



Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA



LIVRE: A Video Extension to the LIRE Content-Based Image Retrieval System

Degree's Final Project Dissertation
Telecommunications Engineering

Author: Gabriel de Oliveira Barra
Advisors: Mathias Lux and Xavier Giró-i-Nieto

Alpen-Adria University of Klagenfurt (AAU Klagenfurt)
Universitat Politècnica de Catalunya (UPC)
2015

Abstract

This project explores the expansion of Lucene Image Retrieval Engine (LIRE), an open-source Content-Based Image Retrieval (CBIR) system, for video retrieval on large scale video datasets. The fast growth of the need to store huge amounts of video in servers requires efficient, scalable search and indexing engines capable to assist users in their management and retrieval. In our tool, queries are formulated by visual examples allowing users to find the videos and the moment of time when the query image is matched with. The video dataset used on this scenario comprise over 1,000 hours of different news broadcast channels.

This thesis presents an extension and adaptation of Lire and its plugin for Solr, an open-source enterprise search platform from the Apache Lucene project, for video retrieval based on visual features, as well as a web-interface for users from different devices.

The results are assessed from two perspectives: a quantitative and qualitative one. While the quantitative one follows the standard practices and metrics for video retrieval, the qualitative assessment has been based on an empirical social study using a semi-interactive web-interface. In particular, a thinking aloud test was applied to analyze if the user expectations and requirements were fulfilled.

Resum

Aquest projecte investiga el potencial de LIRE, una llibreria de codi obert per a consulta d'imatges mitjançant exemple (CBIR), com a eina de consulta i indexació per a grans bases de dades de vídeos. El ràpid creixement de les necessitats d'emmagatzemar enormes quantitats de vídeos en diversos servidors requereix un motor de cerca i indexació escalable i eficient, així com una eina que proporcioni als usuaris mètodes per a la gestió i consulta de contingut. En la nostra eina, les consultes es formulen mitjançant exemples visuals, permetent als usuaris trobar els vídeos i els segons als quals pertany un fotograma en concret. La recopilació de vídeos utilitzats en aquest escenari comprèn més de 1000 hores de vídeo recopilats de diferents canals de notícies internacionals.

En aquesta tesi es presenta una extensió i adaptació de Lire i el seu plugin per Solr, una plataforma de cerca empresarial de codi obert del projecte Apache Lucene, per a la consulta i indexació de vídeos en servidors basat en característiques visuals, així com una interfície web accessible per a usuaris de diferents dispositius.

Els resultats són avaluats des de dues perspectives: una quantitativa i una qualitativa. Mentre que la part quantitativa segueix l'estàndard de les pràctiques i mètriques empleades en video retrieval, l'avaluació qualitativa ha estat basada en un estudi social empíric mitjançant una interfície web semi-interactiva. Particularment, s'ha utilitzat el mètode "thinking aloud test" per analitzar si la nostra eina compleix amb les expectatives i necessitats dels usuaris a l'hora d'utilitzar l'aplicació.

Resumen

Este proyecto investiga el potencial de LIRE, una librería de código abierto para consulta de imágenes mediante ejemplo (CBIR), como herramienta de consulta e indexación para grandes bases de datos de vídeos. El rápido crecimiento de las necesidades de almacenar enormes cantidades de vídeos en diversos servidores requiere un motor de búsqueda e indexación escalable y eficiente, así como una herramienta que proporcione a los usuarios métodos para la gestión y consulta de contenido. En nuestra herramienta, las consultas se formulan mediante ejemplos visuales, permitiendo a los usuarios encontrar los vídeos y los segundos a los que pertenece un fotograma en concreto. La recopilación de vídeos utilizados en este escenario comprende más de 1000 horas de vídeo recopilados de diferentes canales de noticias internacionales.

En esta tesis se presenta una extensión y adaptación de Lire y su plugin para Solr, una plataforma de búsqueda empresarial de código abierto del proyecto Apache Lucene, para la consulta e indexación de vídeos en servidores basado en características visuales, así como una interfaz web accesible para usuarios de diferentes dispositivos.

Los resultados son evaluados desde dos perspectivas: una cuantitativa y una cualitativa. Mientras que la parte cuantitativa sigue el estándar de las prácticas y métricas empleadas en vídeo retrieval, la evaluación cualitativa ha sido basada en un estudio social empírico mediante una interfaz web semi-interactiva. Particularmente, se ha utilizado el método "thinking aloud test" para analizar si nuestra herramienta cumple con las expectativas y necesidades de los usuarios a la hora de utilizar la aplicación.

Key Words

CBVR, Feature Extraction, Video Indexing, Video Retrieval

Acknowledgements

I would like to acknowledge ETSETB for forming me and giving me all the opportunities to develop myself during those last years, not only as an Engineer, but also as a human being. Also to the Alpen-Adria University of Klagenfurt and all the ITEC department for fostering, hosting and mentoring me during those last 6 months.

I would like to express my greatest appreciation to my tutor, Professor Mathias Lux, for giving me this great opportunity, as well as for his warmth welcome to the beautiful Carinthia, teaching me many things and make me feel comfortable in Klagenfurt from the first day. Also to my co-advisor Xavier Giró who, from Barcelona, has been guiding me, stimulating suggestions and welcoming me among his young and promising research group of students. Finally, to my colleagues and friends Nektarios Anagnostopoulos and Jennifer Roldán, whose advises and help were always been very valuable both inside as well as outside the laboratory, and who encouraged me not to be afraid of big challenges.

Last but not least, my thanks goes to my family who has given me the greatest encouragement, and supporting me with every final choice I have ever had, including my departure to Austria and finish my studies.

Danke fur alles. Bis immer.

Agraïments

M'agradaria donar les gràcies a l'ETSETB, per formar-me i oferir-me moltíssimes oportunitats per desenvolupar-me durant aquests últims anys, no només com a enginyer, sinó també com a persona. També a l'Alpen-Adria University of Klagenfurt i a tot el departament ITEC per acollir-me i guiar-me durant aquests últims 6 mesos.

M'agradaria expressar la meva gratitud al meu tutor, el Professor Mathias Lux, per oferir-me aquesta gran oportunitat, així com per la seva rebre'm en Caríntia, per ensenyar-me moltes coses i per fer-me sentir còmode en Klagenfurt des del primer dia. També, agrair al meu company i amic Nektarios Anagnostopoulos i Jennifer Roldán, d'ells qui els consells han estat sempre molt valuosos tant dins com fora del laboratori i els qui m'ha animat a no tenir por a afrontar grans desafiaments.

Finalment, però no per això menys important, agrair especialment a la meua família que m'ha animat i m'ha donat suport amb cada decisió final que he fet, incloent el meu viatge a Àustria i acabar els meus estudis.

Danke fur alles. Bis immer.

Agradecimientos

Me gustaría dar las gracias a la ETSETB, por formarme y ofrecerme muchísimas oportunidades para desarrollarme durante estos últimos años años, no sólo como ingeniero, sino también como persona. También a la Alpen-Adria University of Klagenfurt y a todo el departamento ITEC por acogerme y guiarme durante estos últimos 6 meses.

Me gustaría expresar mi gratitud a mi tutor, el Professor Mathias Lux, por ofrecerme esta gran oportunidad, así como por su recibirme en Carintia, por enseñarme muchas cosas y por hacerme sentir cómodo en Klagenfurt des del primer día. También, agradecer a mi compañeros y amigos Nektarios Anagnostopoulos y Jennifer Roldán, de quienes los consejos han sido siempre muy valiosos tanto dentro como fuera del laboratorio y quienes me ha animado a no tener miedo a afrontar grandes desafíos.

Finalmente, pero no por eso menos importante, agradecer en especial a mi familia que me ha animado y me ha dado soporte con cada decisión final que he hecho, incluyendo mi viaje a Austria y acabar mis estudios.

Danke fur alles. Bis immer.

Contents

1	Introduction	1
1.1	Focus of the Thesis	2
1.2	Motivation	3
1.3	Outline of the Thesis	4
1.4	Work Plan of the Thesis	4
2	Related Work	6
2.1	Related Content-Based Video Retrieval (CBVR) systems	7
3	Requirements	13
3.1	Content requirements	14
3.1.1	Parsing requirements	14
3.1.2	Indexing requirements	15
3.1.3	Retrieval requirements	15
3.2	Evaluation requirements	16
3.2.1	Quantitative evaluation requirements	16
3.2.2	Qualitative evaluation requirements	16
4	Developed solution	17
4.1	Selected global descriptors	17

4.2	Solr as the LlvRE CBVR system search engine	18
4.3	Parsing, Indexing and Retrieval	19
4.3.1	Parsing procedure	20
4.3.2	Indexing procedure	21
4.3.3	Retrieval procedure	22
5	Evaluation	26
5.1	The dataset	26
5.2	Quantitative study	28
5.2.1	Evaluation Metrics	29
5.2.1.1	Scene Retrieval metrics	29
5.2.1.2	Temporal Refinement metrics	29
5.2.2	Results Evaluation	30
5.2.2.1	Scene Retrieval results	31
5.2.2.2	Temporal Refinement results	34
5.2.3	Qualitative user study	36
5.2.3.1	Evaluation Method and Procedure	36
5.2.3.2	Participants	37
5.2.3.3	Results	38
6	Conclusions and Further Work	40

List of Figures

1.1	Original project planning stages.	2
1.2	Gantt chart. The chart shows the calendar and organization of the Thesis.	5
1.3	Proposed schedule from the Gantt chart. The grid shows the expected start date for each task, as well as a guideline for the amount of days and deadlines.	5
2.1	Sample visualization of the Stanford I2V ground-truth. Query image on the left is retrieved in three different videos at their specific time.	7
2.2	Diagram of the Jiku Director system for a scenario where different people are recording the same at the same time but with different devices and in different positions.	8
2.3	Main and second window for the medical semi-interactive interface for endoscopic video retrieval	9
2.4	Digimatge user interface with visual and keyword queries.	9
2.5	Video retrieval interface for the broadcaster archive.	10
2.6	GOS (Graphical Object Searcher) user interface.	11
2.7	MediaMill semantic video search engine browsers.	11

2.8	VIREO-VH Screenshots. Above, galaxy of visual snippets (or technically clusters of graphs). There are several visual snippet structures in this galaxy view. The structure carries semantic meanings. A fully connected snippet (highlighted with green) indicates highly redundant videos, a sparse structure (highlighted with red) implies an evolving event and a highly centralized structure (hub) with excessive hyperlinks to other videos. Below: zooming in a visual snippet, each video is represented as a series of keyframe thumbnails.	12
3.1	Diagram with the global requirements for a CBVR system on a previously indexed dataset. A specific implementation for the LlvRE CBVR, structured with blocks for parsing, indexing and retrieving is presented on Chapter 4.	13
4.1	Diagram of the developed solution for the LlvRE CBVR system with the 3 consecutive blocks: Parsing, Indexing and Retrieving (Querying). These independent blocks are implemented to allow the user to perform CBVR on a given video dataset.	20
4.2	Diagram of the Retrieving procedure for the LlvRE CBVR system. A keyframe from one of the videos of the dataset is used as query input in the user interface and similar results from different videos are presented. Both user interface and query results are displayed scaled according to the user's device screen size.	22
4.3	Screen capture of the Retrieving web-interface for the LlvRE CBVR system as displayed on small screens allowing the selection of the image descriptor to use as well as some other settings parameters.	24
4.4	Sample image used as query input and screen capture of the first video match from this query, as shown on small screen size devices. The result shows info about the metadata from the video on the top and tiles on the bottom with info about the ranked results and time refinement where they are found.	25
5.1	Statistics of the Stanford I2V dataset. The light version of the dataset, a subset of the full version, is used for evaluating LlvRE.	27
5.2	Statistics of the Stanford I2V dataset. The light version of the dataset, a subset of the full version, is used for evaluating LlvRE.	27

