las métricas presentadas no se calculan sobre los dos conjuntos de videos y se presentan de forma individual. Los valores obetnidos se muestran en Tabla 4.1.

De los resultados obtenidos se puede concluir que SUMM-SURF no presenta la mejor

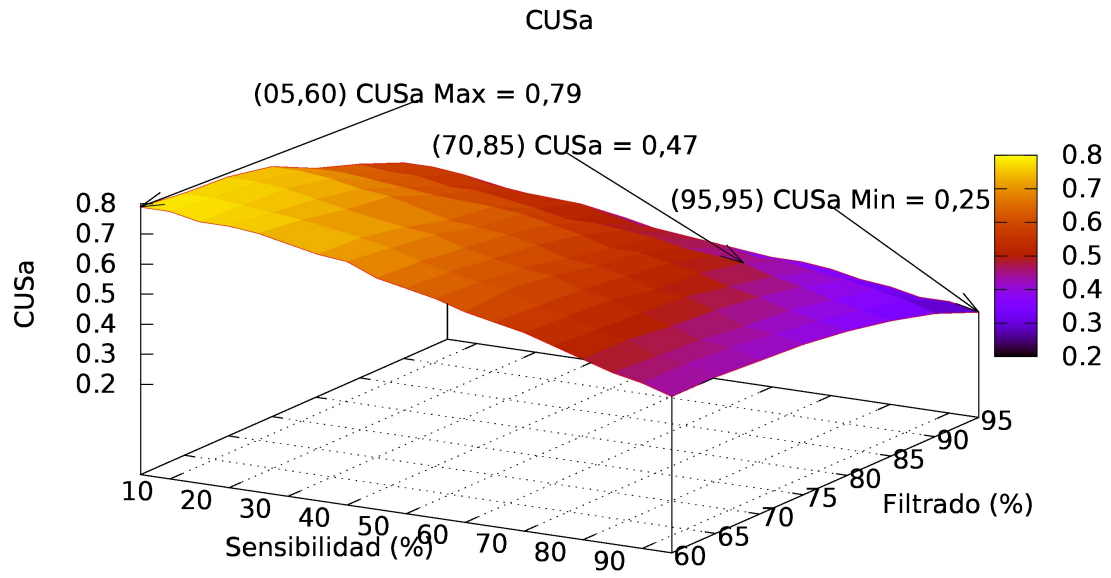

Figura 4.5 Resultados de CUS_A para SUMM-SURF sobre el conjunto de datos y usuarios provistos por CUS

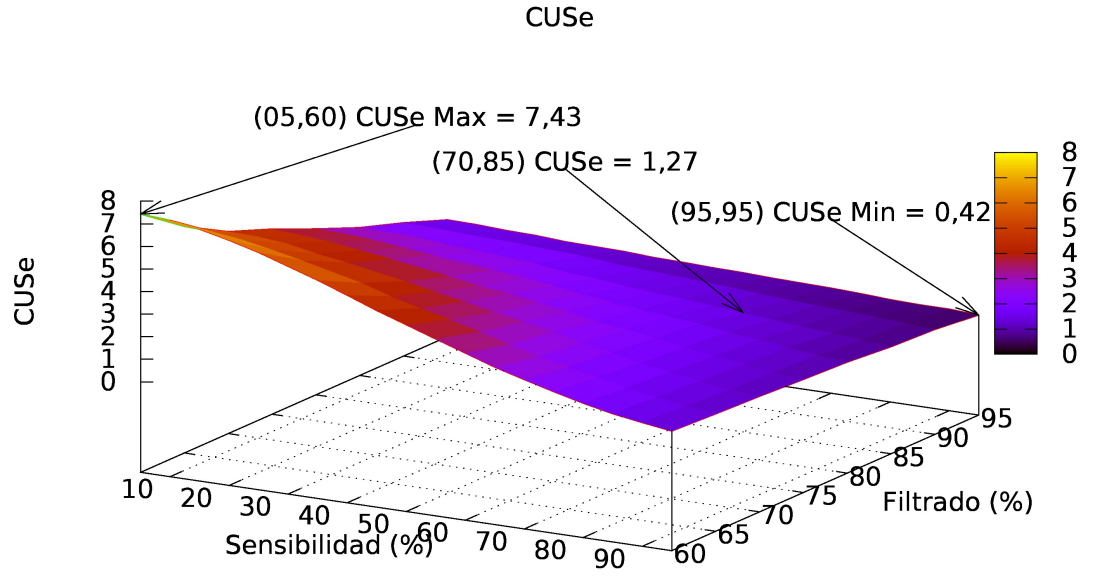

Figura 4.6 Resultados CUS_E para SUMM-SURF sobre el conjunto de datos y usuarios provistos por CUS

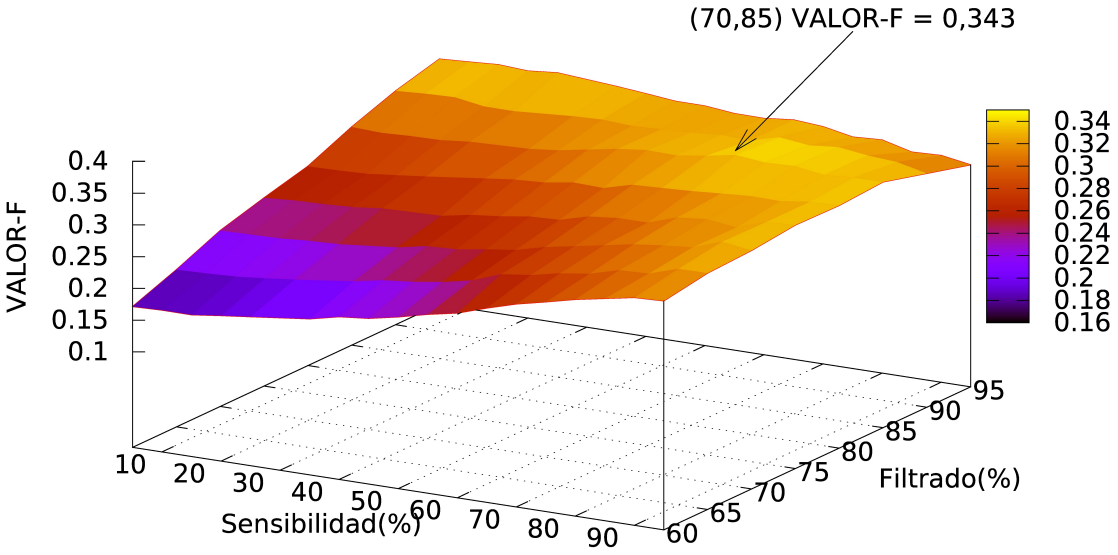

Figura 4.7 Resultados de Valor-F para SUMM-SURF sobre el conjunto de datos y usuarios provistos por CUS

Tabla 4.1 Resultados comparativos con el resto de los métodos usando el método, los videos y los usuarios provisos por CUS

Método	Valor-F
OV	0.616
DT	0.582
STIMO	0.626
VSUMM1	0.762
VSUMM2	0.710
SURF-SUMM	0.343

Tabla 4.2 Listado de resultados producidos por CUS a ser analizados

Video	Valor-F	Aspecto destacado
43	0.545	Valor-F por encima de la media obtenida por SURF-SUMM, valor máximo obtendido.
62	0.348	Valor-F cercano a la media obtenida por SURF-SUMM.
37	0.100	Valor-F por debajo de la media obtenida por SURF-SUMM, valor mínimo obtenido.
25	0.545	Valor-F cercano a la media obtenida por SURF-SUMM, ejemplo de video usado para la comparación cuantitativa.

calidad de sumarización. Como el resultado no parece alentador a primera vista, se decidió analizar si CUS es verdaderamente un método de comparación justo y genérico.

Análisis de Resultados Usando Medición CUS

Luego de obtener valores cuantitativos para SUMM-SURF y la comparación con los métodos disponibles, se procedió al análisis del método CUS. De la totalidad de los videos provistos, se eligieron 4 ejemplos. Los videos analizados se muestran en la Tabla 4.2. En la tabla se pueden obervar los motivos por los cuales fueron elegidos los 4 videos listados.

La Figura 4.8 muestra los cuadros clave seleccionados por los usuarios provistos por CUS. En la Figura 4.9 se documentan los resultados producidos por los métodos de sumarización evaluados en este trabajo. En el caso particular del video 43, SUMM-SURF obtuvo el máximo Valor-F sobre todo el conjunto de videos. A pesar de obtener el máximo valor, SUMM-SURF sigue por debajo de la calificación del resto de los métodos comparados.

Las Figuras 4.10 y 4.11 muestran los cuadros claves seleccionados por los usuarios y por los métodos de sumarización. El video 62 fue seleccionado debido a que el Valor-F obtenido por SUMM-SURF es muy cercano a la media. En este caso se pueden observar varios aspectos “curiosos” en lo que respecta a CUS. El primer resumen de VSUMM obtiene la puntuación más alta y es el resumen que produce la menor cantidad de cuadros claves. Además no muestra la bandera, la cual fue elegida por más de un usuario.

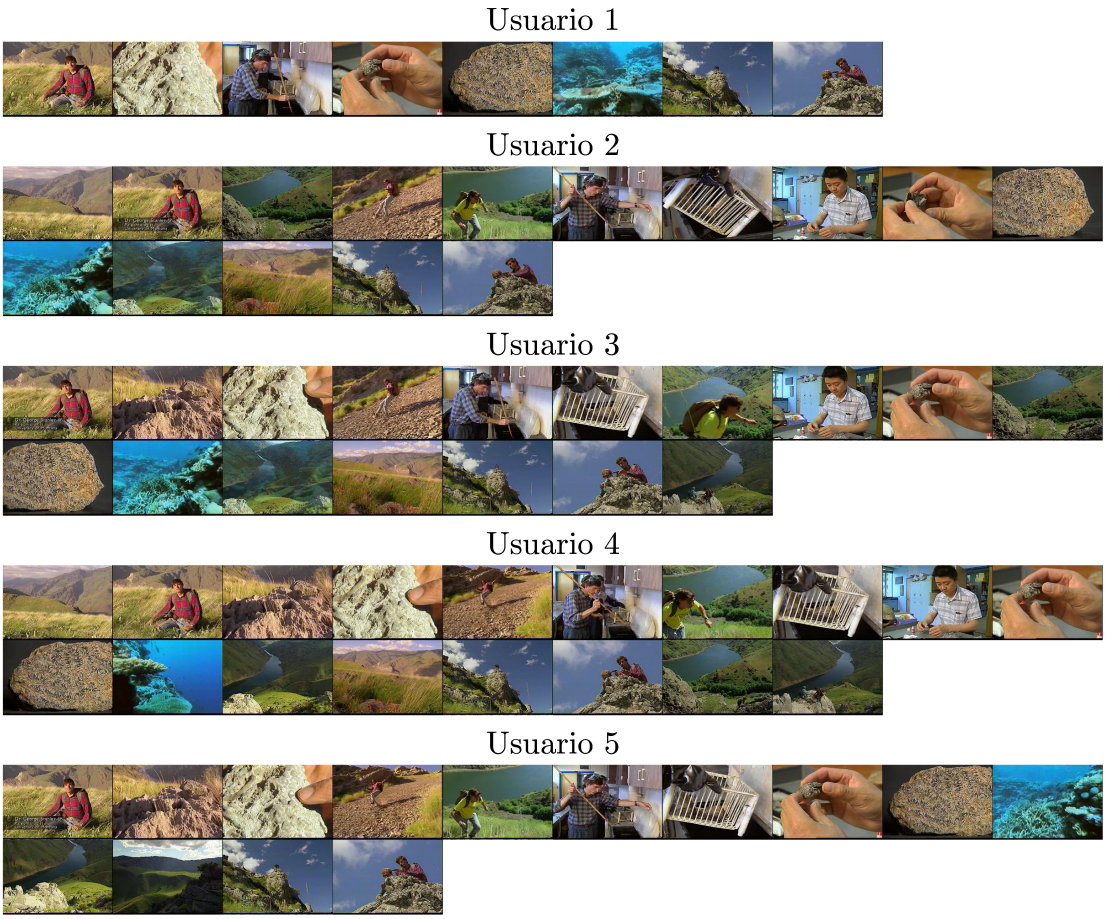

Figura 4.8 Análisis de CUS: cuadros clave elegidos por los usuarios provistos por CUS para el video 43

