The *CUBRIK* Project

Human-Enhanced Time-Aware Multimedia Search [*]

Piero Fraternali
Marco Tagliasacchi
Davide Martinenghi
Alessandro Bozzon
Ilio Catallo
Eleonora Ciceri
Politecnico di Milano, Italy
first.last@polimi.it

Francesco Nucci
Vincenzo Croce
Engineering
Italy
first.last@eng.it

Ismail Sengor Altingovde
Wolf Siberski
L3S
Germany
lastname@l3s.it

ABSTRACT

The Cubrik Project is an Integrated Project of the 7th Framework Programme that aims at contributing to the multimedia search domain by opening the architecture of multimedia search engines to the integration of open source and third party content annotation and query processing components, and by exploiting the contribution of humans and communities in all the phases of multimedia search, from content processing to query processing and relevance feedback processing. The CUBRIK presentation will showcase the architectural concept and scientific background of the project and demonstrate an initial scenario of human-enhanced content and query processing pipeline.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*; D.2.11 [**Software Engineering**]: Software architectures

Keywords

CUBRIK, Multimedia, Human computation, Architecture, Application, Logo detection

1. INTRODUCTION

The CUBRIK project is a 36 month large-scale Integrating Project partially funded by the European Commission's 7th Framework ICT Programme for Research and Technological Development, started on October 1st, 2011. CUBRIK embodies several results of the FP6 PHAROS, VITALAS, RUSHES (and others) Multimedia Search Projects, and aims at developing a multimedia search framework to advance the openness of the multimedia search technology to incorporate human and social computation and to enrich the semantics of multimedia content and query processing with the support of temporal and spatial entities. Such a framework will allow an ecosystem of developers to easily custom-design innovative search systems tailored to specific multimedia search scenarios.

1.1 Motivations and Vision

Multimedia search is a highly competitive research field, where software systems need to emulate the human capacity of recognizing the meaning of objects and events, a skill matured in millions of years of evolution. CUBRIK advocates a quantum leap in the openness of multimedia search platforms and the involvement of human beings in all multimedia search processing phases as the key factors for innovation in next-generation multimedia search applications.

Multimedia search can be regarded as the coordinated result of three main processes: *content processing*, *query processing* and *relevance feedback processing*. Content processing aims at acquiring multimedia content, analyzing it for extracting knowledge and indexing such knowledge to make it available for search. Query processing aims at deriving from a user an information need and producing a sensible response to it. Feedback processing aims at gathering quality feedback on the appropriateness of search results and retrofitting the system with such a feedback.

Humans and machines can cooperate in all the three processes and the intervention of humans in the computation can occur at three levels:

- *implicit*, in which the computerized system directly harnesses the sensing capacity of humans, e.g., by using biometry to register the unconscious reactions caused by the exposition to content and search results;
- *decisional,* in which the computerized system exploits the explicit rationality of individuals, e.g., in knowledge extraction or result comparison/evaluation; and
- *social*, in which the computation exploits the capability of humans to work cooperatively towards the achievement of complex tasks in a more efficient and faster manner, e.g., by distributing across large communities micro-tasks in knowledge extraction or result evaluation.

The multimodal analysis of video content offers an example of the fruitful intertwining of human and automatic tasks. In video analysis, due to the complexity of the medium, it is often the case that multiple feature extractors are applied in parallel in a multimodal way (e.g., audio with speaker's

[*] Project Web Site: http://www.cubrikproject.eu/

WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

turn recognition and image with shot boundary detection). Due to the uncertainty of the analysis algorithms, parallel multimodal feature extraction components often provide conflicting opinions (e.g., on the actual start time of a sequence, or on the outdoor/indoor nature of a scene). In this case, the CUBRIK system will be able to detect the conflict and trigger a human to reinforce the confidence of the inference (e.g., a game for spotting the change of a scene in the video). Therefore, CUBRIK bases its architecture on the general principle of *performing content, query and feedback processing by means of open, distributed, and expandable workflows of components, which are assembled to constitute pipelines of work, where also human processing can be incorporated.*

1.2 Objectives

The technical goal of CUBRIK is to build an open search platform grounded on four main objectives:

1. **Advance the architecture of multimedia search**: CUBRIK aims at advancing the architecture for content enrichment, query processing, and relevance feedback by exploiting open source components and pipelines of components and services for search-driven applications. As an application of such an architecture, specific time and location-aware and entity-based components will be incorporated into content and query processing pipelines and integrated with external components instantiating current state-of-the-art multimedia processing algorithms, so as to provide unique multimedia search capabilities.

2. **Humans in the Loop**: an innovative human computation framework will be developed that combines the conventionally segregated areas of user behavior analysis, crowdsourcing, social network and trust analysis, and games with a purpose (GWAPs) to support the full range of contributions that humans make to enrich the multimedia spaces that they occupy. Languages for the specification and execution of processes will be extended, by allowing complex tasks to be executed by machines and human actors at various levels: sensorial, decisional, and social, with optimization policies based on Social Network Analysis and task allocation policies integrating crowdsourcing practices.