5.3	The search process is based on two steps. First, <i>Scene Retrieval</i> : the system returns a ranked list of the most likely story clips to contain the query image. Second, <i>Temporal Refinement</i> : if the user is interested in a given clip, the system returns the specific segments/moments of time within the clip that contain the query image.	28
5.4	Scene Retrieval results for 10K candidates. Best results for mAP and mAP@1 are obtained with the PHOG descriptor with values 0.212 and 0.436, respectively.	31
5.5	Scene Retrieval results for 50K candidates. Best results for mAP and mAP@1 are obtained with the PHOG descriptor with values 0.221 and 0.448, respectively.	32
5.6	Scene Retrieval results for 100K candidates. Best results for mAP and mAP@1 are obtained with the PHOG descriptor with values 0.223 and 0.45, respectively.	32
5.7	Temporal Refinement results for 10K candidates. Best results for mJaccard are obtained with the JCD and Color Layout descriptors with values 0.091 and 0.09, respectively.	34
5.8	Temporal Refinement results for 50K candidates. Best results for mJaccard are obtained with the JCD and Color Layout descriptors with values 0.091 and 0.09, respectively.	34
5.9	Temporal Refinement results for 100K candidates. Best results for mJaccard are obtained with the JCD and Color Layout descriptors with values 0.092 and 0.09, respectively	35
5.10	Sample image queries asked on the online form for the users to search for on the qualitative evaluation with the 4 different topics mentioned above.	36
5.11	Screenshots of the different movements from the user's test 1. . .	38

Chapter 1

Introduction

Many research efforts try to automatize data processing in several fields where this task is being performed manually, reducing the amount of time needed to do them as well as improving an user's daily life. The automation of these processes and the availability of innovative tools help users to perform tasks with greater efficiency, agility and flexibility.

In several professional fields where monitoring and storage of videos are involved, the fast growth of the need to store huge amounts of video in servers requires efficient, modern, scalable query-by-image and query-by-video search and indexing tools. To find a specific video or videos given one of its frames as well as the specific second in that video that this frame belongs to can be a tedious task if done manually, and depending on the amount of hours of video to search within, an unbearable thing to do. It is important to provide tools that are capable to assist users and professionals both in the management of the content as well as its retrieval, helping them do these huge, unbearable tasks within few seconds, as well as some other new uses such as finding similar videos or similar frames within different videos, video hyperlinking or extension to Computer Vision applications among others.

Although there are already many studies about Content-Based Video Retrieval (CBVR) as well as tools available, two of the most recurrent challenges are the amount of videos to bare with and the consequent increment of time that it takes both to retrieve content as well as for indexing new one.

In our approach, we try to expand the Lucene Image Retrieval Engine (LIRE), an open-source Content-Based Image Retrieval (CBIR) library to work with video, provide tools to automate the tasks of indexing big amounts of content and provide an user interface to perform queries and visualizing the results.

1.1 Focus of the Thesis

This thesis focuses on an evaluation investigating whether extending Lire from a CBIR to a CBVR tool is useful and effective for users with different needs, as well as its performance with big amounts of video, using a quantitative and qualitative study.

The project requirements defined at the start were the following:

1. Understand the existing procedure to index images with the Lire software library, which is oriented to the indexing and retrieval of still images, and explore solutions to extend it to video.
2. Understand and setup Solr, an open-source enterprise search platform from Apache Lucene project, as well as LireSolr, the plugin designed to extend the functionality of Lire to it.
3. Design tools for the automation of the process of parsing and indexing a given video dataset.
4. Design a web interface to work with the previously parsed and indexed video dataset.
5. Perform a proof of concept with an available video dataset that is big enough and provides ground-truth for CBVR.
6. Design and run a quantitative study with the ground-truth provided by this dataset.
7. Perform a qualitative study on the video domain with both expert (video monitoring professionals) and non-expert users to assess the human-computer interaction for the indexing, search and browsing of videos.

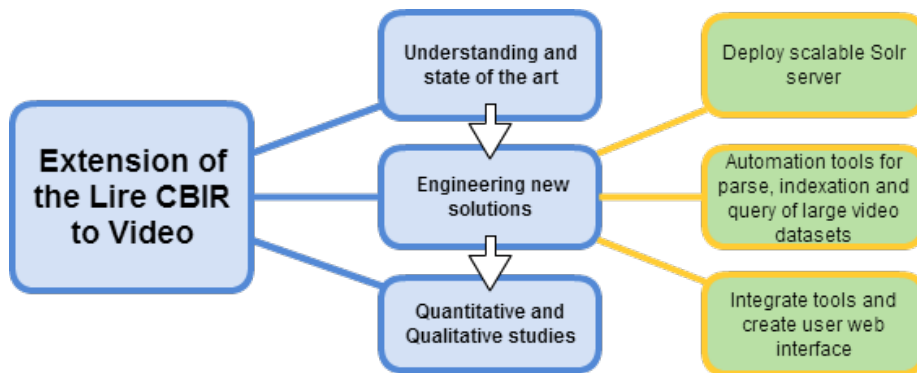


Figure 1.1: Original project planning stages.

This project was carried out at the University of Klagenfurt, in Austria, in the framework of the European Union Erasmus program for student mobility.

This scientific project investigates how users and professionals can be supported by information systems and innovative multimedia services. This thesis has been possible thanks to the support and co-operation of the Information Technology department of the Klagenfurt University in Austria. The main concept for this project were provided by my supervisor Professor Mathias Lux, who also provided software and documentation for the necessary work. The work was co-advised with Professor Xavier Giro-i-Nieto from the ETSETB school at the Universitat Politecnica de Catalunya. A monthly videocall together with Professor Lux allow a remote supervision of the work.

1.2 Motivation

Visual search is the problem of indexing and querying a large collection of visual data. There exist several variations of this problem, depending on the type of content in the database and the type of query. For example, image-to-image (I2I) visual search can be used for product search using an image taken with a mobile device. Video-to-video (V2V) is commonly used for copyright enforcement in online video-sharing websites. Video-to-image (V2I) is useful for augmenting the world seen by a head-mounted camera. Yet another flavor is image-to-video (I2V) visual search, where an image-based query is issued to retrieve relevant videos.

Applications such as advertisement monitoring, video lecture search using slides, organizing and searching a personal video collection or an archive of video or content linking where a relevant video has to be found based on an image from a certain event (e.g., from a website or news article) can be greatly simplified with a tool for I2V that can handle large amounts of data and perform searches within reasonable time.

Currently, there are few publicly available tools providing Image to Video (I2V) retrieval. The extension of an already available, efficient, open-source Image to Image (I2I) retrieval tool such as Lire to be used with video as an I2V tool could solve some problems of users and professionals related with the growth of the large amount of data stored day by day, not only for the challenge of finding scenes and moments that a specific frame belongs to, but might also open doors for brand new uses and possibilities that would suppose a gigantic effort otherwise.

In order to re-find easily these shot and summarize better the video content, a web user interface is developed, mainly, for covering the users' need to retrieve parts of the videos on demand.

This application allows on-line video retrieval based on still image features. The image-based retrieval is based on parsing a query image with a given image descriptor (such as Color layout, Edge Histogram, Joint Composite Descriptor, Pyramid Histograms of Oriented Gradients, etc.). The result is then given used as input for performing the query on a previously indexed dataset with the goal

to obtain the scene and the moment that the query belongs to. The results with the matching frames are presented on the user interface so that the matched frames are displayed, sorted according to its relevance and users' settings. One of the central question of this thesis is assessing whether this application is actually useful for professionals.

1.3 Outline of the Thesis

The rest of the thesis is structured as follows: Chapter 2 describes the state of the art, focusing on similar I2V / CBVR systems. Chapter 3 presents the system requirements that had to be satisfied by this project. Chapter 4 describes the software core adopted in this project and the different methods and tools developed for parsing, indexing and querying on a given video dataset. Chapter 5 describes the interface for our tool and the evaluation through a quantitative and a qualitative test. Chapter 6 presents the final conclusions of the study.

1.4 Work Plan of the Thesis

The Gantt chart shown on the Figures 1.2 and 1.3 illustrates the activities of the Thesis conducted over the exchange program in Klagenfurt. The length of each activity is shown on the horizontal axis throughout the entire semester.

The activities were divided in three larger blocks. First, I got acquainted with the environment of the Thesis and the side-tools needed to integrate in the project. In order to get used to the Lire software and the Solr search server, as well as other side-tools. Secondly, I started deploying a small dataset with few videos and engineering a partially automated system to extract, store, parse and update keyframes on a running Solr search server with the LireSolr plugin set-up. This step, together with developing an initial web interface where the user could perform queries and obtain results were completed in May. The last part of the activity was to set-up the system with a real-case scenario, parsing and indexing the Stanford I2V Dataset, a huge video dataset with ground-truth test-cases available to evaluate the results in a quantitative and qualitative way, which were done between June and July

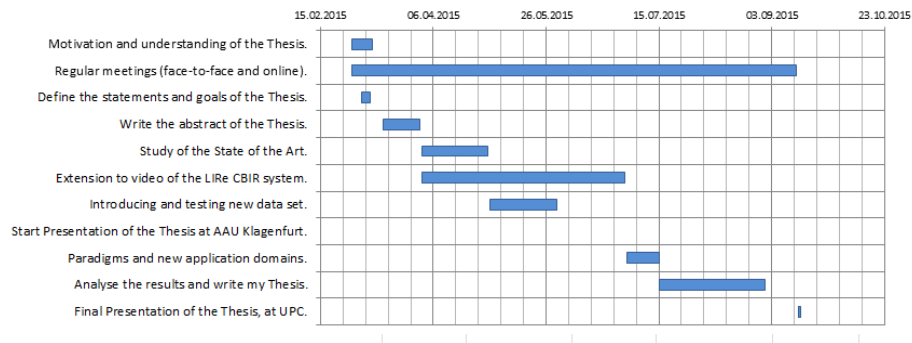


Figure 1.2: Gantt chart. The chart shows the calendar and organization of the Thesis.

Task	Start date	Duration (days)	Deadline
Motivation and understanding of the Thesis.	01.03.2015	9	10.03.2015
Regular meetings (face-to-face and online).	01.03.2015	197	14.09.2015
Define the statements and goals of the Thesis.	05.03.2015	4	09.03.2015
Write the <i>abstract</i> of the Thesis.	15.03.2015	16	31.03.2015
Study of the <i>State of the Art</i> .	01.04.2015	29	30.04.2015
Extension to video of the LIRe CBIR system.	01.04.2015	90	30.06.2015
Introducing and testing new data set.	01.05.2015	30	31.05.2015
Start Presentation of the Thesis at AAU Klagenfurt.			
Paradigms and new application domains.	01.07.2015	14	15.07.2015
Analyse the results and write my Thesis.	15.07.2015	47	31.08.2015
Final Presentation of the Thesis, at UPC.	15.09.2015	1	16.09.2015

Figure 1.3: Proposed schedule from the Gantt chart. The grid shows the expected start date for each task, as well as a guideline for the amount of days and deadlines.

Chapter 2

Related Work

In this Chapter we describe the state of the art for Content-Based Video Retrieval (CBVR) as well as other multimedia retrieval systems for different applications.

The aim of content-based retrieval systems must be to provide maximum support in bridging the semantic gap between the simplicity of available visual features and the richness of the user semantics. [13]. Content-based video retrieval is the most challenging and important problem when it comes to content retrieval. It can help users to retrieve desired video from a large video database efficiently based on the video contents through user interactions. A video retrieval system can be roughly divided into three major components: a module for extracting representative features from video frames, a module for indexing the extracted features for its future retrieval and one defining an appropriate similarity model to find similar video frames from the database. [20].