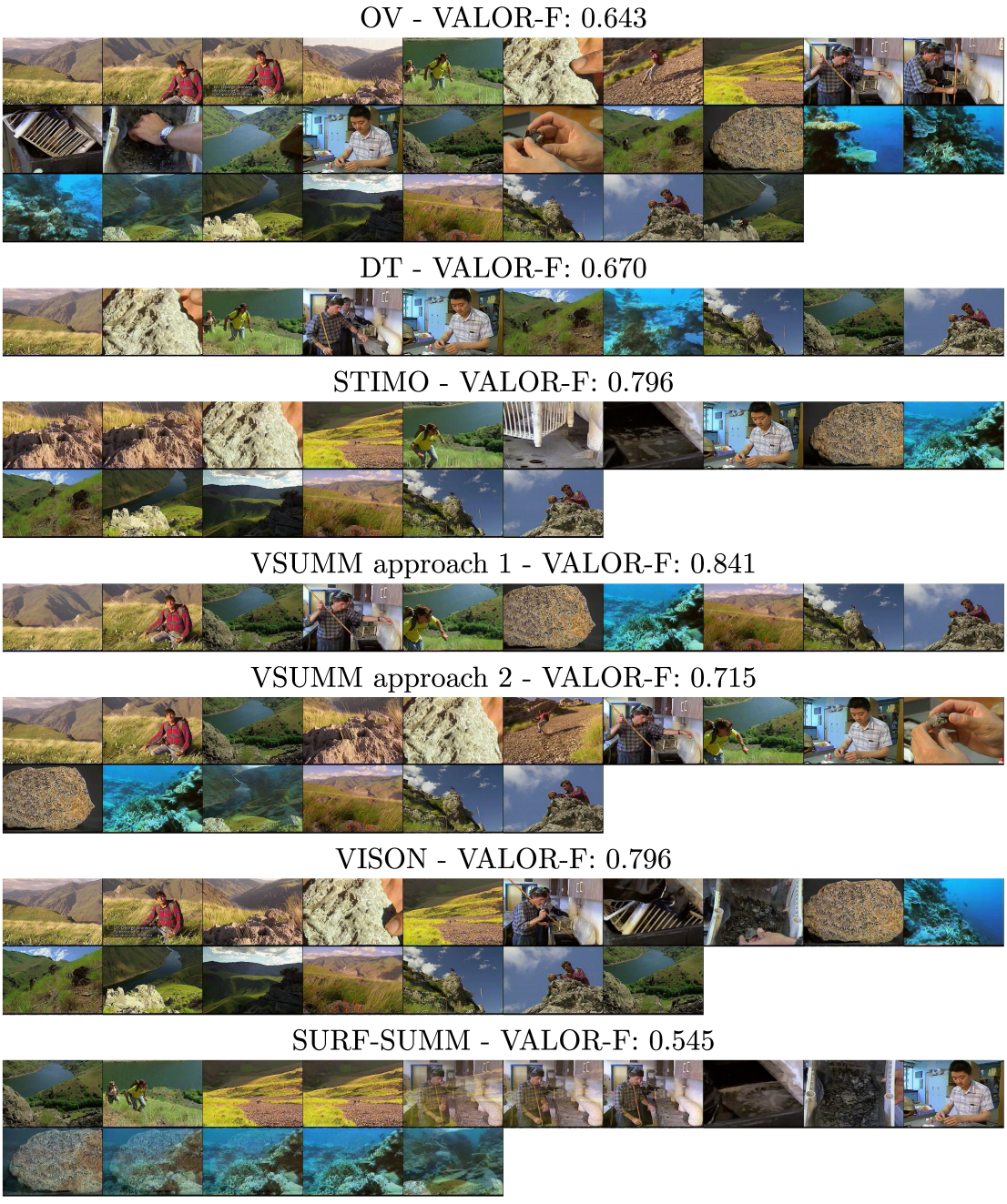

Figura 4.9 Análisis de CUS: resúmenes de los los métodos de sumarización para el video 43

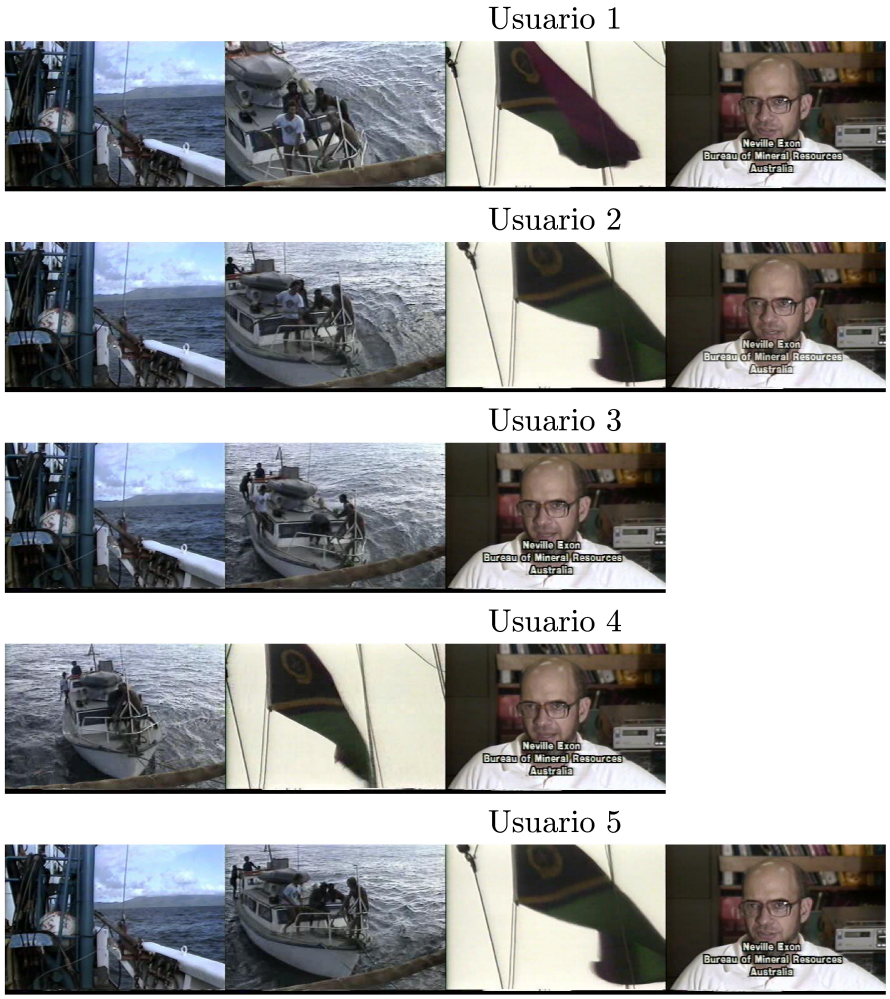

Figura 4.10 Análisis de CUS: cuadros clave elegidos por los usuarios provistos por CUS para el video 62

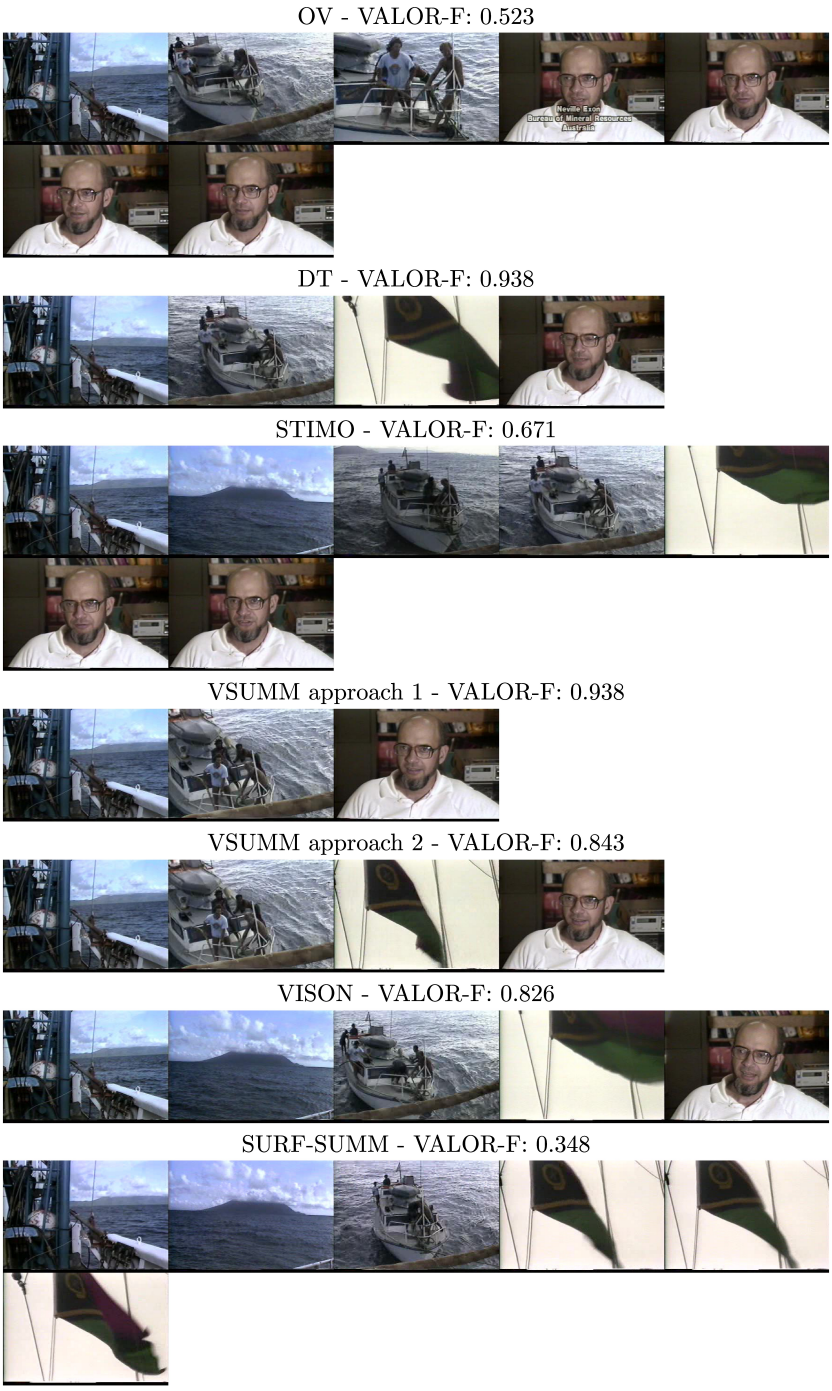

Figura 4.11 Análisis de CUS: resúmenes de los los métodos de sumarización para el video 62

Figura 4.12 Análisis de CUS: cuadros clave elegidos por los usuarios provistos por CUS para el video 37

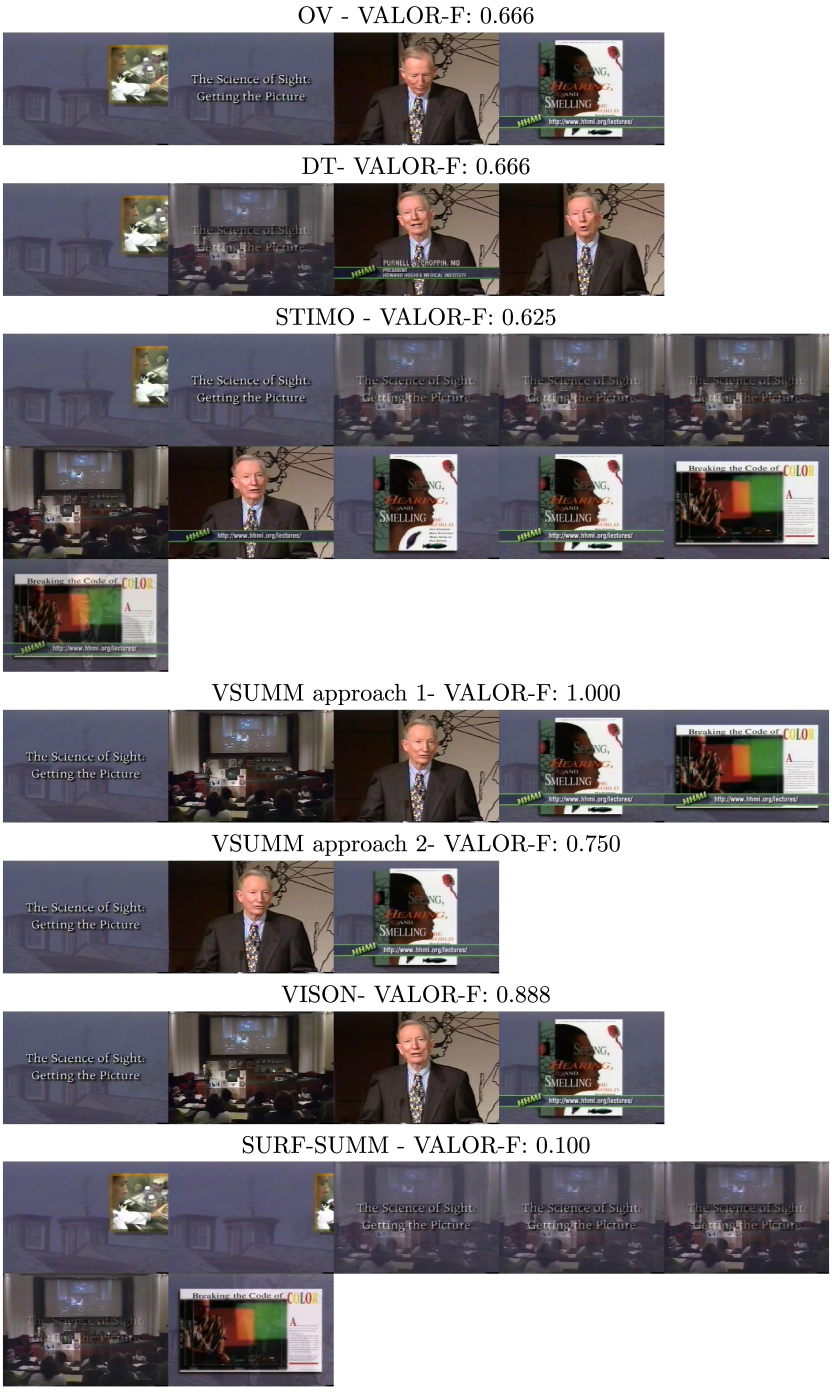

Figura 4.13 Análisis de CUS: resúmenes de los los métodos de sumarización para el video 37

La Figura 4.12 contiene los cuadros clave seleccionados por los usuarios para el video 37, mientras que la figura 4.13 muestra los cuadors clave seleccionados por los métodos automáticos. El video 37 fué elegido porque SUMM-SURF obtiene la peor puntuación del conjunto de datos. Hay varias coincidencias que explican las razones de la baja marca que obtiene SUMM-SURF . La primera es la coincidencia de los usuarios en la elección de los cuadros clave. En la Figura 4.12 se puede ver que los cinco usuarios han coincidido en los cuadros elegios. Este escenario va a penalizar marcadamente a cualquier resumen que se desvíe mínimamente de los cuadros seleccionados como referencia. Otro motivo por el cual SUMM-SURF obtiene baja puntuación, de debe a que en video 37 se pueden que las caracterísitas locales detectan cambios mínimos. El el comienzo del resumen se puede ver un efecto de edición en el cual el cuadro se retira de la imagen deslizándose. Esa información es detectada por las caracterísitas locales. Además se puede ver que las caracterísitas locales son “sensibles” al cambio de intensidad del texto superpuesto sobre las imágenes. Ambos casos de falla de SUMM-SURF se deben a casos especiales de post-edición del video. Si se considera la totalidad de los videos disponibles, son dos casos altamente atípicos y pueden decirse que SUMM-SURF no pierde caracter de método de sumarización para un conjunto general de videos.