3. **Open the search box**: the component-based development of tailor-made multimedia search applications will be enabled, so as to create a community of practice, where technology providers, application designers and software integrators will be able to construct end-to-end multimedia search applications by integrating their components with complementary technologies packaged as community-built pipelines. To this end, CUBRIK results will be made available under open source licensing models (e.g., LGPL and Eclipse Public License for the software and Common Creative for deliverables including content collections, ground truth data sets, and entity repositories).

4. **Start up a search business ecosystem**: the CUBRIK platform will be validated by implementing application scenarios in real world conditions and by building up a business ecosystem for vertical search domains, made of technology developers, software integrators, social network or crowdsourcing service providers, with content owners whose business model includes offering users "search on the side" and SMEs who offer customized search solutions.

2. THE CUBRIK ARCHITECTURE

CUBRIK is a distributed system layered in four main tiers, as shown in Figure 1. The *Content and user acquisition tier* is responsible for registering content and users into the system. The *Subscription Manager* handles the registration to the platforms of two classes of users: searchers and performers. Searchers use CUBRIK applications for finding and interacting with information; they may be exploited to get feedback on query result quality (e.g., via click stream analysis). Performers execute tasks (via gaming or Query&Answer) to provide contribution, semantic annotation, and conflict resolution. Content is added to a CUBRIK platform via upload or by scheduled crawling; crawling may import into CUBRIK external metadata in popular formats (e.g., Dublin Core, or a useful subset of MPEG-7). Content registration gives content element an internal ID, and then stores: the content element in raw format; the associated crawled or manual metadata, and rights information.

In the *Content Processing Tier*, the *Content Processing Manager* listens to a queue of pending content processing requests and is responsible for starting, suspending, resuming, terminating, and rescheduling tasks. It processes contents at different granularities: collection, sub-collection, content element, and derivative content element (e.g., video key frames). Process control is implemented on top of the SMILA pipeline engine, which orchestrates the execution of pipelines made of graphs of Web service or local component calls, which embody both general-purpose (e.g., video/ audio segmentation) and domain-specific (e.g., face recognition) content processing logic. Content processing is incremental: the same content element can be subject to several processing pipelines, where each pipeline adds a different annotation. The goal is to enable real-time search, e.g., to process a content-based query issued by a searcher, while the infrastructure is performing full annotation of stored collections. To this end, pipelines are associated with a "priority" flag (e.g., real-time, deferred, etc.) that instructs the SMILA pipeline engine about the importance of a given content processing task. Control flow among pipelines is expressed declaratively (e.g., as a macro-pipeline formed of sub-pipelines, with runtime conditions governing the flow of control). The output of a content processing task consists of: derivative content (e.g., key frames, thumbnails, audio summaries); low-level features, facts (i.e., annotations + confidence values); entities; and conflicts (i.e., low confidence facts and contradictory facts).

The *Conflict Manager* is the core component for integrating human computation; it manages the set of conflicts and the assignment of conflicts to applications and performers. In the simplest case, conflicts can be assigned to an application, which manages their allocation to performers, possibly using data provided by the Performer Manager. This is the typical case for simple GWAP applications, where the interaction logic is user-independent and only basic profile data, like the skill level of the gamer, are employed to decide which conflict to present. Alternatively, conflicts can be assigned to an application-performer pair: in this case, the association of the performer is managed by CUBRIK