Most important related conferences and workshops where video and multimedia retrieval challenges take place are TRECVID ¹, SIGIR ², ACM ICMR ³ and ImageCLEF ⁴.

¹<http://trecvid.nist.gov/>

²<http://sigir.org/>

³<http://press.liacs.nl/icmr/>

⁴<http://www.imageclef.org>

2.1 Related Content-Based Video Retrieval (CBVR) systems

In contrast to most works on Content-Based Image Retrieval (CBIR) systems, CBVR systems addresses the problem of video retrieval instead of still images.

CBVR systems benefit from an extensive literature and previously done work on CBIR and introduces new approaches. While CBIR has been addressed from low-level features to high level semantics and concept detectors [32], as well as other ways to exploit visual features such as generating automatic text descriptors with computer vision algorithms [14] to use these labels as a support for text-based queries, on CBVR systems these CBIR techniques can be used and combined with new ones such as retrieval based on soundtracks and semantic analysis of audio [8] or audio patterns [16], as well as retrieval based on automatic concept detectors [26] [24].

A similar case to ours is presented in the [1]. The full dataset contains of 3,800 hours of newscasts and features 200 queries for retrieval evaluation providing a ground truth. Our system, LlvRE, relies on the light version of the previous dataset, with 1.035 hours of video and 78 queries for retrieval evaluation with provided ground truth. The challenge is to find specific images on the video sources, an approach referred by the authors as image-to-video, I2V. Moreover, a system operating on the dataset in [3] is presented by the authors (Figure 2.1).

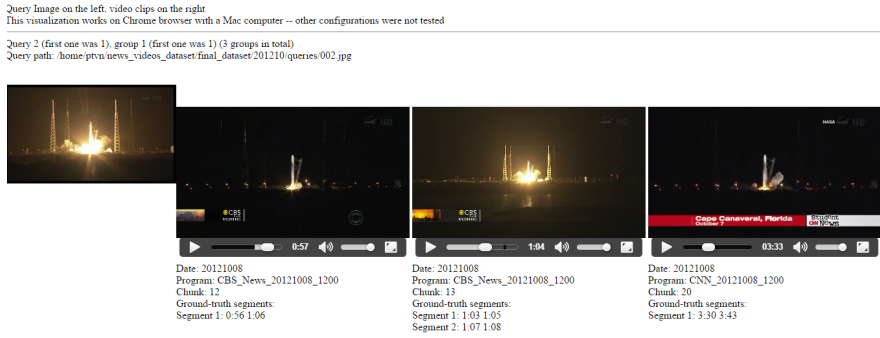


Figure 2.1: Sample visualization of the Stanford I2V ground-truth. Query image on the left is retrieved in three different videos at their specific time.

In the Jiku Director[21] (Figure 2.2), a system which automatically creates an event summary based on different videos from different users and view points is presented by the authors. The system relies solely on metadata. It is a Web-based application that the main contribution is the creation of the summary from event videos uploaded by users. The application uses an algorithm that considers view quality, video quality such blockiness or illumination, and spatial-temporal diversity in order to create this summary. The system, in contrast to

ours, is not focused on the retrieval of scenes. Following this system, in [27] the authors present a system focusing on videos taken at events, employing a controlled and holistic approach. Videos recorded with their software are automatically enriched with metadata, i.e. sensor readings, allowing for faster and easier retrieval.

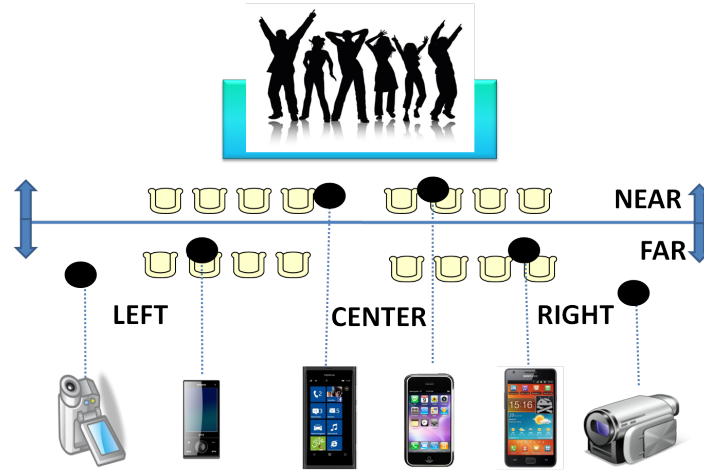


Figure 2.2: Diagram of the Jiku Director system for a scenario where different people are recording the same at the same time but with different devices and in different positions.

For medical videos, [9] proposes a framework that uses principal video shots for video content representation and feature extraction. The classification is implemented mainly by elementary semantic medical concepts, such as “Traumatic surgery” or “Diagnosis”. Moreover, [22] presents a framework to retrieve short videos in real time by modeling the motion content with a polynomial model. The system has been successfully applied to automated recognition of retinal surgery steps on a 69-video data set.

Also, for medical videos, [23] proposes a video retrieval system for endoscopic procedures (Figure 2.3). In the system, video frames taken during surgery allows for video retrieval based on visual features and late fusion for surgeons to re-find shots taken during the procedure.

A rich internet application for video retrieval from a multimedia asset management system, Digimatge, is proposed in [11] (Figure 2.4). It describes the integration of two new services aimed at assisting into the retrieval of video content from an existing Multimedia Asset Manager (MAM) of a TV broadcaster archive. The first tool suggests tags after a first textual query, and the second ranks the keyframe of retrieved assets according to their visual similarity.

Related with video retrieval from broadcaster archives, where video retrieval through text queries is a common practice, [10] presents a system (Figure 2.5) where the query keywords are compared to the metadata labels that documen-

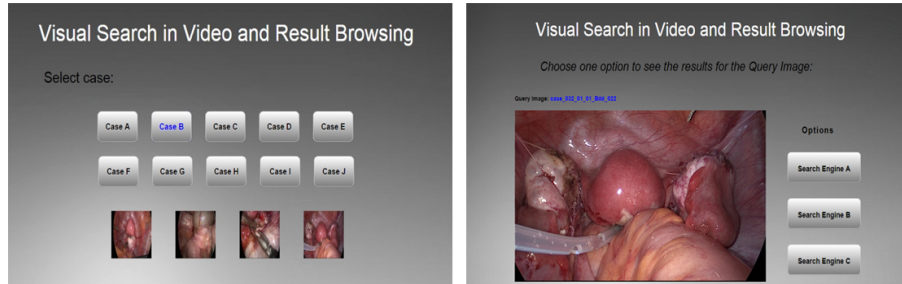


Figure 2.3: Main and second window for the medical semi-interactive interface for endoscopic video retrieval

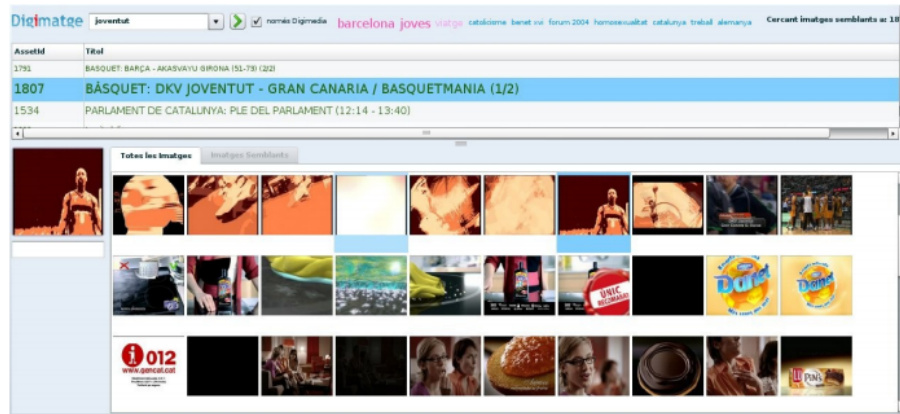


Figure 2.4: Digimatge user interface with visual and keyword queries.

talists have previously associated to the video assets, focusing on a ranking strategy to obtain more relevant keyframes among the top hits of the results ranked lists but, at the same time, keeping a diversity of video assets.

For video keyframe retrieval, a hierarchical navigation and visual search engine referred as GOS (Grafical Object Search, Figure 2.6) is proposed in [28]. This works presents a browser that supports two strategies for video browsing: the navigation through visual hierarchies and the retrieval of similar images. In this system, keyframes are extracted from videos and hierarchically clustered with the Hierarchical Cellular Tree (HCT) algorithm, an indexing technique that also allows the creation of data structures suitable for browsing. The navigation can directly drive the user to find the video timestamps that best match the query or to a keyframe which is globally similar in visual terms to the query.

On the comercial field, we find MediaMill, a semantic video search engine from the University of Amsterdam [30] (Figure 2.7). The basis for the engine is a semantic indexing process which is currently based on a lexicon of 491 concept detectors. To support the user in navigating the collection, the system defines a visual similarity space, a semantic similarity space, a semantic thread space, and browsers to explore them.

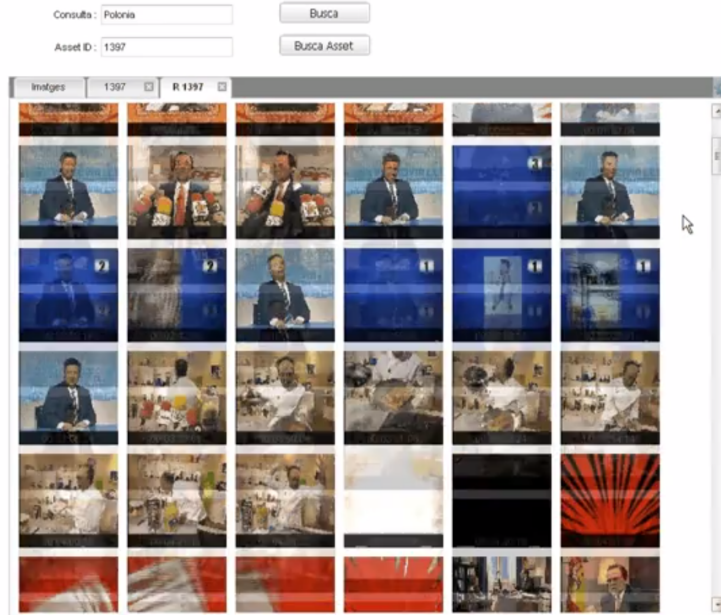


Figure 2.5: Video retrieval interface for the broadcaster archive.

On the open-source field, we find VIREO-VH (Video Hyperlinking) [25], an open source software providing end-to-end support for threading and visualizing large video collections. This software includes components for near-duplicate keyframe retrieval, partial near-duplicate video alignment, and Galaxy visualization of the video collection. A sample galaxy of visual snippets (or technically clusters of graphs) can be visualized on Figure 2.8.

Also on the the open-source field, we find a similar content-based image retrieval tool to LIRE, [17] the library that we will be using on this project. This library is the ImageTerrier [12], a scalable, high-performance search engine platform for content-based image retrieval applications. The ImageTerrier platform provides a test-bed for experimenting with image retrieval techniques. The platform incorporates an implementation of the single-pass indexing technique for constructing inverted indexes and is capable of producing highly compressed index data structures. ImageTerrier is written as an extension to the open-source Terrier test-bed platform for textual information retrieval research.