Las Figuras 4.14 y 4.15 contienen los cuadros clave seleccinados por los usuarios y por los métodos de abstracción automáticos para el video 25 del conjunto de datos provisto por CUS. Este video fué usado el la Sección 4.2 para realizar un análisis cualitativo. En este caso, la salida de SUMM-SURF contiene menos cuadros clave debido a que los umbrales de sensibilidad y filtrado no han sido incrementados.

En el caso del video 25 se puede observar que SUMM-SURF supera en puntuación a algunos de los métodos disponibles. De todas formas sigue sin poder acercarse a la puntuación de VSUMM o VISON. Un detalle a destacar en este caso es que algunos usuarios destacan el mapa con las divisiones políticas como un cuadro relevante. Ninguno de los métodos con mayor puntaje elige un cuadro que se incluya en la secuencia del mapa con divisiones.

4.3.2. Falencias del método de medición CUS

A partir de lo observado en los resultados del método CUS, en esta sección se mencionan las falencias observadas. Se han encontrado al menos 4 falencias que impactan significativamente en la evaluación. La primer falencia es la forma en la que se buscan las coincidencias de los cuadros. El segundo punto discutible se trata de las métricas numéricas usuadas. En tercer lugar, la elección de cuadros por parte de los usuarios está influenciada por el hecho de solo permitirles elegir dentro un subconjuto de cuadros.

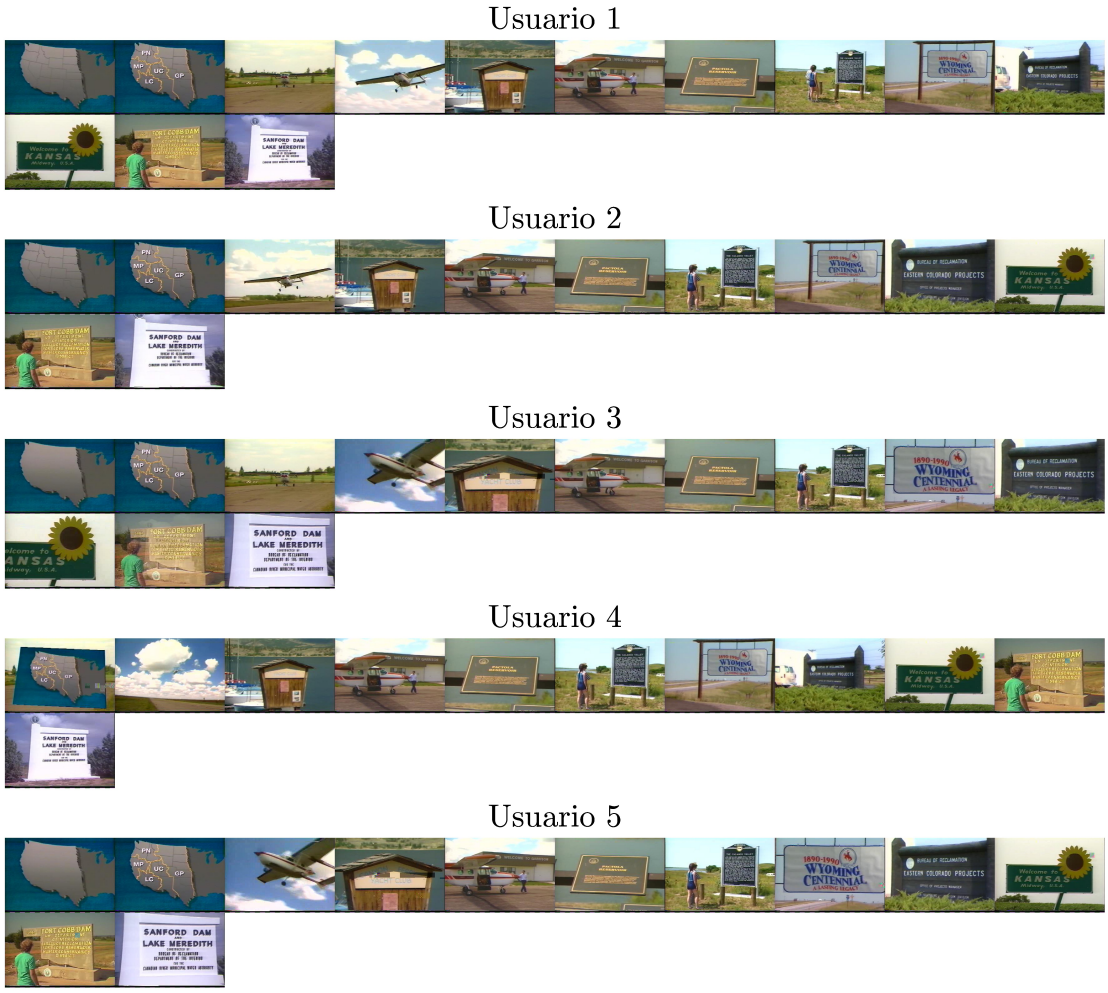

Figura 4.14 Análisis de CUS: cuadros clave elegidos por los usuarios provistos por CUS para el video 25

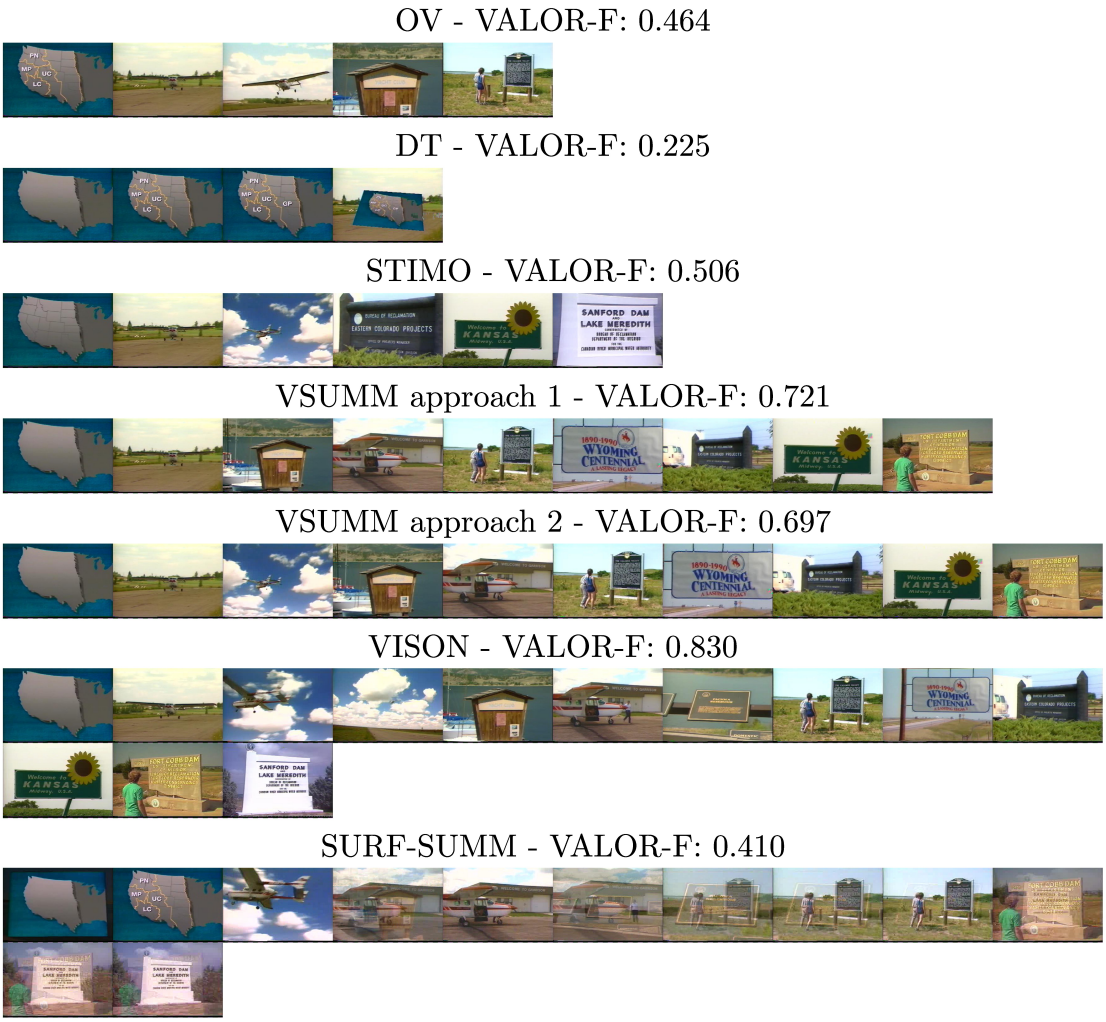

Figura 4.15 Análisis de CUS: resúmenes de los los métodos de sumarización para el video 25

Finalmente, la metodología propuesta por CUS para encontrar dos cuadros similares se basa en comparar los histogramas de colores. No tiene en cuenta en lo absoluto la distancia temporal entre ambos. El mismo cuadro puede ser repetido con una diferencia arbitraria de tiempo en un video. Para CUS, la diferencia temporal no influye.

Las métricas propuestas por CUS son CUS_A y CUS_E. El Valor-F no es usado por CUS pero se introduce como coeficiente único de evaluación. Todas las métricas usadas hasta el momento descartan el hecho de que una sumarización de video es un típico ejemplo de un clasificador con clases desbalanceadas [55]. Un caso típico es encontrar 5 cuadros en un video de 2 minutos de duración. Ese escenario indica que 0.138 % de los cuadros son verdaderos positivos para un video típico de 30 cuadros por segundo. Es necesario elegir otro tipo de métrica para cuantificar resultados de coincidencias en clases desbalanceadas.

Respecto a la elección de los cuadros clave por parte de los usuarios presentada a CUS tiene dos aspectos discutibles. El primer aspecto es que los usuarios seleccionan cuadros del conjunto resultante de muestrear el video a un cuadro por segundo. Esta decisión es arbitraria debido a que si en un video estándar la frecuencia es de 30 cuadros por segundo, es altamente probable que el cuadro clave seleccionado por el usuario se encuentre muy distante del momento en el que el cambio ha sucedido. En el caso de la detección de cambios en el video, la distancia en cuadros de dos sumarizaciones debe ser tenida en cuenta en la creación de una métrica.

Las Figuras 4.16 y 4.17 muestran los cuadros selecciandos por los usuarios proporcionados por CUS y por los métodos de sumarización para el video 21. Entre los cuadros seleccionados por los usuarios se puede ver diversidad de “opiniones” respecto a los cuadros clave. Lo destacable respecto a CUS, es que uno de los métodos de sumarización (DT) selecciona un solo cuadro clave y obtiene mejor puntuación que SUMM-SURF. Esta evidencia muestra la falencias de las métricas obtenidas por CUS.

El video 21 tiene como particularidad que cuenta un historia usando imágenes monocromáticas. Los usuarios 3 y 4 provistos por CUS marcan los cuadros de esta historia como relevantes (ver Figura 4.16). Sin embargo, el único método que detecta la historia en su gran mayoría es SUMM-SURF. Este beneficio se debe al uso de caracterísitas locales para la detección de cambios relevantes.

4.4. Nuevo método de medición

En esta Sección se propone un nuevo método de medición de calidad de sumarización. A partir de las falencias de CUS mencionadas, se construye una propuesta. Se introducen cambios a tres nieveles. El primer aspecto a tener en cuenta es la distancia entre los cuadros

Figura 4.16 Cuestionamiento a CUS: cuadros clave elegidos por los usuarios provistos por CUS para el video 21

Figura 4.17 Cuestionamiento a CUS: resúmenes de los los métodos de sumarización para el video 21

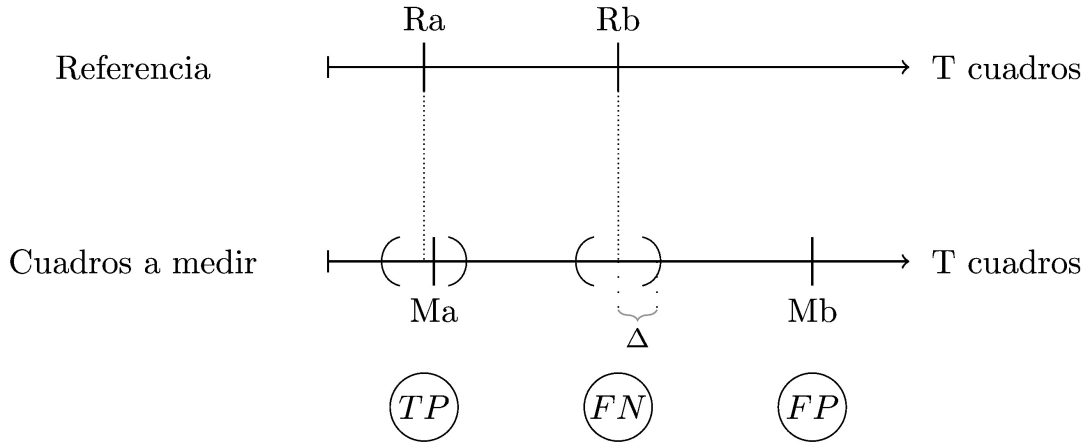

Figura 4.18 Nuevo método de medición propuesto

a comparar. El segundo cambio es la introducción de una nueva métrica que tenga en cuenta el caso de clases desbalanceadas. Finalmente, se presenta un nuevo conjunto de cuadros clave generado por usuarios.