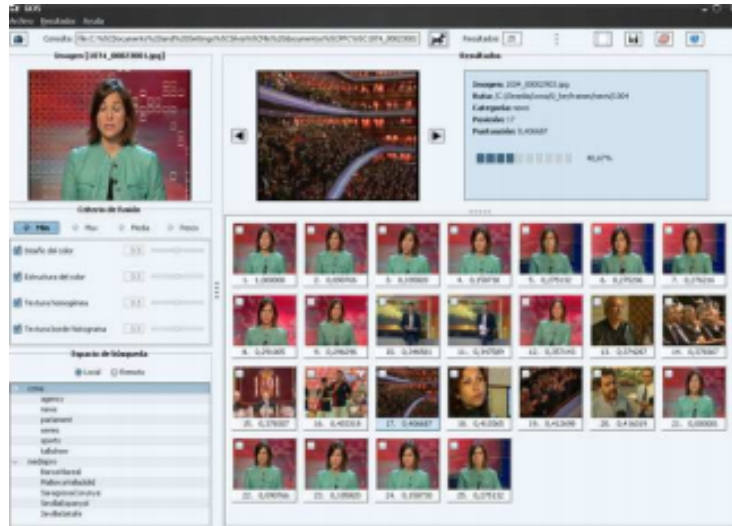
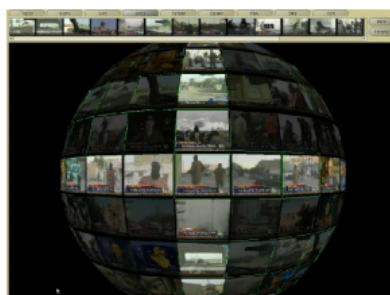


Figure 2.6: GOS (Graphical Object Searcher) user interface.



Cross Browser



Galaxy Browser

Figure 2.7: MediaMill semantic video search engine browsers.

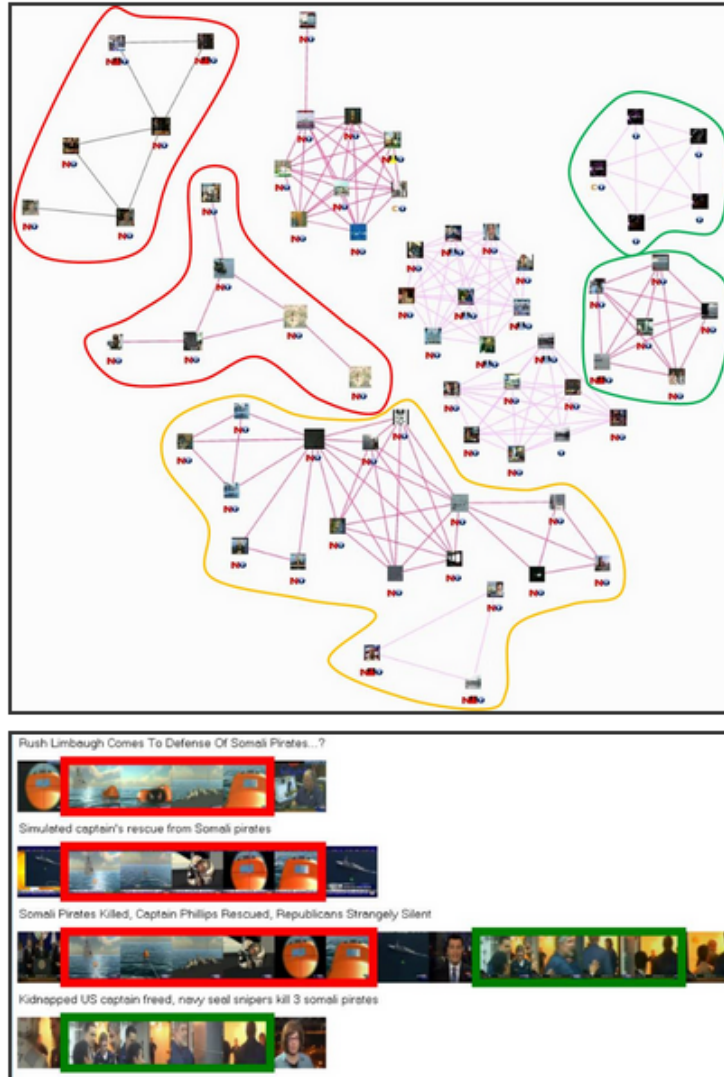


Figure 2.8: VIREO-VH Screenshots. Above, galaxy of visual snippets (or technically clusters of graphs). There are several visual snippet structures in this galaxy view. The structure carries semantic meanings. A fully connected snippet (highlighted with green) indicates highly redundant videos, a sparse structure (highlighted with red) implies an evolving event and a highly centralized structure (hub) with excessive hyperlinks to other videos. Below: zooming in a visual snippet, each video is represented as a series of keyframe thumbnails.

Chapter 3

Requirements

The global requirements for the LIVRE CBVR system is to provide a series of tools for, given a video dataset with thousands of hours of videos, to be able to parse and index the whole dataset as well as a user interface to retrieve videos from it. The index should be scalable so that new videos can be added to an existing index and the retrieval user interface should be web-based and accessible from different devices. Queries should handle images as input. The system should present results on the web interface showing one or more video results, the matching frames on those videos and the specific time refinement on those videos where the matched result frames are located.

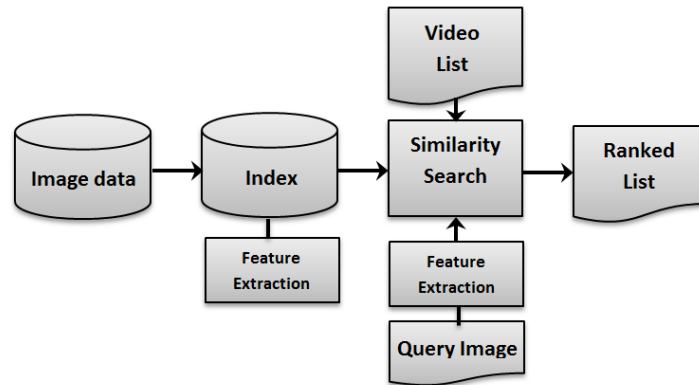


Figure 3.1: Diagram with the global requirements for a CBVR system on a previously indexed dataset. A specific implementation for the LIVRE CBVR, structured with blocks for parsing, indexing and retrieving is presented on Chapter 4.

As already stated in the previous chapter, video retrieval systems can serve for different purposes; they can be developed for specific contents as well as for different user profiles. In this chapter we analyze the requirements from the perspective of the user.

The goal of our project is to design, deploy and evaluate the performance of the extension to video of the LIRE software library [18], referred as L_{iv}RE system in this project. For this goal to be achieved, some side-tools as well as a web-based interface to work with a large-scale video dataset should be provided. With the LIRE library we can integrate up to 20 different visual features which can be extended to work with video. This tool is able to generate automatically a ranked list of videos according to the similarity of their frames to a query image selected by the user, and provides as well settings to refine and extend results. In this project we focus solely on visual information, and we report on experiments based on a research dataset.

The main difference between our system and the related work in Chapter 2 should be the adoption of Solr¹ as the backend for indexing and retrieval on a given dataset. Solr is an open source enterprise search platform, written in Java, from the Apache Lucene project, and should be deployed and adapted to work together with the LIRE CBIR library to become the core of the CBVR engine. We want the L_{iv}RE CBVR system to be fast both at indexing and retrieval so that it could be used not only with academic purposes. More details about Solr, its major features, keypoints and its implementation in this project can be found in Chapter 4.

We will then be handling the extracted features from the provided video dataset and processing it in such a way that we can efficiently index and retrieve them afterwards with the Solr search platform and our web-based used interface.

3.1 Content requirements

We can structure the content requirements of the L_{iv}RE CBVR system in 3 main blocks: Parsing, Indexing and Retrieval.

3.1.1 Parsing requirements

Given a video dataset in a given folder structure, the user must be given a tool to perform the following actions automatically:

- Generate a list with all the supported video files inside that structure.
- Extract the keyframes from those videos and store them on separate folders.
- Parse all the extracted keyframes with the selected image descriptors (Color Layout, Edge Histogram, JCD and PHOG) and present the results

¹<http://lucene.apache.org/solr/>

in an XML document supported by Solr. More info about the selected image descriptors can be found at Chapter 4

3.1.2 Indexing requirements

Given a running and set-up deployment of the Apache Solr search engine with the LireSolr plugin installed and configured, as well and the XML documents obtained during the parsing stage containing the image features of the keyframes, the user must be given a tool to perform the following actions automatically:

- Generate a list with all the XML documents in the folder structure.
- Upload those documents (index them) in the Solr core.
- Commit changes in the core.

3.1.3 Retrieval requirements

Given a previously indexed video dataset on a deployment of the Apache Solr search engine, with the LireSolr plugin installed and configured, the user must be given a web-based interface to perform the following actions:

- Search field to introduce any image to search on the video dataset.
- Settings to interact with the Solr Core engine and refine search results, such as number of candidates, accuracy, image descriptor to use, etc.

The results should be presented on the web-based interface to the user with the following requirements:

- Candidate videos to contain the searched image from the dataset displayed using HTML5.
- Thumbnails of the similar frames on the candidate video.
- Information about the specific moment of time (hour, minute, second) where the similar image is present on the candidate video.
- Other similar images to the searched one. These ones should be usable to perform a new searches.
- Other information and metadata about the retrieved video results, as well as the time needed to perform the search.

In addition, the web-based interface should independent from the device, OS, and web browser. It should as well be scalable and modular to be usable from any screen size.

3.2 Evaluation requirements

The approach should be validated by verifying whether its results fulfill the original requirements in an objective, quantitative way as well in a more subjective, qualitative way.

3.2.1 Quantitative evaluation requirements

A suitable dataset with ground-truth available should be used to evaluate objectively the query results obtained from the LlvRE CBVR system. These tests should be performed automatically and should present results on a standard format with the goal of comparing it to other existing systems.

The elements from the LlvRE CBVR system to evaluate are:

- Scene retrieval performance - Evaluates if the searched frame is present or not in the video results.
- Time refinement performance - Evaluates if the searched frame is present not only on the correct video, but as well in the correct times in the video.

3.2.2 Qualitative evaluation requirements

Our system must be evaluated by participants subjectively through an interactive web-interface survey, based on some videos and queries selected from the whole data set. In addition, some of these surveys will be performed using *thinking aloud tests*, with the goal of recording their movements, voice and opinions while using the web-based user interface with the goal of being able to evaluate the results of our final visual research.

Chapter 4

Developed solution

This chapter presents a deeper overview of the implementation and set up that have been assessed in this thesis to cope with the requirements stated on the Chapter 3 for each one of the 3 blocks from the requirements: Parsing, Indexing and Retrieval. The evaluation of the developed solution, as well as more information about the video dataset selected to perform these evaluations, the Stanford I2V Dataset [2], will be later introduced in Chapter 5.

The global descriptors selected for this project are presented in Section 4.1. A brief introduction about Solr and why is it the search engine chosen as the core for the LIRE CBVR system can be found at Section 4.2.

4.1 Selected global descriptors

Our solution does not focus on developing or implementing any specific image descriptor, but it is rather a platform where existing and new ones can be added for any specific or academic purpose. As long as a the image descriptor can be treated as text, so that the Apache Lucene Core ¹ running inside Solr ² can handle it properly, these descriptors can be based on features of any kind.

The version of LIRE [17] used in this work implements up to 20 different visual descriptors for visual indexing. LIRE presents a modular architecture which allows adding existing and new descriptors.

The four descriptors selected for this implementation and further evaluation are:

MPEG-7 Color Layout (CL) [15]: The Color Layout is an image color de-

¹<https://lucene.apache.org/core/>

²<https://lucene.apache.org/solr/>

scriptor that represents the spatial layout of color images in a very compact form. It is based on generating a tiny (8x8) thumbnail of an image, which is encoded via Discrete Cosine Transform (DCT) and quantified. As well as efficient visual matching, this also offers a quick way to visualize the appearance of an image, by reconstructing an approximation of the thumbnail inverting the DCT.