El nuevo método de sumarización compara los cuadros seleccionados por los usuarios (referencia) con un método de sumarización automático (cuadros amedir). Dado un cuadro de referencia, se busca un cuadro en un entorno al mismo dentro de los cuadros a medir. Solo en el caso en que se encuentre, se compara la similitud calculando la correlación de los histogramas de colores. En el caso en que el nivel de correlación supere un umbral dado, se consideran que los cuadros son iguales.

La Figura 4.18 muestra un gráfico que documenta la metodología seguida. Dado un cuadro de referencia, se busca en un entorno al mismo de Δ cuadros. De los cuadros a medir mostrados en la figura, solo *Ma* va a ser comparado con *Ra*. En el caso que la correlación de los histotramas de colores de *Ma* y *Ra* sea mayor al umbral ε, los cuadros son considerados iguales. La Tabla 4.3 documenta el significados de los parámetros usados por el nuevo método propuesto.

En la implementación del procedimiento, a cada cuadro se lo transforma al espacio de colores HSV. Se utilizan 50 clases para H y 60 clases para S. El rango de variación de H va de 0 a 179 y el de S de 0 a 255. Antes de comparar los histogramas, se normalizan. Finalmente, se calcula la correlación la cual puede variar entre 0 y 1. Siendo 1 conincidencia perfecta y 0 sin correlación.

El nuevo método de sumarización se denomina OSM (Open Summarization Metrics). El código fuente del método OSM está disponible en un sitio de Intenet con acceso público

Tabla 4.3 Parámetros del nuevo método de medición propuesto

Símbolo	Descripción
Δ	Amplitud del entorno en cuadros. Distancia máxima en cuadros.
e	Umbral de correlación correlación de histograma de colores.

Figura 4.19 Ejemplo de coincidencias para OSM con un $\Delta = 2$

[5]. Es importante destacar la libre disponiblidad del código de medición. Esto facilita el intercambio de información entre investigadores y contribuye a simplificar el proceso de comparación de propuestas de sumarización de video.

La Figura 4.19 muestra un ejemplo de coincidencias detectadas por OSM para un $\Delta = 2$. En la parte superior se puede observar los cuadros originales del video y la identificación numérica del orden de aparición. En la parte intermedia de la figura se muestran los cuadros elegidos por el usuario y el método de sumarización. El usuario ha elegido los cuadros 71 y 78, mientras que el método ha elegido los cuadros 74 y 78. OSM toma como referencia a los cuadros elegidos por el usuario. Como la distancia máxima permitida para este ejemplo es 2, solo se considera como coincidencia el cuadro 78 (TP).

[5]http://www.javieriparraguirre.net/open-sumarization-metrics/

Tabla 4.4 Cálculo de una tabla de contingencia a partir de dos conjuntos de cuadros clave

		Referencia		Total
		No	Si	
Medido	No	C_{N_r,N_m} (VN)	C_{S_r,N_m} (FN)	C_{N_M}
	Si	C_{N_r,S_m} (FP)	C_{S_r,S_m} (VP)	C_{S_M}
Total		C_{N_R}	C_{S_R}	T

Coeficiente Kappa de Cohen

El ceficiente Kappa de Cohen [133] fué inicialmente propuesto para encontrar una medida de consenso entre diferentes evaluaciones subjetivas. Para el caso de una situación definida, cada uno de los observadores clasifica los casos en categorías. Con la clasificación de los observadores, se construye una tabla de contingencia la cual expresa el resultado de las comparaciones correlacionadas. Este coeficiente se adapta mejor al problema de clases desbalanceadas y es usado en OSM.

Para construir una tabla de contingencia se consideran dos conjuntos de cuadros. Un conjunto es la referencia o los elegidos por un usuario, el otro es el conjunto a medir o los cuadros seleccionados por un método automático (ver Figura 4.18). Por cada uno de los cuadros en el video original, los dos observadores responden si consideran al cuadro en cuestión como cuadro clave. Con las respuestas pueden darse cuatro casos. El primer caso es que los dos observadores coincidan declararlo cuadro clave, ese caso se denomina un verdadero positivo (TP). Cuando la referencia considera un cuadro clave y el método no lo detecta, se considera un falso negativo (FN). Cuando el método detecta un cuadro clave pero este no es encontrado en la referencia, se considera el caso como un falso positivo (FP). Finalmente, cuando los dos observadores coinciden en descartar el cuadro, es considerado un verdadero negativo (TN).

La Tabla 4.4 muestra cómo se construye la tabla de contingencia a partir de los casos mostrados en la Figura 4.18. Es posible observar como se relacionan los cuatro casos posibles (TP, FP, TN y FN) mostrados en la Tabla y en la Figura. El total de cuadros del video original se reprenta con la constante T. El valor C_{N_r,N_m} representa la cantidad de cuadros declarados como negativos por la referencia y declarados negativos por el método. C_{N_M} significa la cantiad de cuadros declarados negativos por el método, mientras que C_{N_R} representa la cantidad de cuadros descartados por la referencia.

Una vez consituída la taba de contingencia, se puede calcular el coeficiente Kappa de Cohen. La Ecuación 4.8 define formalmente el coeficiente. Las Ecuaciones 4.6 y 4.7 relacionan el coeficiente con la tabla de contingencia. P_o indica el acuerdo observado entre los evaluadores y P_e cuantifica la probabilidad hipotética de posibilidad de acuerdo.

Tabla 4.5 Interpretación de los valores de kappa

Kappa	Nivel del acuerdo
<0.01	Elección al azar
0.01 - 0.20	Acuerdo bajo
0.21 - 0.40	Acuerdo razonable
0.41 - 0.60	Acuerdo moderado
0.61 - 0.80	Acuerdo elevado
0.81 - 0.99	Acuerdo casi perfecto

$$P_o = \frac{C_{N_r,N_m} + C_{S_r,S_m}}{T} \tag{4.6}$$

$$P_e = \frac{C_{N_M}}{T} * \frac{C_{N_R}}{T} + \frac{C_{S_M}}{T} * \frac{C_{S_R}}{T} \tag{4.7}$$

$$k = \frac{P_o - P_e}{1 - P_e} \tag{4.8}$$

El valor del coeficiente Kappa puede tomar valores entre 0 y 1, donde 0 indica que los evaluadores han elegido al azar sus respuestas y 1 significa un acuerdo perfecto entre ambos. La Tabla 4.5 indica el significado de los posibles valores que puede tomar la métrica. Es imporante destacar que un valor del coeficiente Kappa superor a 0.2 se considera un acuerdo razonable.

Ejemplo cálculo de Coeficiente Kappa de Cohen

En esta sección se muestra un ejemplo práctico de cómo obtener un valor del coeficiente Kappa a partir de un caso típico. Se parte con el ejemplo de un video el cual tiene un total de 2000 cuadros. Un típico caso es que el usuario elija 4 cuadros clave y el método automático elige 6.

A partir de los cuadros, se comienza a analizar las coincidencias buscando en las cercanías de los cuadros clave elegidos por el usario. En el caso que se encuentre una coincidencia, se retiran los cuadros del listado y se prosigue con el siguiente cuadro elegido por la referencia. El procedimiento se repite hasta que se hayan recorrido todos los cuadros de la referencia. Si el usuario ha elegido 4 cuadros, se hacen 4 búsquedas de conincias.

Suponiendo que se encuentran dos coincidencias, se puede construir una tabla de contincencias como se muestra en la Tabla 4.6. En este caso hay 2 verdaderos positivos. Sabiendo que la referencia ha elegido 4 cuadros, se puede inferir que hay 2 falsos negativos. Además, si el método ha elegido 6 cuadros, hay 4 falsos positivos.

Tabla 4.6 Ejemplo de una tabla de contingencia

		Referencia		Total
		No	Si	
Medido	No	1992	2	1994
	Si	4	2	6
Total		1996	4	2000

A partir de la Tabla 4.6, se puede calcular la métrica. P_o se calcula como lo indica la Ecuación 4.9. El cálculo de P_e se muestra en la Ecuación 4.10. Finalmente, el valor del coeficiente Kappa se obtiene como se muesta en la Ecuación 4.11.

$$P_o = \frac{1992+2}{2000} = 0.997 \tag{4.9}$$

$$P_e = \frac{1994}{2000} * \frac{1996}{2000} + \frac{6}{2000} * \frac{6}{2000} = 0.995012 \tag{4.10}$$

$$k = \frac{P_o - P_e}{1 - P_e} = 0.3986 \tag{4.11}$$

4.4.1. Nuevos cuadros de referencia

Debido a las falencias encontrdas en los cuadros provistos por los usuarios CUS, se decidió generar un nuevo conjunto de cuadros claves con nuevos usuarios. Se convocó a un grupo de estudiantes de postgrado de la Universidad Nacional del Sur, los cuales no estuviesen relacionados con la problemática y se les pidió que participen de la experiencia. Cabe destacar que se prestó mucha atención a que ninguna de las personas involucradas estén relacionadas con la problemática para minizar los posibles errores experimentales debido a subjetividades.

A cada uno de los usuarios se les proveyó un conjunto de videos y la totalidad de los cuadros de cada video. Se propueso que para cada video, el usuario elija los cuadros que mejor represente el contenido del video según su criterio. Se les indicó que no había restricción respecto a la cantidad de los cuadros a elegir.

Antes de comenzar con la experiencia, solo fue indicado un solo lineamiento que los usuarios deberían seguir. En lo casos de tomas con cuadros muy similares, se les sugirió que elijan el cuadro clave que se encuentre lo más adelante posible en términos de tiempo. La indicación se basó en dos suposiciones. La primera premisa es que cuando un cuadro clave es detectado, es importante que la detección sea lo más temprana posible. Además, dado un cuadro clave, naturalmente se asume que todos los cuadros que lo siguen estan relacionados

Tabla 4.7 Videos seleccionados de CUS para realizar mediciones con el nuevo método

Video	ID	Cantidad cuadros
Video 21	1	3282
Video 25	2	1795
Video 28	3	3559
Video 29	4	1942
Video 37	5	3411
Video 43	6	4791
Video 51	7	2931
Video 58	8	3180
Video 60	9	2091
Video 62	10	2615

con el cuadro seleccionado hasta que se elija un nuevo cuadro.

A partir de la metodología planteada se obtuvo un nuevo conjunto de cuadros clave. La principal diferenca respecto a los cuadros de usuarios provisos por CUS es que el nuevo conjunto minimiza las "demoras" entre cambio producido en el video y el cuadro clave seleccionado. Salvo la variante indicada, se procedió de la misma forma que CUS.

Como se dijo previamente, CUS provee 100 videos divididos en dos conjuntos de 50. EL primer conjunto contiene videos extraídos de OVP y el segundo contiende videos provenites de YouTube. Debido a la gran cantidad de información, solo se eligieron al azar 10 videos del conjunto que contiene videos pertenecientes a OVP. No se eliegieron videos del segundo conjunto debido a que no se disponen datos de sumarizaciones de terceros sobre ese conjunto.

En la Tabla 4.7 se muestran los videos seleccionadados y la cantidad de cuadros que contiene cada uno de ellos. Por cada uno de los videos, 5 usuarios eligieron cuadros clave. Los cuadros clave elegidos por los usuarios (usuarios OSM) están disponibles públicamente en Internet [6].

4.4.2. Resultados cuantitativos usando OSM y usuarios de CUS

A partir del nuevo método de medición definido, se procedió a realizar una nueva serie de experimentaciones. Se ejecutó OSM para todos los resultados de los métodos de sumarización disponibles usando como referencia los usuarios provistos por CUS. Además de buscó el máximo valor del coeficiente Kappa para SUMM-SURF. En este experiemento, solo se usaron los 10 videos seleccionados para poder comparar luego con los usuarios

[6] http://www.javieriparraguirre.net/open-sumarization-metrics/

Tabla 4.8 Resultados cuantitativos: método OSM, usuarios CUS. Sensibilidad 10, filtrado 80.

Método	Doble	Δ	*e*	CUSa	CUSe	F	Kappa
DT	SI	60	0.4	0.275	0.462	0.302	0.303
OV	SI	60	0.4	0.463	0.792	0.399	0.402
STIMO	SI	60	0.4	0.420	0.954	0.354	0.354
VSUMM1	SI	60	0.4	0.544	0.830	0.480	**0.489**
VSUMM2	SI	60	0.4	0.395	0.632	0.393	0.399
VISON	SI	60	0.4	0.514	0.886	0.444	0.448
SUMM-SURF	SI	60	0.4	0.484	2.194	0.302	0.307

OSM.

El máximo valor de Kappa para SUMM-SURF se obtuvo con un valor de sensibilidad de 10 % y un nivel de filtado de 80 %. En la Tabla 4.8 se pueden observar los resultados obtendios. En la tabla se indica si la distancia en cuadros (Δ), el valor del umbral de correlación (*e*) y si el entorno de búsqueda de coincidencia es hacia atrás y adelante en el tiempo (Doble).

En este experimento se eligió una distancia de 60 cuadros. Lo cual significa que ante un cuadro elegido por el usuario, se buscan coincidencias hasta 60 cuadros por delante y hasta 60 cuadros por detrás. Considerando un video estándar de 30 cuadros por segundos, se buscan cuadros similares con una distancia de hasta 2 segundos desde el cuadro clave de referencia.

De los resultados mostrados en la Tabla 4.8 se puede concluir que el método de sumarización que consigue el mayor valor del coeficiente Kappa de Cohen es VSUMM 1. Además se puede observar que las diferencias entre los distintos métodos de sumarización incluyendo nuestra propuesta no son tan pronunciadas como en los resultados producidos por CUS.

4.4.3. Resultados cuantitativos usando OSM y usuarios de OSM

En esta sección se presentan los resultados obtenidos utilizando el método medición OSM y los nuevos cuadros de referencia. Todos los resultados mostrados en esta sección se ejecutaron sobre los 10 videos seleccionados indicados en Tabla 4.7. Se ejecutó la medición de calidad de sumarización para todos los métdos dispoibles y para SUMM-SURF.

Siguiendo la misma metodología, se buscó el máximo valor del coeficiente Kappa de Cohen para SUMM-SURF sobre este nuevo conjunto de datos. Se variaron los valores de sensibilidad entre 5 y 95 % y los valores de filtrado entre 60 y 95 %. Luego de la

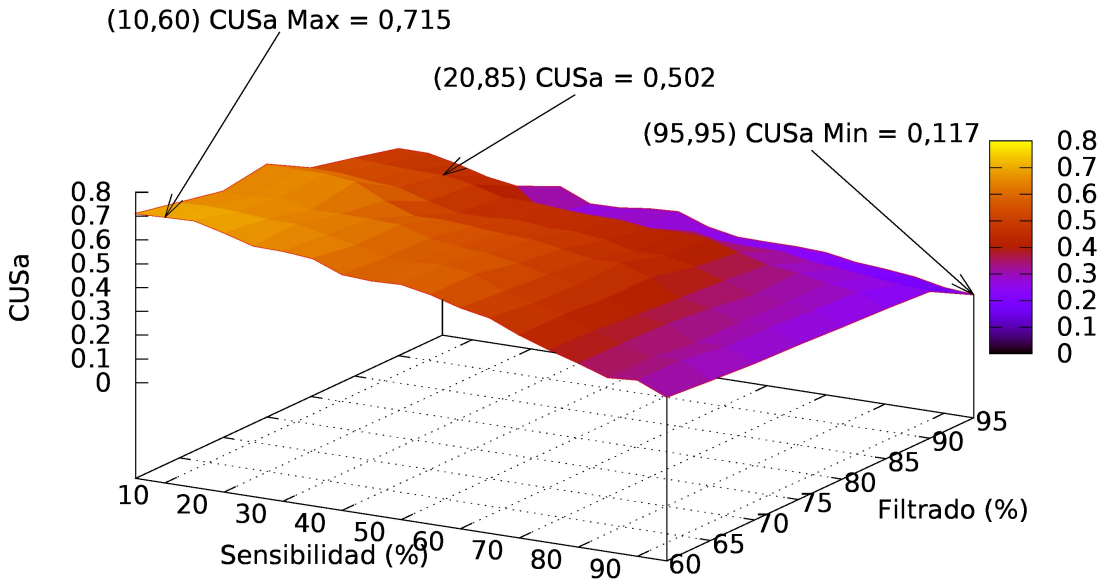

Figura 4.20 Resultados de CUSa para SUMM-SURF usando el método OSM y los usuarios provistos por OSM

exploración de los resultados, se obtuvo un valor máximo de Kappa de 0.409 para los valores de sensibilidad 20 % y filtrado 85 %.

La Figura 4.20 muesta los valores tomados por la métrica CUSa para los distintos valores explorados. De una forma equivalente, los valores de CUSe son mostrados en la Figura 4.21. Los valores del valor-F se muestran en la Figura 4.22. Finalmente, los valores del coeficiente Kappa para SUMM-SURF con variación de parámetros pueden ser vistos en la figura 4.23.

En la Tabla 4.9 se muestran los resultados obtenidos con el método OSM y los nuevos cuadros de referencia. En este caso se observa que las sumarizaciones producidas por SUMM-SURF obtienen la mejor calificación. Es importante destacar al menos dos aspectos relevantes. El primer aspecto para destacar es que no existen disparidades tan marcadas por los resultados mostrados por CUS. Además es importante destacar que las sumarizaciones que obtenían mejores resultados quedan relegadas con el nuevo método de referencia. Los resultados producidos por SUMM-SURF usados para realizar estos experimentos están disponibles en un sitio de Internet con acceso público [7].

[7] http://www.javieriparraguirre.net/online-summarization/

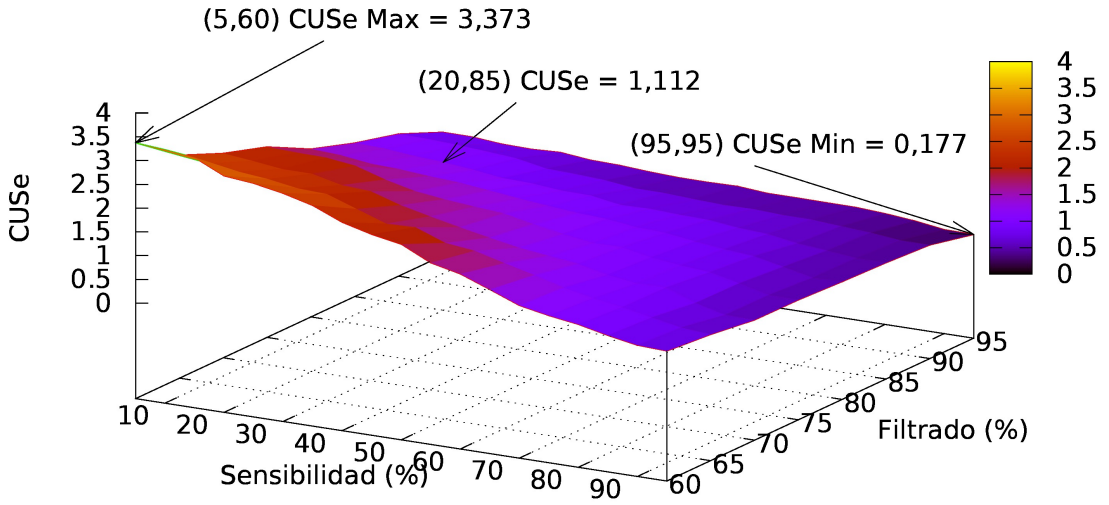

Figura 4.21 Resultados de CUSe para SUMM-SURF usando el método OSM y los usuarios provistos por OSM

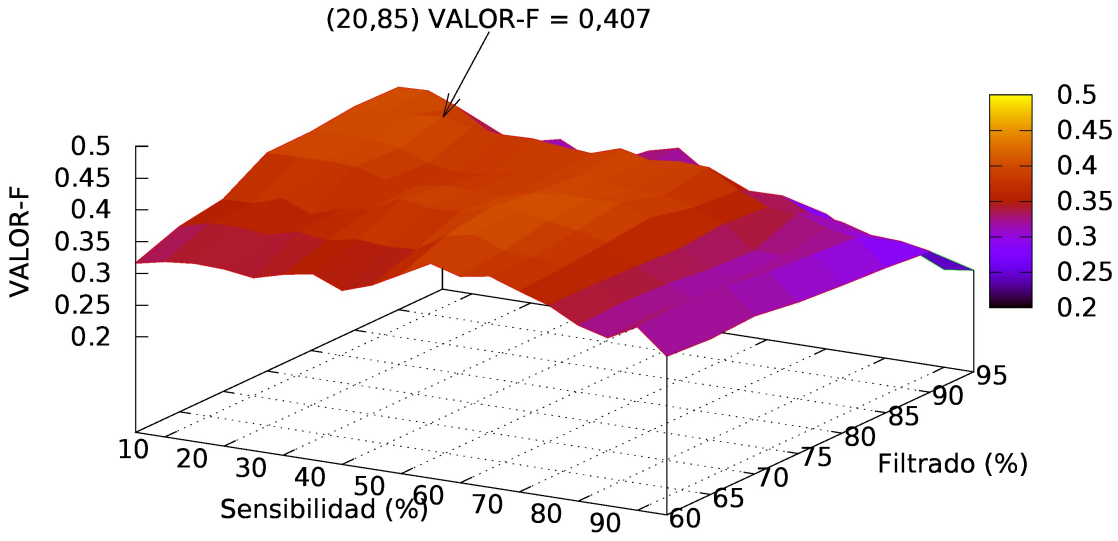

Figura 4.22 Resultados de Valor-F para SUMM-SURF usando el método OSM y los usuarios provistos por OSM

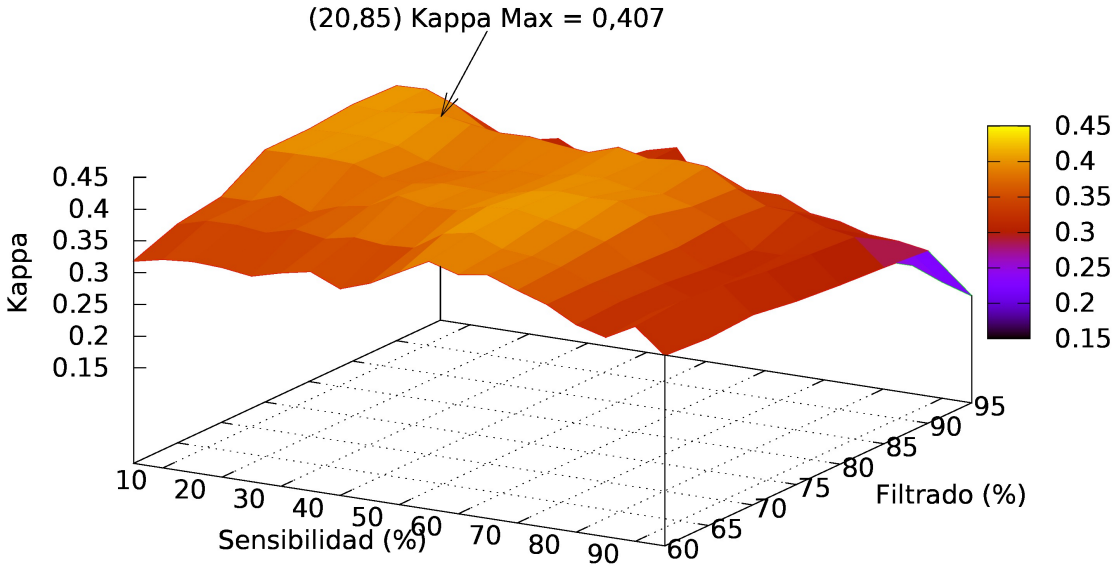

Figura 4.23 Resultados de coeficiente Kappa de Cohen para SUMM-SURF usando el método OSM y los usuarios provistos por OSM

Tabla 4.9 Resultados cuantitativos: método OSM, usuarios OSM. Sensibilidad 20, filtrado 85.

Método	Doble	Δ	*e*	CUSa	CUSe	F	Kappa
DT	SI	60	0.4	0.306	0.221	0.379	0.378
OV	SI	60	0.4	0.341	0.525	0.348	0.346
STIMO	SI	60	0.4	0.323	0.655	0.325	0.324
VSUMM1	SI	60	0.4	0.338	0.631	0.346	0.348
VSUMM2	SI	60	0.4	0.210	0.519	0.255	0.230
VISON	SI	60	0.4	0.309	0.687	0.308	0.308
SUMM-SURF	SI	60	0.4	0.502	1.112	0.407	**0.409**

4.5. Discusión de resultados

En este capítulo se han presentado los resultados correspondientes a SUMM-SURF. Los resultados se presentaron en forma cualitativa y en forma cuantitativa. El foco del análisis presente se centra en los resultados cuantitativos. Los experimentos realizados involucraron dos metodologías de medición (CUS y OSM). Además se usaron dos grupos de usuarios, los cuales fueron los provistos por CUS y los provistos por OSM.

Los videos utilizados fueron los provistos por CUS. No se eligieron videos que no estén contenidos en el conjunto de datos provistos por CUS. Esta decisión se basa en la necesidad de poder comparar la calidad de la sumarizaciones con trabajos previos.

Las métricas obtenidas en los resultados cuantitativos se pueden agrupar en tres casos. En el primer caso se siguió la metodología propuesta por CUS y lo cuadros clave elegidos por los usuarios provistos por CUS. En el segundo grupo de evaluaciones se siguió la metodología OSM con los cuadros seleccionados por usuarios de CUS. Finalmente, se presentaron las métricas obtenidas luego de usar OSM con los usuarios OSM.

Cuando se utilizó la metodología propuesta por CUS y los usuarios provistos por CUS, se encontró que VSUMM es el mejor método de sumarización. SUMM-SURF fue el método de sumarizacón que tuvo la peor puntuación en términos de valor-F. El resto de los algoritmos analizados recibieron calificaciones más cercanas a VSUMM, pero no pudieron superarlo. En la Tabla 4.1 se pueden observar los resultados mencionados.

Cuando se utilizó la metodología OSM y los cuadros de los usuarios CUS, se obtuvo una distribución de las métricas sensiblemente diferente. Aunque VSUMM1 fue la propuesta que obtuvo la mejor puntuación, el resto de los métodos se acercaron notablemente en términos de puntuación. De hecho, VSUMM2 fue superado. SUMM-SURF no obtuvo la peor calificación en este caso y se acercó en términos de puntuación al resto de las propuestas. Los resultados de este experimento se muestran en Tabla 4.8.