MPEG-7 Edge Histogram (EH) [29]: The Edge Histogram is an image texture descriptor that represents the spatial distribution of five types of edges (four directional edges and one non-directional). It consists of local histograms of these edge directions, which may optionally be aggregated into global or semi-global histograms.

Joint Composite Descriptor (JCD) [31]: The Joint Composite descriptor is a combination of two other descriptors: The Color and Edge Directivity Descriptor (CEDD) [6], a compact joint histogram of fuzzy color and texture, and the Fuzzy Color and Texture Histogram (FCTH) [7], a descriptor that combines in one histogram color and texture information. The structure of these descriptors consists of n texture areas. In particular, each texture area is separated into 24 sub regions, with each sub region describing a color. CEDD and FCTH use the same color information, as it results from 2 fuzzy systems that map the colors of the image in a 24-color custom palette. JCD is made up of 7 texture areas, with each area made up of 24 sub regions that correspond to color areas.

Pyramid Histogram of Oriented Gradients (PHOG) [5]: The Pyramid Histogram of Oriented Gradients descriptor represents an image by its local shape and the spatial layout of the shape. Local shape is captured by the distribution over edge orientations within a region, and spatial layout by tiling the image into regions at multiple resolutions. The descriptor consists of a histogram of orientation gradients over each image subregion at each resolution level – a Pyramid of Histograms of Orientation Gradients (PHOG). The distance between two PHOG image descriptors then reflects the extent to which the images contain similar shapes and correspond in their spatial layout.

4.2 Solr as the LlvRE CBVR system search engine

Solr ³ is an open source enterprise search platform, written in Java, from the Apache Lucene project. Its major features include full-text search, hit highlighting, faceted search, real-time indexing, dynamic clustering, database integration, NoSQL features and rich document handling, among others. As for today, Solr is the most popular enterprise search engine. ⁴

³<http://lucene.apache.org/solr/>

⁴<http://db-engines.com/en/ranking/search+engine>

LIRE uses the same core text search engine as Solr, Lucene ⁵, to maintain the index. Lucene allows for indexing of text and metadata in combination with the low-level features, has a small footprint, is easy to use and manage, and has proven to be fast enough for many different scenarios. An image file with all its features is typically reflected by a single Lucene Document. [17]

The integration of LIRE into Solr, then, benefits from the fact that both share the same core text search engine and both are written in Java. It also benefits from other main characteristics from Solr such as the distributed search and index replication and scalability, which allows to extend our dataset and performance requirements and add redundancy if needed, also allows easy migration into another system. Other factors that makes our system robust is that Solr provides fault tolerance.

Our CBVR system, LlvRE, relying then on Solr, runs as a standalone full-text search server. It uses the Lucene Java search library at its core for full-text indexing and search, and has REST-like HTTP/XML and JSON APIs that make it usable from most popular programming languages. The use of LireSolr plugin ⁶ allows LlvRE to be tailored to many types of applications and more advanced customization.

4.3 Parsing, Indexing and Retrieval

In this section we describe the developed solution for the requirements of the LlvRE CBVR system.

Given a video dataset, the first block, Parsing, performs the first step by taking this video dataset as input and outputs documents containing all the image features from the keyframes of each one of the videos. These image features are the 4 global descriptors mentioned in Section 4.1.

The second block, Indexing, will then take as input the documents generated from Parsing and integrate them into the Solr search engine making them ready to be retrieved.

The third and final block, Retrieval, is integrated with the web-based user interface to query the Solr engine and present the results to the user in a modular screen, independent from the user's browser, hardware or OS.

A diagram with the developed process from the input dataset to the user interface for retrieval is shown in the Figure 4.1.

⁵<http://lucene.apache.org/core/>

⁶<https://bitbucket.org/dermotte/liresolr>

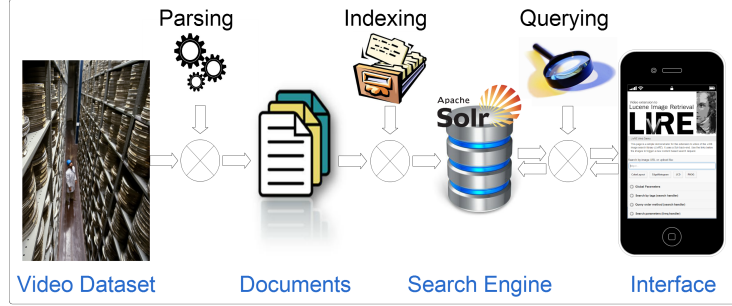


Figure 4.1: Diagram of the developed solution for the LIRE CBVR system with the 3 consecutive blocks: Parsing, Indexing and Retrieving (Querying). These independent blocks are implemented to allow the user to perform CBVR on a given video dataset.

4.3.1 Parsing procedure

Given a video dataset in a given folder structure, we provide the user a Python script, "extractParseDataset.py", that performs the actions below, sequentially, in either Unix or Windows OS system folder structures.

1. Generates a list called "videoFiles.txt" with all the supported video files inside that folder structure.
2. Extracts the keyframes from those videos at 1 frame/s rate ⁷ and store them on separate folders following the same folder structure as the dataset. For this step to be performed, the FFmpeg tool ⁸, a standalone multimedia framework able to decode, encode, transcode, mux, demux, stream, filter and play a wide variety of multimedia content is required to be either on the same folder as the Python script or added to the OS's path.
3. Parses all the extracted keyframes with each of the selected image descriptors (Color Layout, Edge Histogram, JCD and PHOG) and presents the results in an XML document with the format supported by Solr. For this task to be performed, the LIRE Request Handler [19], a Solr plug-in for large scale CBIR, should be compiled and available in the same folder as the Python script or added to the OS's path. Detailed instructions on how to generate the "lire-request-handler.jar" JAR file are given on <https://bitbucket.org/dermotte/liresolr>.

⁷The extracted keyframes from the video files are named as the second from the video from which they were extracted. This is convenient during the Retrieval procedure for an accurate temporal refinement, finding which second of a given video the retrieved frame belongs to.

⁸<https://www.ffmpeg.org/>

This step is by general means high resource consuming. Depending on the size of the video dataset to parse and the machine performing the task it can take from few minutes to several hours. To optimize this task, parallelization is added to the Python script. By default, 4 threads run in parallel when parsing a dataset. This setting can be modified inside the script to optimize the speed of the parsing task according to the available hardware resources.

4.3.2 Indexing procedure

Before moving on with the Indexing procedure, we need to set-up our Apache Solr instance ⁹ with the following instructions. In first place, we need to install the Lire Solr plugin ¹⁰. After downloading the source code from the repository, <https://bitbucket.org/dermotte/liresolr>, a single JAR should be generated using Apache Ant3 (ant task dist). Then, after adding this .jar to the classpath so that Solr can find it, the new request handler and the sort function have to be added to the "solrconfig.xml" configuration file. Finally, the fields for content based retrieval are added to the index schema along with the definition of the custom index field type storing the feature vectors.

Once the user is successfully installed this instance of the Apache Solr search engine with the Lire Solr plugin properly set-up, given the XML documents obtained during the Parsing procedure containing the image features of the keyframes, the user is given another Python tool, "uploadDataset.py", that performs the following actions automatically:

- Generate a list "XMLFiles.txt" with all the XML documents in the folder structure.
- Upload those documents containing the image descriptors to the Solr core. For this step to be performed, Curl ¹¹, an open source command line tool and library for transferring data with URL syntax supporting several protocols is required to be either on the same folder as the Python script or added to the OS's path.
- Commit changes in the core.

Once this step is done, our dataset is indexed, ready to be queried and browser.

⁹For this project, the version 4.10.2 of Apache Solr is used. For newer versions some slight modifications on the set-up procedure might be needed

¹⁰Lire as well as the Lire Solr plugin are licensed under GNU General Public License (GPL) v2. Both Lire Solr plugin and its detailed installation instructions are provided on <https://bitbucket.org/dermotte/liresolr>

¹¹<http://curl.haxx.se>

4.3.3 Retrieval procedure

Given a previously indexed video dataset on a deployment of the Apache Solr search engine, with the LireSolr plugin installed and configured, the user is given a web-based interface to perform queries over it and interact with the Lucene Core. This interface requires a browser that supports javascript and HTML5 with the video tag enabled.

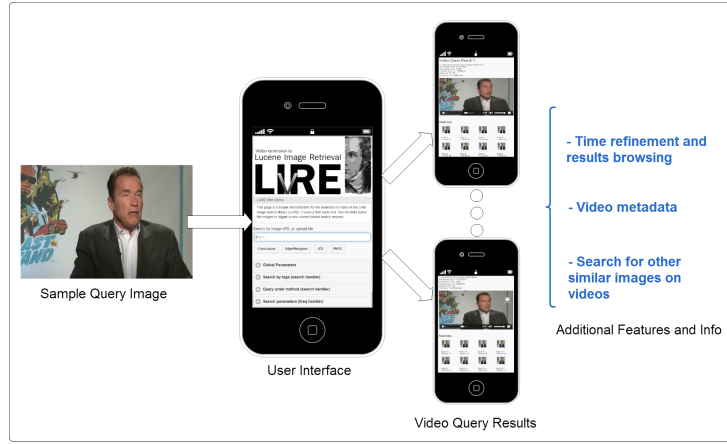


Figure 4.2: Diagram of the Retrieving procedure for the LlvRE CBVR system. A keyframe from one of the videos of the dataset is used as query input in the user interface and similar results from different videos are presented. Both user interface and query results are displayed scaled according to the user's device screen size.

As mentioned in the requirements, the web-based interface includes:

- A search field to introduce any image to search on the video dataset. Here the user can use as input any image URL, upload a local image from his device or select one of the keyframes from the dataset.
- Settings to interact with the Lucene Core engine and refine search results, such as number of candidates, accuracy, image descriptor to use and number of videos and similar images to display. The interface also allows to perform tag-based search although no annotation mechanism is yet implemented.

The results are be presented on the web-based interface to the user as follows:



- Candidate videos to contain the searched input image from the dataset are displayed using HTML5 video tag.
- Thumbnails of the similar frames on the candidate videos are displayed close to the video and link to the specific time where they are displayed.

- Information about the specific moment of time (hour, minute, second) where the similar image is present on the candidate video is displayed below the thumbnails.
- Other similar images to the searched one (optional). These are usable to perform a new searches.
- Information and metadata about the retrieved video results, as well as the time needed for a new query to be performed.

The web-based interface is independent from the device, OS, and web browser. In order for the interface to be scalable and modular so that it fits any screen size, Bootstrap ¹² , a popular HTML, CSS, and JS framework for developing responsive mobile projects was integrated on the web interface.

¹²<http://getbootstrap.com>

Video extension to
Lucene Image Retrieval



LivRE Web Demo

This page is a simple demonstrator for the extension to video of the LIRE image search library (LivRE). It uses a Solr back-end. Use the links below the images to trigger a new content based search request.

Search by image URL or upload file:

ColorLayout

EdgeHistogram

JCD

PHOG

+

Global Parameters

+

Search by tags (search handler)

+

Query order method (search handler)

+

Search parameters (lireq handler)

Figure 4.3: Screen capture of the Retrieving web-interface for the LivRE CBVR system as displayed on small screens allowing the selection of the image descriptor to use as well as some other settings parameters.



Figure 4.4: Sample image used as query input and screen capture of the first video match from this query, as shown on small screen size devices. The result shows info about the metadata from the video on the top and tiles on the bottom with info about the ranked results and time refinement where they are found.

Chapter 5

Evaluation

In this chapter we evaluate the LlvRE CBVR system in a quantitative and qualitative way. Both evaluations are performed using the Stanford I2V dataset [2]. An overview of the dataset is presented in Section 5.1.