En el caso de las métricas arrojadas por la metodología OSM usando los usuarios OSM, se puede ver que SUMM-SURF supera al resto de los métodos de sumarización. La Tabla 4.9 muestra las calificaciones obtenidas. El segundo mejor algoritmo fué DT. VSUMM1 y VSUMM2 quedaron entre los que recibieron peor puntuación.

Es notable el cambio de puntuación de VSUMM usando OSM. Una explicación posible a esa variación es que VSUMM realiza un agrupamiento de cuadros, y selecciona el más representativo del grupo como cuadro clave. Generalmente, los primeros cuadros del grupo no son los elegidos como el más relevante. Devido a que OSM tiene en cuenta la posición del cuadro clave, VSUMM no obtiene la mejor calificación.

De los resultados obtenidos es posible inferir que SUMM-SURF es la propuesta que mejor se desempeña con la metodología OSM. Además, se puede ver que la distribución de las calificaciones de OSM es "pareja" en el rango obtenido. Las métricas obtenidas son consistentes para Kappa y valor-F. Se puede concluir que tanto SUMM-SUR como OSM superan el aporte relevado en la literatura a la fecha de realización de este trabajo.

4.6. Evaluación de desempeño

El algorimo propuesto tiene grandes ventajas en términos del desempeño. Aunque el foco del presente trabajo no ha sido optimizar el algortimo para que maximice el desempeño, el diseño permite escalar y se cuenta con una versión que sumariza en tiempo real para bajas resoluciones del video.

El hecho de que se use SURF como componente principal del desarrollo, permite utilizar desarrollos existentes que facilitan alta velocidad en la sumarización. Es posible encontrar en la literatura implementaciones de SURF en FPGA [124]. También se pueden encontrar optimizaciones para dispoitivos móviles [140]. Desde hace un tiempo se cuentan con versiones de SURF que se ejecutan sobre procesadores gráficos (GPUs) [96].

En varios casos se logra ejecutar SURF con frecuencias mayores a 30 cuadros por segundos en videos en alta definición (1920 x 1080). A partir de resultados reportados en trabajos científicos, es posible inferir que no representa una gran dificultad optimizar SUMM-SURF para ser ejecutado en tiempo real sobre HD (alta definición). Este hecho permite gran aplicación al desarrollo presentado. En el siguietne capítulo se explora con más detalle los desarrollos futuros y se realiza una comparación en términos de costos computacionales.

Capítulo 5

Aporte de este trabajo, desarrollo futuro y conclusiones

En este capítulo se mencionan las ventajas de SUMM-SURF y se destacan los aportes de la propuesta. En términos de las clasificaciones propuestas en el Capítulo 2, SUMM-SURF es un método que produce cuadros clave, trabaja con video descomprimido, realiza sumarizaciones en línea y no demanda de la totalidad del conjunto de cuadros para producir resultados.

5.1. Comparación con trabajos relacionados

SUMM-SURF presenta características que lo hacen particular y novedoso respecto a los aportes similares encontrados en la literatura. Uno de los aspectos novedosos es que puede producir cuadros clave y video condensado al mismo tiempo. Esta ventaja no es reportada por ninguno de los métodos presentados en la literatura.

Entre los aspecto comunes, se puede mencionar que SUMM-SURF tiene similitudes con el aporte de Guan et. al. [53]. Ambos trabajos hacen uso de las características locales para realizar la sumarización. Sin embargo, la propuesta de Guan et. al. [53] demanda la totalidad del video antes de producir resultados. Este hecho se basa en la necesidad de agrupar los cuadros para luego clasificarlos. Otro factor diferencial, es que la popuesta de Guan et. al. [53] propone un submuestreo de los datos para reducir el costo computacional. SUMM-SURF solo demanda unos pocos cuadros para comenzar a producir resultados y no realiza submuestreo.

En un trabajo posterior, Guan et. al. [54] proponen una nueva versión del algoritmo. Aunque siguen usando características locales y ese es un aspecto común con SUMM-SURF,

Tabla 5.1 Resumen de las principales ventajas de los trabajos relacionados.

Método	Formato	Demanda de video
Kleban et al. [71]	Sin comprimir	Completo
Pan et al. [103]	Sin comprimir	Completo
Le and Satoh [74]	Sin comprimir	Completo
Putpuek et al. [112]	Sin comprimir	Completo
Bredin et al. [19]	Sin comprimir	Completo
Chasanis et al. [23]	Sin comprimir	Completo
Besiris et al. [18]	Sin comprimir	Completo
Guan et al. [53]	Sin comprimir	Completo
Guan et al. [54]	Sin comprimir	Completo
Murdur et al. [98] (DT)	Sin comprimir	Completo
Furini et al. [45] (VISTO)	Sin comprimir	Completo
Furini et al. [46] (SITMO)	Sin comprimir	Completo
Fontes de Avila et al. [35] (VSUMM)	Sin comprimir	Completo
Chew and Kankanhalli [25]	Comprimido	Completo
Peker and Divakaran [105]	Comprimido	Completo
Benini et al. [17]	Comprimido	Completo
Herranz and Martínez [61]	Comprimido	Completo
Almeida et al. [5][4](VISON)	Comprimido	Parcial
SUMM-SURF	Sin comprimir	Parcial

crean un conjunto global de cuadros clave para ser usado en la sumarización. Nuevamente, el uso de puntos clave en forma global demanda el uso del video completo.

Otra propuesta que tiene enfoques comunes con SUMM-SURF es el trabajo de Almeida et. al. [4, 5] (VISON). El algoritmo presentado trabaja en línea al igual que lo hace SUMM-SURF. Sin embargo, VISON hace uso de histogramas de colores para representar un cuadro del video. Este es un ejemplo de caracterísitca global y es un claro diferencial respecto a SUMM-SURF. Además VISON hace uso de videos comprimidos. Este hecho limita las posibilidades de análisis a partir de la división en bloques que realizan los métodos de compresión MPEG. SUMM-SURF no tiene este limitante.

La Tabla 5.1 presenta un resúmen de los aspectos distintivos de los trabajos relacionados más relevantes. Además, se muestra cómo SUMM-SURF se relaciona con las propuestas de terceros. Se puede observar por cada uno de los métodos el formato de video utilizado y si demanda de la totalidad del conjunto de datos para producir resultado. De la evidencia presentada, se puede concluir que SUMM-SURF es el único algoritmo que trabaja con formato descomprimido, tiene demanda de video parcial y produce sumarizaciones en línea.

Tabla 5.2 Costo computacional y requerimientos de espacio

Método	Costo computacional	Requerimientos de espacio
DT	$O(n*log(n))$	$O(nd)$
STIMO	$O(nk)$	$O(nd)$
VSUMM	$O(nk)$	$O(nd)$
VISON [4, 5]	$O(n)$	$O(d)$
SUMM-SURF	$O(n)$	$O(d)$

La Tabla 5.2 muestra el costo computacional y los requerimientos de espacio para los métodos comparados, donde n representa la cantidad de cuadros en el video, k indica la cantidad de cuadros en la sumarización producida y d es la dimensión del vector de características usadas. Debido a que pocos autores publican este tipo de infrormación, no es posible comparar con una mayor cantidad de métodos. Se puede observar que SUMM-SURF tiene características comparables a los métodos más destacados.

5.2. Limitaciones y beneficios de la propuesta

Nuestra propuesta para la sumarización de video tiene dos grandes ventajas. La primera de ellas es que toma en cuenta las características locales de las imágenes. Este hecho permite tener la capacidad de detectar el mínimo cambio entre de dos cuadros. En el caso de usar características globales, un cambio menor dentro de la escena pasa inadvertido a la detección. Un típico caso son los videos de vigilancia. En estos casos es importante detectar un cambio de forma o una pequeña rotación de un objeto en la escena.

La segunda ventaja que posee nuestro método es la de poder procesar el video en línea (es decir, sobre la secuencia de video en tiempo real). No es necesario contar con el total del contenido del video para detectar cuadros clave. En nuestra propuesta, basta solo con unos pocos cuadros para poder producir resultados. El uso de SURF permite desempeños aceptables para procesar video a velocidades del orden de tiempo real. En el prócimo capítulo se discutirán los aspectos relacionados al tiempo de procesamiento de nuestro método.

A partir de los resultados obtenidos se pueden mencionar limitaciones en el método propuesto. Una clara limitación es que SUMM-SURF es sensible a movimientos de cámara bruscos tales como por ejemplo cuando la cámara realiza un zoom. Las condiciones mencionadas producen que SUMM-SURF produzca cuadros clave “repetidos” sobre una misma escena. Esto se debe a que las características locales cambian y por lo tanto el algoritmo detecta los cambios y los reporta como cuadros clave. De los resultados

cuantitativos se puede concluir que la limitación no es significativa y SUMM-SURF supera a los métodos existentes en la literatura.

El uso de características locales permite la detección de los mínimos detalles y puede detectar sutilezas. Entre los ejemplos a mencionar son las divisiones dentro de un mapa, o una historia contada en forma monocromática como se pudo ver en la Figura 4.17. En particular el uso de características locales es sumamente útil en videos de vigilancia donde se enfocan grandes áreas y hasta los cambios mínimos pueden tener relevancia.

5.3. Aportes del trabajo

El presente trabajo es una respuesta a los desafíos mencionados en la Sección 2.6. SUMM-SURF es un método general, es capaz de trabajar en línea, posee un costo computacional reducido y produce resultados de calidad igual o superior a las alternativas publicadas según los trabajos revisados a la fecha. Las características mencionadas posicionan a SUMM-SURF como una herramienta ideal para procesar grandes volúmentes de video. Este aporte es inédito y resuelve los desafiíos pendientes.

Además del método de sumarización propuesto, este trabajo hace el aporte de OSM. La metodología planteada en OSM es compatible con métodos existentes y supera el estado del arte. Se hace uso del coeficiente Kappa de Cohen a partir de la necesidad de lidiar con la realidad de que la evaluación de una sumarización de video es un problema de clases desbalanceadas. Se define un marco claro de evaluación y se hace disponible públicamente la implementación.

5.4. Trabajo futuro

Como trabajo futuro se proponen 4 líneas de trabajo. La primer línea de trabajo es implementar SUMM-SURF para ser ejecutado en plataformas computacionales que permitan procesar grandes volúmenes de datos. La segunda línea de trabado es complementar el desarrollo con un sistema de aprendizaje de abstracciones que permita realizar abstracciones de mayor nivel. El tercer objetivo a explorar es el etiquetado (tagging) de video. Finalmente se propone la sumarización multi-perspectiva.

La combinación de las líneas de trabajo tienen como objetivo final poder contar con la capacidad de catalogar e producir inferencias sobre grandes volúmenes de video en forma no-supervisada. Esta herramienta es altamente demandada en la actualidad y tiene un gran número de aplicaciones posibles. Se espera poder producir avances de impacto y aportar nuevas soluciones a demandas claramente planteadas.

5.5. Conclusiones

Se ha presentado una propuesta de summarización de video novedosa. Luego de realizar un análisis exahustivo del estado del arte se puede conluir que SUMM-SURF presenta un significativo aporte. El método es novedoso, y produce resultados cuantitativos iguales o mejores a las alternativas existentes.

Se ha propuesto una solución al gran problema de la evaluación de sumarizaciones. OSM se basa en un desarrollo existene y mejora la metodología usada hasa ahora. Se mantienen métricas compatibles que permite evaluar todos los resultados disponibles.

El trabajo tiene gran potencial de aplicación y presenta varias líenas de trabajo para seguir transitando. La realidad de disponer grandes volúmenes de datos demanda este tipo de propuestas y se espera un impacto significativo a partir de los resultados documentados en este trabajo.

Bibliografía

[1] Lalitha Agnihotri, Nevenka Dimitrova, and John R Kender. Design and evaluation of a music video summarization system. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, volume 3, pages 1943–1946. IEEE, 2004.

[2] Lalitha Agnihotri, John Kender, Nevenka Dimitrova, and John Zimmerman. Framework for personalized multimedia summarization. In *Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pages 81–88. ACM, 2005.

[3] Kiyoharu Aizawa, Kenichiro Ishijima, and Makoto Shiina. Summarizing wearable video. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 3, pages 398–401. IEEE, 2001.

[4] Jurandy Almeida, RDS Torres, and Neucimar J Leite. Rapid video summarization on compressed video. In *Multimedia (ISM), 2010 IEEE International Symposium on*, pages 113–120. IEEE, 2010.

[5] Jurandy Almeida, Neucimar J Leite, and Ricardo da S Torres. Online video summarization on compressed domain. *Journal of Visual Communication and Image Representation*, 2012.

[6] Jurandy Almeida, Neucimar J Leite, and Ricardo da S Torres. Vison: Video summarization for online applications. *Pattern Recognition Letters*, 33(4):397–409, 2012.

[7] Aya Aner and John R Kender. Video summaries through mosaic-based shot and scene clustering. In *Computer Vision—ECCV 2002*, pages 388–402. Springer, 2002.

[8] Yasuo Ariki, Masahito Kumano, and Kiyoshi Tsukada. Highlight scene extraction in real time from baseball live video. In *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 209–214. ACM, 2003.

[9] Jürgen Assfalg, Marco Bertini, Carlo Colombo, Alberto Del Bimbo, and Walter Nunziati. Semantic annotation of soccer videos: automatic highlights identification. *Computer Vision and Image Understanding*, 92(2):285–305, 2003.