For the quantitative evaluation, two tests are performed: *Scene Retrieval* and *Time Refinement* evaluations. Both are automated procedures based on the ground truth provided by Stanford I2V, which takes place automatically by using Python scripts specifically developed for this purpose.

For the qualitative evaluation, an interactive web-interface survey based on some videos and queries selected from the whole dataset is done by both expert and non-expert users. In addition, some of these surveys are performed using *thinking aloud tests*, with the goal of recording their movements, voice and opinions while using the web-based user interface.

5.1 The dataset

To evaluate LlvRE fulfilling the requirements stated on Chapter 3 we need a video dataset with thousands of hours of videos and ground-truth. The dataset chosen for evaluating this project is the Stanford I2V dataset, [2] which is a large dataset that fits our purpose of evaluating the task of retrieving videos using images as queries. Figure 5.2 provides statistics on the dataset composition. Although the full version of the dataset contains 3.8k hours of video, distributed across 84k video clips on average 2.7 minutes long, the dataset version that we use for our evaluation, the so called light dataset, is a subset of the full version with the intended use of faster experimentation. Each video clip on the light dataset corresponds to a single news story, collected from October 2012 until January 2013 from 39 different recurring newscasts in 25 different channels, segmented from a full-length newscast. These story clips are assembled from a

coherent collection of successive shots which cover a single event. Hence, each story clip usually contains tens of shots. These story clips, in the context of news videos, are the equivalent of scenes for general-purpose videos.

The dataset is accompanied by a set of queries with ground truth annotations. Image queries are collected from news websites, and they usually depict important events. Some sample query images from the ground truth and video frames from the dataset are shown on Figure 5.1 The light version of the dataset contains 79 queries. For each query image, a list of all database clips where it is found is provided along with a list of all precise segments it is shown in the clips.

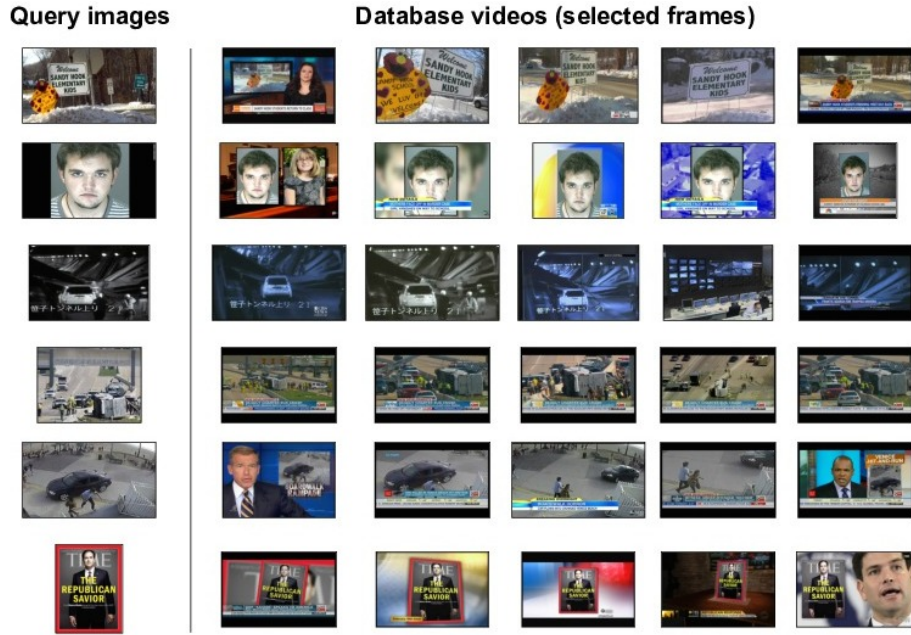


Figure 5.1: Statistics of the Stanford I2V dataset. The light version of the dataset, a subset of the full version, is used for evaluating LlvRE.

	# Video hours	# Queries	# Video clips	# Keyframes @ 1fps	Average clip duration (min.)
Full version	3,801	229	84,443	13,966,820	2.70
Light version	1,035	78	23,437	3,808,760	2.65

Figure 5.2: Statistics of the Stanford I2V dataset. The light version of the dataset, a subset of the full version, is used for evaluating LlvRE.

5.2 Quantitative study

In order to mimic a broad set of applications, we divide the experiments performed with the dataset in two stages, reflecting the two levels of annotation granularity, see also Figure 5.3. In the first stage, called *Scene Retrieval*, the objective is to return the correct story clips in the top of a ranked list. The second stage is *Temporal Refinement* where, given a story clip, the precise segments where the query image is visible have to be found.

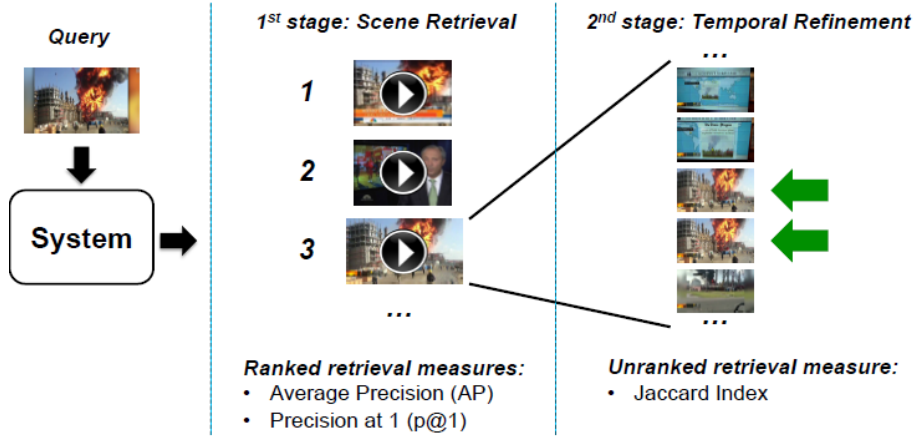


Figure 5.3: The search process is based on two steps. First, *Scene Retrieval*: the system returns a ranked list of the most likely story clips to contain the query image. Second, *Temporal Refinement* : if the user is interested in a given clip, the system returns the specific segments/moments of time within the clip that contain the query image.

5.2.1 Evaluation Metrics

In this subsection, the performance assessment protocol for the the dataset is presented.

5.2.1.1 Scene Retrieval metrics

The first step, *Scene Retrieval*, is considered a ranked retrieval type of problem, and we measure performance in this case using Average Precision (AP)

$$AP = \sum_{k=1}^n P(k) \Delta r(k)$$

where k is the rank in the sequence of retrieved documents, n is the number of retrieved documents, $P(k)$ is the precision at cut-off k in the list, and $\Delta r(k)$ is the change in recall from items $k-1$ to k . Although Average Precision assesses the quality of the returned ranked list of results and is useful in applications where a list of potential results is shown to the user, we also measure Precision at 1 ($p@1$), since it is important in cases where the best result is directly returned to the user (for example, in the case where the system would start playing the best clip match without further interaction with the user).

For results over a set of queries during *Scene Retrieval*, we also report the mean of the Average Precision scores for each query, called mean Average Precision (mAP)

$$mAP = \frac{\sum_{q=1}^Q AP(q)}{Q}$$

where Q is the number of queries. Similarly, mean Precision at 1 ($mp@1$) results are also presented.

5.2.1.2 Temporal Refinement metrics

In the second step, *Temporal Refinement*, is performed for the case when the user is interested in a particular clip. The system then indicates which points in time in the clip the query image was found. In practice, the LlvRE web-based interface presents these matches by showing thumbnails with ranking and time info nearby the video player. For this second stage, we have an unranked retrieval case, where a match should be presented to the user if the system is confident enough. Since the system may retrieve one or more segments within each ground truth clip, we assess performance in this case using the Jaccard index. In this case, the Jaccard index is computed by the ratio between the intersection of the retrieved and ground truth sequences and their union.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Similarly to the *Scene Retrieval* step, for results over a set of queries during *Temporal Refinement*, we report the mean of the Jaccard index scores for each query, called mean Jaccard index (mJac).

$$mJac = \frac{\sum_{q=1}^Q J(A, B)(q)}{Q}$$

where Q is the number of queries.

Note that, for Temporal Refinement evaluation, we consider that the correct story clip is given, so the system only needs to find the correct segments within the given clip. Also, in practice, we observed that the precise time segment annotations might vary by up to 1 second due to the usage of different video players. To avoid incorrect scoring, we introduce 1 extra second at the beginning and at the end times of the sequence. For example, if a ground truth sequence defined by our annotation process starts at 1:12 and ends at 1:23, we will score the retrieved sequence with respect to the time segment starting at 1:11 and ending at 1:24.

Implementations of scoring functions for both steps are provided on the Stanford I2V dataset website ¹, and the Python code to adapt the output provided by our system into the template input format required by these scoring function are provided together with this project.

It is important to mention that no objective evaluation metric for timing or latency are performed, not only because it depends on the configuration of the system used, but because the Solr core has an inner cache for queries. Since same images are used many times as input with different settings, the timing results obtained are not representative to a real-case scenario. Therefore, indicative values of retrieval timing are only evaluated qualitatively on Section 5.2.3 using the values provided to users in the LlvRE web-based interface.

5.2.2 Results Evaluation

In this subsection we present and comment the experimental results obtained applying the metrics in 5.2.1. For this task, in both *Scene Retrieval* and *Temporal Refinement* we use the 4 image descriptors specified in 4.1 and we perform the test over the ground truth in function of accuracy ² values for different number of candidates ³.

¹<http://blackhole1.stanford.edu/vidsearch/dataset/stanfordi2v.html>

²Accuracy is settings parameter oriented to *better than linear* runtime complexity. It's a trade-off approach between runtime complexity and precision of results. An accuracy parameter below 1 means that the results are approximate but the search is performed faster.

³The number of candidates is another settings parameter oriented to reduce runtime complexity. Lower values means faster searches but less accurate results.

5.2.2.1 Scene Retrieval results

During this stage, first we obtain a ranked list of keyframes according to the query's most similar results in the Solr core using each one of the 4 image descriptors. From this list, we obtain the top 100 ranked scenes, where a scene score is defined as the best score among all of its constituent keyframes. For each query, we compute AP and p@1 based on the top 100 retrieved scenes. Results are presented in Figures 5.4, 5.5 and 5.6 below:

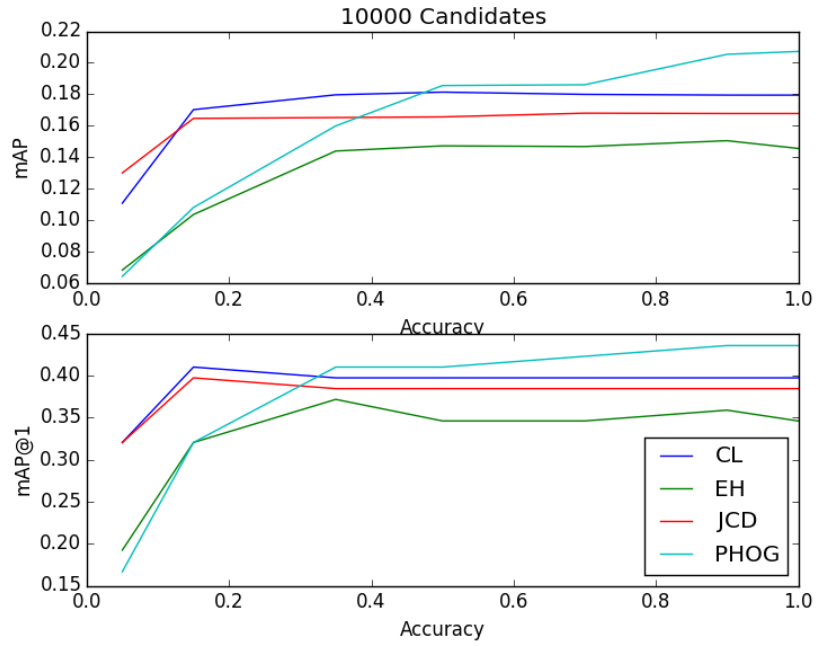


Figure 5.4: Scene Retrieval results for 10K candidates. Best results for mAP and mAP@1 are obtained with the PHOG descriptor with values 0.212 and 0.436, respectively.

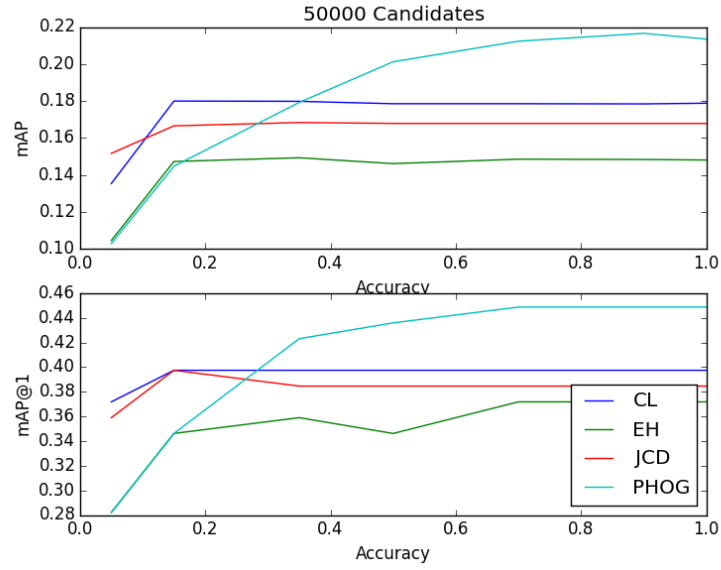


Figure 5.5: Scene Retrieval results for 50K candidates. Best results for mAP and mAP@1 are obtained with the PHOG descriptor with values 0.221 and 0.448, respectively.

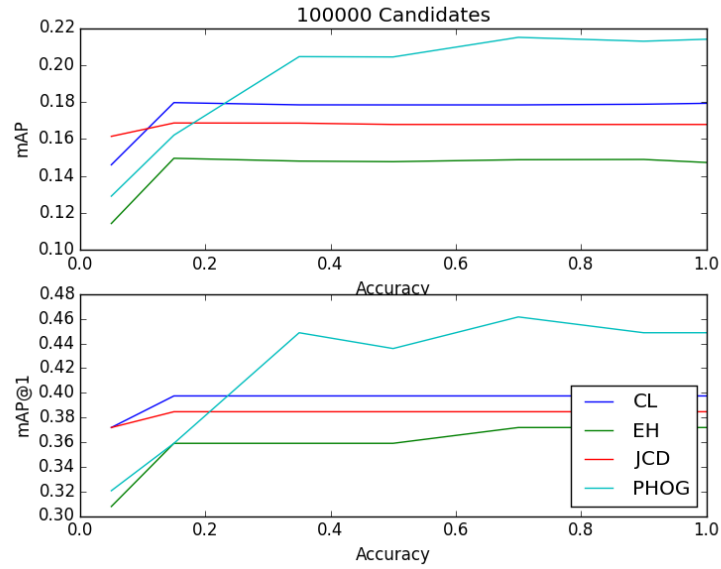


Figure 5.6: Scene Retrieval results for 100K candidates. Best results for mAP and mAP@1 are obtained with the PHOG descriptor with values 0.223 and 0.45, respectively.

Table 5.1: mAP results for 100K candidates.

Descriptor	0.35 accuracy	1 accuracy
Color Layout	0.183	0.183
Edge Histogram	0.154	0.154
JCD	0.174	0.174
PHOG	0.213	0.223

Table 5.2: mAP@1 results for 100K candidates.

Descriptor	0.35 accuracy	1 accuracy
Color Layout	0.397	0.397
Edge Histogram	0.359	0.372
JCD	0.384	0.384
PHOG	0.448	0.45

From the results above we observe that the best values are obtained with the PHOG descriptor, and that the average precision increment decelerates greatly above a parameter of accuracy of 0.35. For a standard set-up, when a balance between retrieval speed and precision is required this is an optimal parameter.

We also find it important to mention that the descriptors here used are mostly oriented for global features, and the ground truth provided for the tests are oriented for both global features and local features. In other words, we could expect with this specific ground truth test set much better results using a local descriptor or some combination between local and global descriptors.

5.2.2.2 Temporal Refinement results

To evaluate the Temporal Refinement stage, we consider each ground-truth clip for each query separately. For each ground truth clip, we find the 50 most similar frames by querying once again the Solr core with the 4 image descriptors. For each query, we compute the mean Jaccard index. Results are presented in Figures 5.7, 5.8 and 5.9 below, where the total mean Jaccard index for all queries is computed:

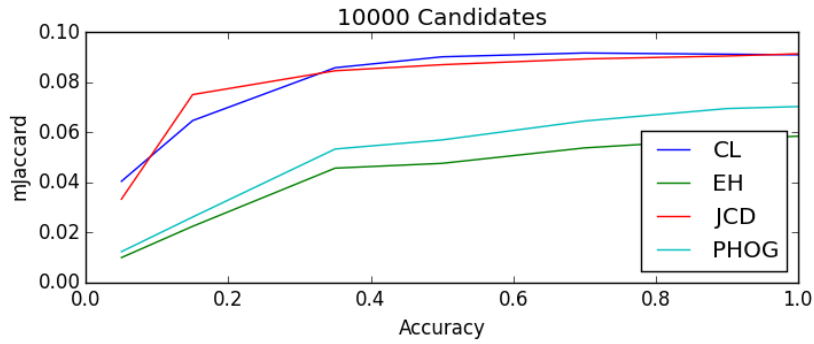


Figure 5.7: Temporal Refinement results for 10K candidates. Best results for mJaccard are obtained with the JCD and Color Layout descriptors with values 0.091 and 0.09, respectively.

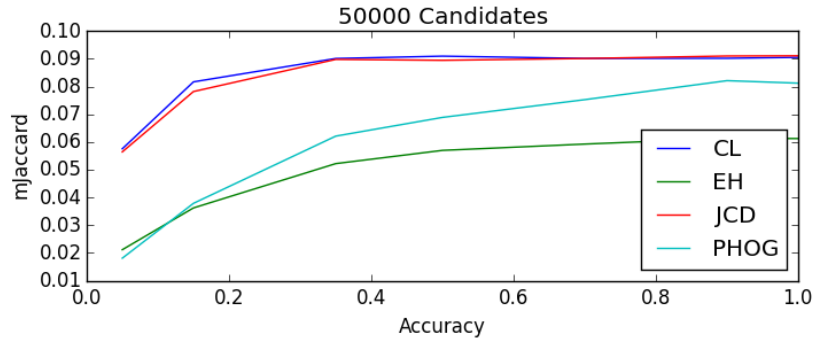


Figure 5.8: Temporal Refinement results for 50K candidates. Best results for mJaccard are obtained with the JCD and Color Layout descriptors with values 0.091 and 0.09, respectively.

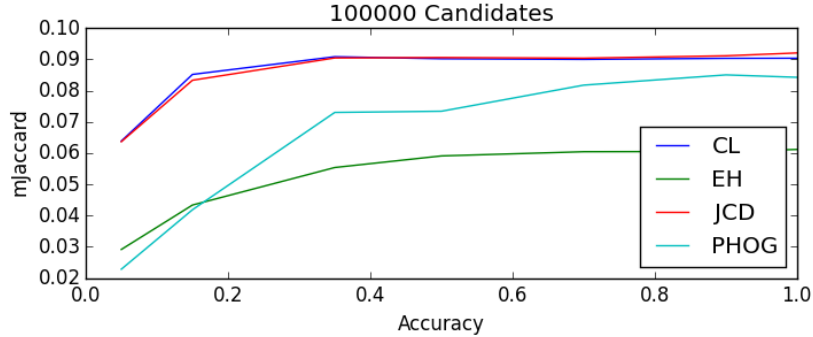


Figure 5.9: Temporal Refinement results for 100K candidates. Best results for mJaccard are obtained with the JCD and Color Layout descriptors with values 0.092 and 0.09, respectively

Table 5.3: mJaccard results for 100K candidates.

Descriptor	0.35 accuracy	1 accuracy
Color Layout	0.0901	0.903
Edge Histogram	0.0553	0.0611
JCD	0.0904	0.092
PHOG	0.0733	0.0842

From the results above we observe that the best values are obtained with the JCD and Color Layout descriptors, and that the mean Jaccard index increment decelerates greatly above a parameter of accuracy of 0.35, similar to the Scene Retrieval case. For a standard set-up, when a balance between retrieval speed and precision is required this is an optimal parameter.

Once more, we find important to mention that the descriptors here used are mostly oriented for global features, and the ground truth provided for the tests are oriented for both global features and local features. If the ground truth queries oriented to local features are removed from the test cases, the mean Jaccard index results raise greatly above 0.5. In order not to introduce any bias to other scenarios, only the original results with all the queries are presented here. We expect much better mean Jaccard index results when using local features descriptors or a combination of local features with global ones.

5.2.3 Qualitative user study

5.2.3.1 Evaluation Method and Procedure

The qualitative study takes in consideration the usual tasks that are expected to be performed on the LlvRE web-based interface. For this study, 6 participants, 4 non-expert and 2 expert users were asked to perform different query tasks and report. In each event, the user has to perform several query topics and images and report some metric values as well as their satisfaction, personal opinions and suggestions.

In order to do the study, we prepared a more user-friendly configuration of the web-based interface, leaving only the PHOG descriptor and eliminating some advanced settings from the interface in order to facilitate its use for the non expert user. We set the accuracy parameter to 1 for 3 users and 0.35 for the other 3 users, both with 50K candidates.

We then run a local Apache web-server on the same machine where the LlvRE system was running and made it available on-line so that remote users could access it and perform the tests remotely.

We prepared an on-line form to fill with 4 different image queries from 4 different topics: a building, a graphic, a portrait and frame with a famous actor. A sample of these image query topics are shown on Figure 5.10. We also prepared a free topic case where the user can search for an image of his own of any popular topic of his choice: (i.e. Famous actors, politicians, sport events, animals...) and use it as input on the search engine.



Figure 5.10: Sample image queries asked on the online form for the users to search for on the qualitative evaluation with the 4 different topics mentioned above.

To avoid random answers from the users when performing the qualitative evaluation, we queried the images in advance to know from which newscast channel they belonged and then asked them to answer this question from a multiple-choice selection menu. If a given user would answer this simple question wrong, his results would not be considered for evaluation.

As mentioned in 5.2.1, here we provide some qualitative timing information about the query time and the ranking time. These metrics are added to the user web-interface and asked for the participants to provide at each query.

With two of the non experienced users we performed a *thinking aloud* test, as described in [4]. Thinking-aloud protocols involve participants thinking aloud as they are performing a set of specified tasks. Users are asked to say whatever they are looking at, thinking, doing, and feeling as they go about their task. This enables the observers to see first-hand the process of task completion. The observers objectively take notes of everything that user's say, without attempting to interpret their actions and words. Test sessions are audio- and video-recorded so that we can go back and refer to what participants did and how they reacted. The purpose of this method is to make explicit what is implicitly present in subjects who are able to perform a specific task.

The evaluation session consisted of two parts; part one being a hands-on experience by the participants while using the web-based interface with the queries described before. Part two is an open interview reflecting his experience with the tool; we ask the volunteers during the interview what they think about the tool and which conclusions they extract from this test.

5.2.3.2 Participants

The tests were run with 4 students and 2 professors. Only the professors were experienced users with previous use of similar applications.