[10] Yannis S Avrithis, Anastasios D Doulamis, Nikolaos D Doulamis, and Stefanos D Kollias. A stochastic framework for optimal key frame extraction from mpeg video databases. *Computer Vision and Image Understanding*, 75(1):3–24, 1999.

[11] Noboru Babaguchi. Towards abstracting sports video by highlights. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 3, pages 1519–1522. IEEE, 2000.

[12] Noboru Babaguchi, Yoshihiko Kawai, and Tadahiro Kitahashi. Generation of personalized abstract of sports video. In *ICME*, 2001.

[13] Jeffrey R Bach, Charles Fuller, Amarnath Gupta, Arun Hampapur, Bradley Horowitz, Rich Humphrey, Ramesh C Jain, and Chiao-Fe Shu. Virage image search engine: an open framework for image management. In *Electronic Imaging: Science & Technology*, pages 76–87. International Society for Optics and Photonics, 1996.

[14] Amit Bagga, Jianying Hu, Jialin Zhong, and Ganesh Ramesh. Multi-source combined-media video tracking for summarization. In *Pattern Recognition, International Conference on*, volume 2, pages 20818–20818. IEEE Computer Society, 2002.

[15] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer, 2006.

[16] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[17] Sergio Benini, Pierangelo Migliorati, and Riccardo Leonardi. Hidden markov models for video skim generation. In *Image Analysis for Multimedia Interactive Services, 2007. WIAMIS'07. Eighth International Workshop on*, pages 6–6. IEEE, 2007.

[18] Dimitrios Besiris, A Makedonas, George Economou, and Spiros Fotopoulos. Combining graph connectivity & dominant set clustering for video summarization. *Multimedia Tools and Applications*, 44(2):161–186, 2009.

[19] Hervé Bredin, Daragh Byrne, Hyowon Lee, Noel E O'Connor, and Gareth JF Jones. Dublin city university at the trecvid 2008 bbc rushes summarisation task. In *Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*, pages 45–49. ACM, 2008.

[20] Rui Cai, Lie Lu, Hong-Jiang Zhang, and Lian-Hong Cai. Highlight sound effects detection in audio stream. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 3, pages III–37. IEEE, 2003.

[21] Hyun Sung Chang, Sanghoon Sull, and Sang Uk Lee. Efficient video indexing scheme for content-based retrieval. *Circuits and Systems for Video Technology, IEEE Transactions on*, 9(8):1269–1279, 1999.

[22] Peng Chang, Mei Han, and Yihong Gong. Extract highlights from baseball game video with hidden markov models. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–609. IEEE, 2002.

[23] Vasileios Chasanis, Aristidis Likas, and Nikolaos Galatsanos. Video rushes summarization using spectral clustering and sequence alignment. In *Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*, pages 75–79. ACM, 2008.

[24] Shu-Ching Chen, Mei-Ling Shyu, Min Chen, and Chengcui Zhang. A decision tree-based multimodal data mining framework for soccer goal detection. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, volume 1, pages 265–268. IEEE, 2004.

[25] Chorng-Meng Chew and Mohan S Kankanhalli. Compressed domain summarization of digital video. In *Advances in Multimedia Information Processing—PCM 2001*, pages 490–497. Springer, 2001.

[26] Nancy A Chinchor, James J Thomas, Pak Chung Wong, Michael G Christel, and William Ribarsky. Multimedia analysis+ visual analytics= multimedia analytics. *Computer Graphics and Applications, IEEE*, 30(5):52–60, 2010.

[27] Patrick Chiu, Andreas Girgensohn, and Qiong Liu. Stained-glass visualization for highly condensed video summaries. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, volume 3, pages 2059–2062. IEEE, 2004.

[28] Michael G Christel, Michael A Smith, C Roy Taylor, and David B Winkler. Evolving video skims into useful multimedia abstractions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 171–178. ACM Press/Addison-Wesley Publishing Co., 1998.

[29] Michael G Christel, Alexander G Hauptmann, Adrienne S Warmack, and Scott A Crosby. Adjustable filmstrips and skims as abstractions for a digital video library. In *Research and Technology Advances in Digital Libraries, 1999. Proceedings. IEEE Forum on*, pages 98–104. IEEE, 1999.

[30] François Coldefy and Patrick Bouthemy. Unsupervised soccer video abstraction based on pitch, dominant color and camera motion analysis. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 268–271. ACM, 2004.

[31] François Coldefy, Patrick Bouthemy, Michael Betser, and Guillaume Gravier. Tennis video abstraction from audio and visual cues. In *Multimedia Signal Processing, 2004 IEEE 6th Workshop on*, pages 163–166. IEEE, 2004.

[32] Matthew Cooper and Jonathan Foote. Summarizing video using non-negative similarity matrix factorization. In *Multimedia Signal Processing, 2002 IEEE Workshop on*, pages 25–28. IEEE, 2002.

[33] Matthew Cooper and Jonathan Foote. Discriminative techniques for keyframe selection. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 4–pp. IEEE, 2005.

[34] Serhan Dagtas and Mohamed Abdel-Mottaleb. Multimodal detection of highlights for multimedia content. *Multimedia Systems*, 9(6):586–593, 2004.

[35] Sandra Eliza Fontes de Avila, Ana Paula Brandão Lopes, et al. Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 32(1):56–68, 2011.

[36] Daniel DeMenthon, Vikrant Kobla, and David Doermann. Video summarization by curve simplification. In *Proceedings of the sixth ACM international conference on Multimedia*, pages 211–218. ACM, 1998.

[37] F Dirfaux. Key frame selection to represent a video. In *Image Processing, 2000. Proceedings. 2000 International Conference on*, volume 2, pages 275–278. IEEE, 2000.

[38] Ajay Divakaran, Kadir A Peker, Regunathan Radhakrishnan, Ziyou Xiong, and Romain Cabasson. Video summarization using mpeg-7 motion activity and audio descriptors. In *Video Mining*, pages 91–121. Springer, 2003.

[39] Anastasios D Doulamis, Nikolaos D Doulamis, and Stefanos D Kollias. A fuzzy video content representation for video summarization and content-based retrieval. *Signal Processing*, 80(6):1049–1067, 2000.

[40] N Doulamis, A Doulamis, Yannis S Avrithis, and Stefanos D Kollias. Video content representation using optimal extraction of frames and scenes. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, volume 1, pages 875–879. IEEE, 1998.

[41] Berna Erol, D-S Lee, and Jonathan Hull. Multimodal summarization of meeting recordings. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 3, pages III–25. IEEE, 2003.

[42] Brigitte Fauvet, Patrick Bouthemy, Patrick Gros, and Fabien Spindler. A geometrical key-frame selection method exploiting dominant motion estimation in video. In *Image and Video Retrieval*, pages 419–427. Springer, 2004.

[43] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, et al. Query by image and video content: The qbic system. *Computer*, 28(9):23–32, 1995.

[44] Charles Frankel, Michael J Swain, and Vassilis Athitsos. Webseer: An image search engine for the world wide web. 1996.

[45] Marco Furini, Filippo Geraci, Manuela Montangero, and Marco Pellegrini. Visto: visual storyboard for web video browsing. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 635–642. ACM, 2007.

[46] Marco Furini, Filippo Geraci, Manuela Montangero, and Marco Pellegrini. Stimo: Still and moving video storyboard for the web scenario. *Multimedia Tools and Applications*, 46(1):47–69, 2010.

[47] David Gibson, Neill Campbell, and Barry Thomas. Visual abstraction of wildlife footage using gaussian mixture models and the minimum description length criterion. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 2, pages 814–817. IEEE, 2002.

[48] Andreas Girgensohn. A fast layout algorithm for visual video summaries. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 2, pages II–77. IEEE, 2003.

[49] Andreas Girgensohn and John Boreczky. Time-constrained keyframe selection technique. In *Multimedia Computing and Systems, 1999. IEEE International Conference on*, volume 1, pages 756–761. IEEE, 1999.

[50] Yihong Gong and Xin Liu. Video summarization and retrieval using singular value decomposition. *Multimedia Systems*, 9(2):157–168, 2003.

[51] Yihong Gong, Xin Liu, and W Hua. Summarizing video by minimizing visual content redundancies. In *ICME*, 2001.

[52] Lifang Gu, Don Bone, and Graham Reynolds. Replay detection in sports video sequences. In *Multimedia'99*, pages 3–12. Springer, 2000.

[53] Genliang Guan, Zhiyong Wang, Kaimin Yu, Shaohui Mei, Mingyi He, and Dagan Feng. Video summarization with global and local features. In *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*, pages 570–575. IEEE, 2012.

[54] Genliang Guan, Zhiyong Wang, Shiyang Lu, Jeremiah Da Deng, and David Dagan Feng. Keypoint based keyframe selection. 2013.

[55] Frank R Hampel, Elvezio M Ronchetti, Peter J Rousseeuw, and Werner A Stahel. *Robust statistics: the approach based on influence functions*, volume 114. John Wiley & Sons, 2011.

[56] Mei Han, Wei Hua, Wei Xu, and Yihong Gong. An integrated baseball digest system using maximum entropy method. In *Proceedings of the tenth ACM international conference on Multimedia*, pages 347–350. ACM, 2002.

[57] Seung-Hoon Han and In-So Kweon. Scalable temporal interest points for abstraction and classification of video events. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 4–pp. IEEE, 2005.

[58] Alan Hanjalic. Generic approach to highlights extraction from a sport video. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 1, pages I–1. IEEE, 2003.

[59] Alan Hanjalic, Reginald L Lagendijk, and Jan Biemond. Automated high-level movie segmentation for advanced video-retrieval systems. *Circuits and Systems for Video Technology, IEEE Transactions on*, 9(4):580–588, 1999.

[60] Liwei He, Elizabeth Sanocki, Anoop Gupta, and Jonathan Grudin. Auto-summarization of audio-video presentations. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 489–498. ACM, 1999.

[61] Luis Herranz and José M Martínez. An efficient summarization algorithm based on clustering and bitstream extraction. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, pages 654–657. IEEE, 2009.

[62] Michael E Houle. The relevant-set correlation model for data clustering. *Statistical Analysis and Data Mining*, 1(3):157–176, 2008.

[63] Weiming Hu, Nianhua Xie, Li Li, Xianglin Zeng, and Stephen Maybank. A survey on visual content-based video indexing and retrieval. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 41(6):797–819, 2011.

[64] Tiejun Huang. Surveillance video: the biggest big data. *Computing Now*, 7(2), 2014.

[65] Javier Iparraguirre and Claudio Delrieux. Procesamiento y reconocimiento de patrones en video digital. In *XII Workshop de Investigadores en Ciencias de la Computación*, 2010.

[66] Javier Iparraguirre and Claudio A. Delrieux. Speeded-up video summarization based on local features. In *Multimedia (ISM), 2013 IEEE International Symposium on*, pages 370–373. IEEE, 2013.

[67] Javier Iparraguirre and Claudio A Delrieux. Online video summarization based on local features. *International Journal of Multimedia Data Engineering and Management (IJMDEM)*, 5(2):41–53, 2014.

[68] Michal Irani and P Anandan. Video indexing based on mosaic representations. *Proceedings of the IEEE*, 86(5):905–921, 1998.

[69] Hong-Wen Kang and Xian-Sheng Hua. To learn representativeness of video frames. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 423–426. ACM, 2005.

[70] Mohan S Kankanhalli and Joo-Hwee Lim. *Perspectives on content-based multimedia systems*, volume 9. Springer, 2000.

[71] Jim Kleban, Anindya Sarkar, Emily Moxley, Stephen Mangiat, Swapna Joshi, Thomas Kuo, and BS Manjunath. Feature fusion and redundancy pruning for rush video summarization. In *Proceedings of the international workshop on TRECVID video summarization*, pages 84–88. ACM, 2007.

[72] Anita Komlodi and Gary Marchionini. Key frame preview techniques for video browsing. In *Proceedings of the third ACM conference on Digital libraries*, pages 118–125. ACM, 1998.

[73] Longin Jan Latecki, Daniel de Wildt, and Jianying Hu. Extraction of key frames from videos by optimal color composition matching and polygon simplification. In *Multimedia Signal Processing, 2001 IEEE Fourth Workshop on*, pages 245–250. IEEE, 2001.

[74] Duy-Dinh Le and Shin'ichi Satoh. National institute of informatics, japan at trecvid 2007: Bbc rushes summarization. In *Proceedings of the international workshop on TRECVID video summarization*, pages 70–73. ACM, 2007.

[75] Hun-Cheol Lee and Seong-Dae Kim. Rate-driven key frame selection using temporal variation of visual content. *Electronics Letters*, 38(5):217–218, 2002.

[76] Hun-Cheol Lee and Seong-Dae Kim. Iterative key frame selection in the rate-constraint environment. *Signal Processing: Image Communication*, 18(1):1–15, 2003.

[77] Hyowon Lee, Alan F Smeaton, Catherine Berrut, Noel Murphy, Seán Marlow, and Noel E O'Connor. Implementation and analysis of several keyframe-based browsing interfaces to digital video. In *Research and Advanced Technology for Digital Libraries*, pages 206–218. Springer, 2000.

[78] Ming-Chieh Lee, Wei-Ge Chen, Chih-lung Bruce Lin, Chuang Gu, Tomislav Markoc, Steven I Zabinsky, and Richard Szeliski. A layered video object coding system using sprite and affine motion model. *Circuits and Systems for Video Technology, IEEE Transactions on*, 7(1):130–145, 1997.

[79] Shih-Hung Lee, Chia H Yeh, and C-CJ Kuo. Video skimming based on story units via general tempo analysis. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, volume 2, pages 1099–1102. IEEE, 2004.