For the hands-on experience, we indicated the methodology of the participants the *thinking aloud test* while they were provided of a Personal Computer running our web-based interface on a standalone web server. The sessions were recorded with one video camera over their shoulders capturing his mechanical interaction with the tool. The camera captures the whole scene of the screen including the voice (cp. Figure 5.11).

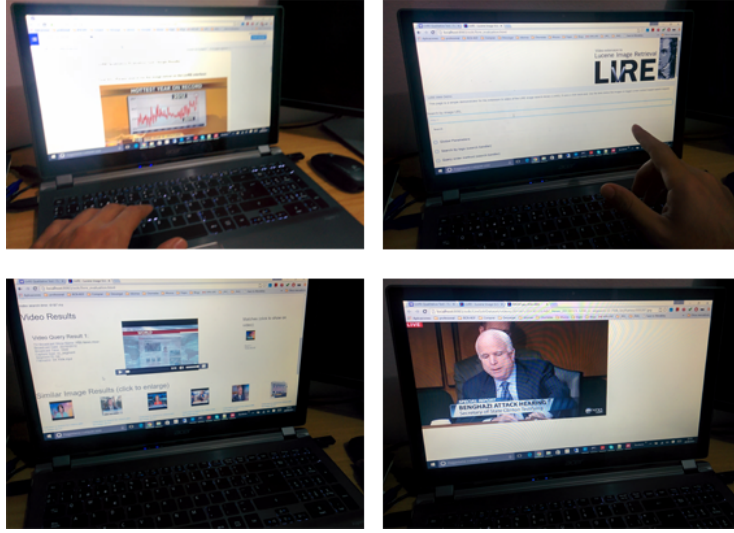


Figure 5.11: Screenshots of the different movements from the user's test 1.

5.2.3.3 Results

Here we present the questions asked to the users on the form and the most relevant answers provided by them.

Firstly, we asked the users about how long did the search take and to introduce the time provided on the web interface. We found this very relevant question because it gives us an indicative value of the time the system needs to perform a query. On the Table 5.4 we present the result obtained in milliseconds. It's important to mention that this test was performed on a Personal Computer with limited shared resources. Although Solr index is loaded on the RAM memory, videos and images are placed on an external HDD and this can make fetching images and videos quite slow compared to a dedicated server/system with more generous resources.

Table 5.4: Timing results for 50K candidates (in milliseconds):

Accuracy	min. time	max. time	average
1	22314	1561	7650
0.35	10003	1372	4320

We then asked the user if the searched image was among the matches, if so, if it was shown in the first position (Rank 1) and if they were able to play the video on that specific time and find the searched image on it. All participants answered affirmative to these questions.

Comments and feedback

When the participants were asked about their general impression of the search engine, users answered mostly that it's "A great application to find images in videos" and that it is "Quite intuitive and usable". We then asked the participants if they can think about any practical use for it, to which they reported "To specifically find an image show in a news article and search for the video from which it was taken from", also for "Finding advertisements and famous people" and "For video security applications".

When asked about if there is something they don't like or would change, most interesting feedback was about making the interface cleaner with actions such as "Moving the Global Parameters and Search by Tags options below the query and video player", "Hiding the similar image results, as they look irrelevant for this application" and "Some time spent in the look and feel (the site design) would be desirable to make it more friendly".

Chapter 6

Conclusions and Further Work

This thesis report has presented the development of a system to extend the LIRE CBIR system into a CBVR system for retrieving shots within large video datasets of any kind.

We divided this project in three main blocks, Parsing, Indexing and Retrieving, and developed a solution for the requirements of each one of them. After the system was developed and working, we developed a quantitative and a qualitative test to evaluate the system. For the qualitative part, using a large video dataset with more than 1000 hours of video and its ground truth, we evaluated the performance of the system with standard metrics for different image descriptors. As for the qualitative test, volunteer participants assessed whether the system satisfies the user' needs.

The LIRE system becomes then the integration of a series of open-source applications, tools, plugins and modules aimed to work together as an efficient CBVR system. The modular and open-source nature of the project makes it specially interesting to include other tools and descriptors in the future such as sound/music add-ons for retrieval, with or without combining them with video. The scalability of Solr allows for live-indexing of bigger systems and the possibility to integrate more specific descriptors and annotation tools opens a wide range of application possibilities. Integrating LIRE with a deep learning framework such as Caffe could be as interesting as challenging, and perhaps a task to consider. Further documentation and a more simple and automated set-up and deployment of the LIRE system could make it a candidate for professional purposes. The overall results encourage further development and investigations with the tool in each and all of those fields and perhaps many other ones.

Bibliography

- [1] A Araujo, J Chaves, D Chen, R Angst, and B Girod. Stanford i2v: A news video dataset for query-by-image experiments. In *Proc. ACM Multimedia Systems*, 2015.
- [2] A Araujo, J Chaves, D Chen, R Angst, and B Girod. Stanford i2v: A news video dataset for query-by-image experiments. In *Proc. ACM Multimedia Systems*, 2015.
- [3] Andre Araujo, David Chen, Peter Vajda, and Bernd Girod. Real-time query-by-image video search system. In *Proceedings of the ACM International Conference on Multimedia*, pages 723–724. ACM, 2014.
- [4] Ted Boren and Judith Ramey. Thinking aloud: Reconciling theory and practice. *Professional Communication, IEEE Transactions on*, 43(3):261–278, 2000.
- [5] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 401–408, 2007.
- [6] Savvas A Chatzichristofis and Yiannis S Boutalis. Cedd: Color and edge directivity descriptor: A compact descriptor for image indexing and retrieval. In *Computer Vision Systems, 6th International Conference, ICVS 2008, Santorini, Greece, May 12-15, 2008, Proceedings*, pages 312–322, 2008.
- [7] Savvas A. Chatzichristofis and Yiannis S. Boutalis. Fcth: Fuzzy color and texture histogram a low level feature for accurate image retrieval. In *WIAMIS 2008 - Proceedings of the 9th International Workshop on Image Analysis for Multimedia Interactive Services*, pages 191–196, 2008.
- [8] Sourish Chaudhuri and Bhiksha Raj. Unsupervised structure discovery for semantic analysis of audio. *Advances in Neural Information Processing Systems 25*, pages 1187–1195, 2012.
- [9] Jianping Fan, Hangzai Luo, and Ahmed K. Elmagarmid. Concept-oriented indexing of video databases: Toward semantic sensitive retrieval and browsing. *IEEE Transactions on Image Processing*, 13(7):974–992, 2004.

- [10] X Giró-i Nieto, M Alfaro, and F Marqués. Diversity ranking for video retrieval from a broadcaster archive. In *1st ACM International Conference on Multimedia Retrieval (ICMR {\textquoteright}11)*, page 1{\textendash}8, 2011.
- [11] Xavier Giro-i Nieto, Ramon Salla, and Xavier Vives. Digimatge, a rich internet application for video retrieval from a multimedia asset management system. In *Proceedings of the International Conference on Multimedia Information Retrieval, MIR '10*, pages 425–428. ACM, 2010.
- [12] Jonathon Hare, Sina Samangooei, and David Dupplaw. Openimaj and imagerterrier: Java libraries and tools for scalable multimedia analysis and indexing of images. In *ACM Multimedia Conference 2011*, pages 691–694, 2011.
- [13] Jonathon S. Hare, Patrick A S Sinclair, Paul H. Lewis, Kirk Martinez, Peter G B Enser, and Christine J. Sandom. Bridging the semantic gap in multimedia information retrieval top-down and bottom-up approaches. In *CEUR Workshop Proceedings*, volume 187, 2006.
- [14] Jayashree Kalpathy-Cramer and William Hersh. Multimodal medical image retrieval. In *Proceedings of the international conference on Multimedia information retrieval - MIR '10*, page 165, 2010.
- [15] E. Kasutani and A. Yamada. The mpeg-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval. *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*, 1, 2001.
- [16] Anurag Kumar, Pranay Dighe, Rita Singh, Sourish Chaudhuri, and Bhiksha Raj. Audio event detection from acoustic unit occurrence patterns. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 489–492, 2012.
- [17] Mathias Lux. Lire: Open source image retrieval in java. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 843–846, 2013.
- [18] Mathias Lux and Savvas A Chatzichristofis. Lire: lucene image retrieval: an extensible java cbir library. In *Proceedings of the 16th International Conference on Multimedia 2008,*, pages 1085–1088, 2008.
- [19] Mathias Lux and Glenn Macstravic. The lire request handler: A solr plugin for large scale content based image retrieval. In C Gurrin, F Hopfgartner, W Hurst, H Johansen, H Lee, and N OConnor, editors, *MultiMedia Modeling*, pages 374–377. Springer, 2014.
- [20] B. V. Patel1 Meshram and B. B. Content based video retrieval.pdf. *The International Journal of Multimedia & Its Applications (IJMA)*, 4(5):77–98, 2012.
- [21] Duong-Trung-Dung Nguyen, Mukesh Saini, Vu-Thanh Nguyen, and Wei Tsang Ooi. Jiku director: A mobile video mashup system. In *Proceedings of the 21st ACM International Conference on Multimedia*, pages 477–478. ACM, 2013.

- [22] Gwenole Quellec, Mathieu Lamard, Guy Cazuguel, Zakarya Droueche, Christian Roux, and Beatrice Cochener. Real-time retrieval of similar videos with application to computer-aided retinal surgery. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pages 4465–4468, 2011.
- [23] Jennifer Roldan-Carlos, Mathias Lux, Xavier Giró-i Nieto, Pia Muñoz, and Nektarios Anagnostopoulos. Visual information retrieval in endoscopic video archives. 4 2015.
- [24] Cees G M Snoek, Bouke Huurnink, Laura Hollink, Maarten De Rijke, Guus Schreiber, and Marcel Worring. Adding semantics to detectors for video retrieval. *IEEE Transactions on Multimedia*, 9(5):975–986, 2007.
- [25] Song Tan, Chong-Wah Ngo, Hung-Khoon Tan, and Lei Pang. Cross media hyperlinking for search topic browsing. *Proceedings of the 19th ACM international conference on Multimedia - MM '11*, page 243, 2011.
- [26] Adrian Ulges, Christian Schulze, Markus Koch, and Thomas M. Breuel. Learning automatic concept detectors from online video. *Computer Vision and Image Understanding*, 114(4):429–438, 2010.
- [27] Seshadri Padmanabha Venkatagiri, Mun Choon Chan, Ieee Wei, and Tsang Ooi. On demand retrieval of crowdsourced mobile video. 15(c):1–10, 2014.
- [28] C Ventura, Manel Martos, X Giró-i Nieto, V Vilaplana, and F Marqués. Hierarchical navigation and visual search for video keyframe retrieval. In *Advances in Multimedia Modeling*, volume 7131 of *Lecture Notes in Computer Science*, pages 652–654. Springer Berlin / Heidelberg, 2012.
- [29] Chee Sun Won, Dong Kwon Park, and Soo J. Park. Efficient use of mpeg-7 edge histogram descriptor. *ETRI Journal*, 24(1):23–30, 2002.
- [30] Marcel Worring, Cees G M Snoek, Ork De Rooij, Giang P Nguyen, and Arnold W M Smeulders. The mediamill semantic video search engine. *IEEE International Conference on Acoustics Speech and Signal Processing ICASSP 2007*, 4:IV–1213 – IV–1216, 2007.
- [31] Konstantinos Zagoris, Savvas A. Chatzichristofis, Nikos Papamarkos, and Yiannis S. Boutalis. *Automatic Image Annotation and Retrieval Using the Joint Composite Descriptor*. 2010.
- [32] Xiang S Zhou. Cbir: from low-level features to high-level semantics. In *Proceedings of SPIE- The International Society for Optical Engineering*, volume 3974, pages 426–431, 2000.