[80] Michael S Lew, Nicu Sebe, Chabane Djeraba, and Ramesh Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 2(1): 1–19, 2006.

[81] Ying Li, Shrikanth Narayanan, and C-C Jay Kuo. Movie content analysis, indexing and skimming via multimodal information. In *Video mining*, pages 123–154. Springer, 2003.

[82] Zhu Li, Guido M Schuster, Aggelos K Katsaggelos, and Bhavan Gandhi. Optimal video summarization with a bit budget constraint. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*, volume 1, pages 617–620. IEEE, 2004.

[83] Wen-Nung Lie and Chun-Ming Lai. News video summarization based on spatial and motion feature analysis. In *Advances in multimedia information processing-PCM 2004*, pages 246–255. Springer, 2005.

[84] Rainer Lienhart, Silvia Pfeiffer, and Wolfgang Effelsberg. Video abstracting. *Communications of the ACM*, 40(12):54–62, 1997.

[85] Rainer W Lienhart. Dynamic video summarization of home video. In *Electronic Imaging*, pages 378–389. International Society for Optics and Photonics, 1999.

[86] Tianming Liu, Hong-Jiang Zhang, and Feihu Qi. A novel video key-frame-extraction algorithm based on perceived motion energy model. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(10):1006–1013, 2003.

[87] Tiecheng Liu and John R Kender. An efficient error-minimizing algorithm for variable-rate temporal video sampling. In *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on*, volume 1, pages 413–416. IEEE, 2002.

[88] Tieyan Liu, Xudong Zhang, Jian Feng, and Kwok-Tung Lo. Shot reconstruction degree: a novel criterion for key frame selection. *Pattern recognition letters*, 25(12): 1451–1457, 2004.

[89] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[90] Shi Lu, Irwin King, and Michael R Lyu. Video summarization by video structure analysis and graph optimization. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, volume 3, pages 1959–1962. IEEE, 2004.

[91] Shi Lu, Michael R Lyu, and Irwin King. Video summarization by spatial-temporal graph optimization. In *Circuits and Systems, 2004. ISCAS'04. Proceedings of the 2004 International Symposium on*, volume 2, pages II–197. IEEE, 2004.

[92] Shi Lu, Michael R Lyu, and Irwin King. Semantic video summarization using mutual reinforcement principle and shot arrangement patterns. In *Multimedia Modelling Conference, 2005. MMM 2005. Proceedings of the 11th International*, pages 60–67. IEEE, 2005.

[93] Yu-Fei Ma, Lie Lu, Hong-Jiang Zhang, and Mingjing Li. A user attention model for video summarization. In *Proceedings of the tenth ACM international conference on Multimedia*, pages 533–542. ACM, 2002.

[94] Ken Masumitsu and Tomio Echigo. Video summarization using reinforcement learning in eigenspace. In *Image Processing, 2000. Proceedings. 2000 International Conference on*, volume 2, pages 267–270. IEEE, 2000.

[95] Tao Mei, Cai-Zhi Zhu, He-Qin Zhou, and Xian-Sheng Hua. Spatio-temporal quality assessment for home videos. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 439–442. ACM, 2005.

[96] Perhaad Mistry, Chris Gregg, Norman Rubin, David Kaeli, and Kim Hazelwood. Analyzing program flow within a many-kernel opencl application. In *Proceedings of the Fourth Workshop on General Purpose Processing on Graphics Processing Units*, page 10. ACM, 2011.

[97] Arthur G Money and Harry Agius. Video summarisation: A conceptual framework and survey of the state of the art. *Journal of Visual Communication and Image Representation*, 19(2):121–143, 2008.

[98] Padmavathi Mundur, Yong Rao, and Yelena Yesha. Keyframe-based video summarization using delaunay clustering. *International Journal on Digital Libraries*, 6(2):219–232, 2006.

[99] Haung Wei Ng, Yasuhito Sawahata, and Kiyoharu Aizawa. Summarization of wearable videos using support vector machine. In *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on*, volume 1, pages 325–328. IEEE, 2002.

[100] Chong-Wah Ngo, Yu-Fei Ma, and HongJiang Zhang. Automatic video summarization by graph modeling. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 104–109. IEEE, 2003.

[101] Nosa Omoigui, Liwei He, Anoop Gupta, Jonathan Grudin, and Elizabeth Sanocki. Time-compression: systems concerns, usage, and benefits. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 136–143. ACM, 1999.

[102] Jian-quan Ouyang, Jin-Tao Li, and Yong-Dong Zhang. Replay boundary detection in mpeg compressed video. In *Machine Learning and Cybernetics, 2003 International Conference on*, volume 5, pages 2800–2804. IEEE, 2003.

[103] Chen-Ming Pan, Yung-Yu Chuang, and Winston H Hsu. Ntu trecvid-2007 fast rushes summarization system. In *Proceedings of the international workshop on TRECVID video summarization*, pages 74–78. ACM, 2007.

[104] Hao Pan, P Van Beek, and M Ibrahim Sezan. Detection of slow-motion replay segments in sports video for highlights generation. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 3, pages 1649–1652. IEEE, 2001.

[105] Kadir A Peker and Ajay Divakaran. Adaptive fast playback-based video skimming using a compressed-domain visual complexity measure. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, volume 3, pages 2055–2058. IEEE, 2004.

[106] Kadir A Peker, Ajay Divakaran, and Huifang Sun. Constant pace skimming and temporal sub-sampling of video using motion activity. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 3, pages 414–417. IEEE, 2001.

[107] Nathalie Peyrard and Patrick Bouthemy. Motion-based selection of relevant video segments for video summarization. *Multimedia Tools and Applications*, 26(3):259–276, 2005.

[108] Silvia Pfeiffer, Rainer Lienhart, Stephan Fischer, and Wolfgang Effelsberg. Abstracting digital movies automatically. *Journal of Visual Communication and Image Representation*, 7(4):345–353, 1996.

[109] Arthur Pope, Rakesh Kumar, Harpreet Sawhney, and Charles Wan. Video abstraction: Summarizing video content for retrieval and visualization. In *Signals, Systems & Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on*, volume 1, pages 915–919. IEEE, 1998.

[110] Sarah V Porter, Majid Mirmehdi, and Barry T Thomas. A shortest path representation for video summarisation. In *Image Analysis and Processing, 2003. Proceedings. 12th International Conference on*, pages 460–465. IEEE, 2003.

[111] Yael Pritch, Sarit Ratovitch, Avishai Hendel, and Shmuel Peleg. Clustered synopsis of surveillance video. In *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 195–200. IEEE, 2009.

[112] Narongsak Putpuek, Duy-Dinh Le, Nagul Cooharojananone, Shin'ichi Satoh, and Chidchanok Lursinsap. Rushes summarization using different redundancy elimination approaches. In *Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*, pages 100–104. ACM, 2008.

[113] Regunathan Radhakrishnan, Ajay Divakaran, and Ziyou Xiong. A time series clustering based framework for multimedia mining and summarization using audio features. In *Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 157–164. ACM, 2004.

[114] Yong Rui, Anoop Gupta, and Alex Acero. Automatically extracting highlights for tv baseball programs. In *Proceedings of the eighth ACM international conference on Multimedia*, pages 105–115. ACM, 2000.

[115] Daniel M Russell. A design pattern-based video summarization technique: moving from low-level signals to high-level structure. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, pages 10–pp. IEEE, 2000.

[116] Huang-Chia Shih and Chung-Lin Huang. Detection of the highlights in baseball video program. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, volume 1, pages 595–598. IEEE, 2004.

[117] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-495-2. doi: http://doi.acm.org/10.1145/1178677.1178722.

[118] John R Smith and Shih-Fu Chang. Visually searching the web for content. *MultiMedia, IEEE*, 4(3):12–20, 1997.

[119] Michael A Smith and Takeo Kanade. Video skimming and characterization through the combination of image and language understanding. In *Content-Based Access of Image and Video Database, 1998. Proceedings., 1998 IEEE International Workshop on*, pages 61–70. IEEE, 1998.

[120] Sanghoon Sull, Jung-Rim Kim, Yunam Kim, Hyun S Chang, and Sang U Lee. Scalable hierarchical video summary and search. In *Photonics West 2001-Electronic Imaging*, pages 553–561. International Society for Optics and Photonics, 2001.

[121] Xinding Sun and Mohan S Kankanhalli. Video summarization using r-sequences. *Real-time imaging*, 6(6):449–459, 2000.

[122] Hari Sundaram and Shih-Fu Chang. Condensing computable scenes using visual complexity and film syntax analysis. In *Proceedings of ICME 2001*. Citeseer, 2001.

[123] Hari Sundaram and Shih-Fu Chang. Video skims: Taxonomies and an optimal generation framework. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 2, pages II–21. IEEE, 2002.

[124] Jan Svab, Tomáš Krajník, Jan Faigl, and Libor Preucil. Fpga based speeded up robust features. In *Technologies for Practical Robot Applications, 2009. TePRA 2009. IEEE International Conference on*, pages 35–41. IEEE, 2009.

[125] Yukinobu Taniguchi, Akihito Akutsu, and Yoshinobu Tonomura. Panoramaexcerpts: extracting and packing panoramas for video browsing. In *Proceedings of the fifth ACM international conference on Multimedia*, pages 427–436. ACM, 1997.

[126] Cuneyt M Taskiran, Arnon Amir, Dulce B Ponceleon, and Edward J Delp III. Automated video summarization using speech transcripts. In *Electronic Imaging 2002*, pages 371–382. International Society for Optics and Photonics, 2001.

[127] Laura Teodosio and Walter Bender. Salient video stills: Content and context preserved. In *Proceedings of the first ACM international conference on Multimedia*, pages 39–46. ACM, 1993.

[128] Dian Tjondronegoro, Yi-Ping Phoebe Chen, and Binh Pham. Integrating highlights for more complete sports video summarization. *IEEE multimedia*, 11(4):22–37, 2004.

[129] Ba Tu Truong and Svetha Venkatesh. Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 3(1):3, 2007.

[130] Ba Tu Truong, Svetha Venkatesh, and Chitra Dorai. Discovering semantics from visualizations of film takes. In *Multimedia Modelling Conference, 2004. Proceedings. 10th International*, pages 109–116. IEEE, 2004.

[131] Shingo Uchihashi, Jonathan Foote, Andreas Girgensohn, and John Boreczky. Video manga: generating semantically meaningful video summaries. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 383–392. ACM, 1999.

[132] Jaco Vermaak, Patrick Pérez, Michel Gangnet, and Andrew Blake. Rapid summarisation and browsing of video sequences. In *BMVC*, pages 1–10, 2002.

[133] Anthony J Viera, Joanne M Garrett, et al. Understanding interobserver agreement: the kappa statistic. *Fam Med*, 37(5):360–363, 2005.

[134] Kongwah Wan and Changsheng Xu. Efficient multimodal features for automatic soccer highlight generation. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 973–976. IEEE, 2004.

[135] Yu-Xiang Xie, Xi-Dao Luan, Song-Yang Lao, Ling-Da Wu, Peng Xiao, and Jun Wen. Edu: A model of video summarization. In *Image and video retrieval*, pages 106–114. Springer, 2004.

[136] Wei Xiong, Chung-Mong Lee, and Rui-Hua Ma. Automatic video data structuring through shot partitioning and key-frame computing. *Machine Vision and Applications*, 10(2):51–65, 1997.

[137] Ziyou Xiong, Regunathan Radhakrishnan, and Ajay Divakaran. Generation of sports highlights using motion activity in combination with a common audio feature extraction framework. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 1, pages I–5. IEEE, 2003.

[138] Ziyou Xiong, Regunathan Radhakrishnan, Ajay Divakaran, and Thomas S Huang. Effective and efficient sports highlights extraction using the minimum description length criterion in selecting gmm structures. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, volume 3, pages 1947–1950. IEEE, 2004.

[139] Itheri Yahiaoui, Bernard Merialdo, and Benoit Huet. Automatic video summarization. In *Proc. CBMIR Conf*, 2001.

[140] Xin Yang and Kwang-Ting Tim Cheng. Accelerating surf detector on mobile devices. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 569–578. ACM, 2012.

[141] Minerva M Yeung and Boon-Lock Yeo. Video visualization for compact presentation and fast browsing of pictorial content. *Circuits and Systems for Video Technology, IEEE Transactions on*, 7(5):771–785, 1997.

[142] Bin Yu, Wei-Ying Ma, Klara Nahrstedt, and Hong-Jiang Zhang. Video summarization based on user log enhanced link analysis. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 382–391. ACM, 2003.

[143] Xiao-Dong Yu, Lei Wang, Qi Tian, and Ping Xue. Multilevel video representation with application to keyframe extraction. In *Multimedia Modelling Conference, 2004. Proceedings. 10th International*, pages 117–123. IEEE, 2004.

[144] Hong Jiang Zhang, Jianhua Wu, Di Zhong, and Stephen W Smoliar. An integrated system for content-based video retrieval and browsing. *Pattern recognition*, 30(4): 643–658, 1997.

[145] Xu-Dong Zhang, Tie-Yan Liu, Kwok-Tung Lo, and Jian Feng. Dynamic selection and effective compression of key frames for video abstraction. *Pattern recognition letters*, 24(9):1523–1532, 2003.

[146] Ming Zhao, Jiajun Bu, and Chun Chen. Audio and video combined for home video abstraction. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 5, pages V–620. IEEE, 2003.

[147] Yueting Zhuang, Yong Rui, Thomas S Huang, and Sharad Mehrotra. Adaptive key frame extraction using unsupervised clustering. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, volume 1, pages 866–870. IEEE, 1998.