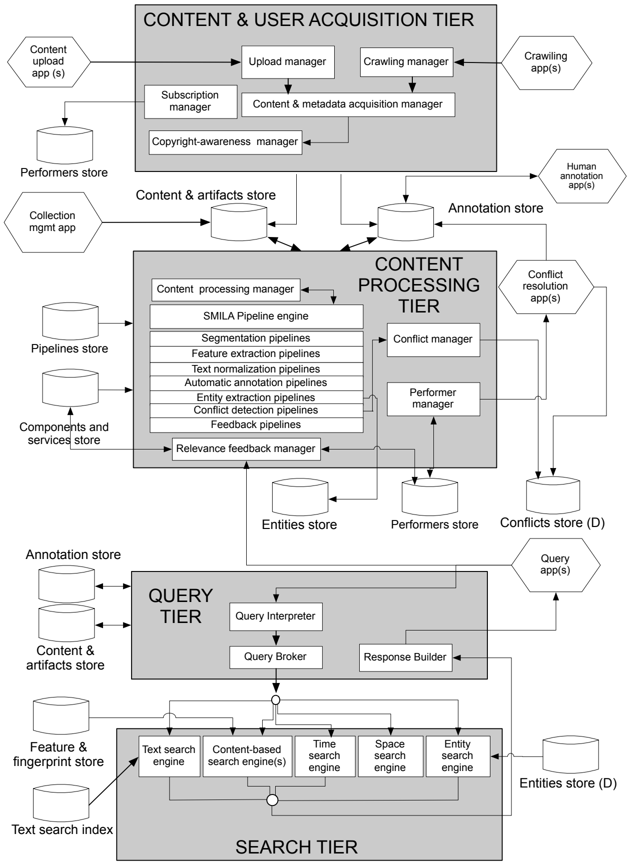

Figure 1: The architecture of the CUBRIK system

and the application routes the conflict to the performer suggested by the platforms. This is the case of more personalized applications, like Q&A, where a mix of history, profile, and trust data of the performer can be used to route the most appropriate questions. A conflict resolution application can also be an existing third-party application (e.g., a crowdsourcing application on top of a commercial platform). The Conflict Manager is responsible for closing a conflict and storing the produced facts; it can implement a policy of escalation (e.g., re-routing a hard conflict to a more skilled performer, or to the CUBRIK admin). The *Performer Manager* is responsible for keeping data about performers (profile, social network centrality measures, history of solved conflicts, throughput, quality of decision, etc.), which are used to optimize task allocation.

Some pipelines are designed to receive feedback from the user on the results of a query. This feedback is routed to a *Relevance Feedback Manager* module that updates the level of trust of performers (human and automatic) in the component and performer store.

The *Query Processing Tier* consists of one or more *Query Applications*; a query app contains the front-end for issuing queries and viewing results; queries are expressed according to a multimodal query language, serialized and submitted to a CUBRIK platform (through Web services API); results are organized according to an application-dependent result schema, serialized, and returned as responses from CUBRIK to the application. Typically, results consist of references to matched content elements; match details for each content element (e.g., time-stamp of match for continuous media), and annotation values for the collection of results (to enable faceted search). The query protocol allows the CUBRIK application to declare the requested result information profile (e.g., only content elements, content elements + associated annotations, content elements + associated annotations + annotation values for the result list collection). The *Query Interpreter* analyzes the query and understands its class. Classes of supported queries are: Keyword, Visual similarity (Still image and video), aural similarity, and multimodal (keyword + one similarity criterion). The *Query Broker* translates the query into the format expected by the search engine(s) of the Search Tier and dispatches the query or sub-queries to the relevant search engine(s). For similarity queries, the Query Broker acts as a content acquisition application. It sends the query content element to the content acquisition and registration manager, with a high priority flag in order to have the content element analyzed and indexed immediately. The *Response Builder* normalizes and fuses the responses from the search engines(s) and creates a single result list, to be returned to the query app. The Query Interpreter, the Query Broker, and the Response Builder have a plug-in structure, whereby it is possible to add a new search engine and register the logics for analyzing the query, translating it, and processing the result list.

Finally, the *Search Tier* contains a collection of independent search engines. Each search engine can access the content and annotation store(s) to build/rebuild its indexes. Indexing is independent and asynchronous w.r.t. content processing and acquisition. Each search engine listens to the content processing manager events, in order to understand when to build, re-build, extend, and update its indexes.

3. THE CUBRIK APPLICATIONS

The goal of the CUBRIK applications is the technical and functional validation, exploitation and testing of the technologies, framework and results developed and accomplished within the CUBRIK Integrated Projects. Two applications are planned to showcase CUBRIK's achievements.

The first application refers to the *History of Europe*. This application will be build upon the audio-visual content of a specialized digital library developed by CVCE (`www.ena.lu`) which is considered as a multimedia reference on this subject. Content consists of various materials like TV/Cinema news on historical events, archives from institutional collections, interviews and on purpose video editing. The CUBRIK History of Europe application will enhance the quality and the usefulness of audiovisual content by making it more searchable (by event, by region with rich query expansion mechanisms). Furthermore, the CUBRIK platform will support a true experience of mixing human and social intelligence (e.g., digital humanities methodologies from researchers, annotations and tags from groups of historic expert amateurs, or groups of residents in some specific region) with human computation (e.g., data analysis, automatic classification, knowledge extraction).

The second one is the *CUBRIK search for SME* innovation application that will focus on searching audiovisual content and information useful for SME innovation. It will provide the possibility to manage information for market and technology analysis, to be realized within Open Innovation Processes inside SME business. The opportunity to use the CUBRIK advanced search techniques can effectively support these processes in many different ways. On one hand, it can demonstrate that human computation as well as social media search techniques could be fundamental to build the new "killer application" in market analysis. On the other hand, particular importance will be given to space and time extension as trend detection is particularly related to timing.

4. THE CUBRIK PRESENTATION

The CUBRIK presentation will feature a general introduction to the system and its objectives, followed by the explanation of the architecture. The core of the presentation will be a demo, whereby attendees will be exposed to the initial implementation[1] of the main concepts and innovations of the system. The demo will focus on the content processing pipelines; it will start from the acquisition of a small content collection to be analyzed; then a multimodal content processing pipeline, made of sub-pipelines, will be launched and the attendees will be given the opportunity of observing the system behavior while producing the annotation of media content. Particular attention will be devoted to the detection of two kinds of conflicts: *low confidence annotations*, extracted by algorithm with imperfect training; and *conflicting annotations*, extracted by two annotation components that apply distinct algorithms to the same content element.

The demo will feature a *trademark logo detection pipeline* illustrated in Figure 4, which exploits human performers with the ambitious goal of increasing precision and recall with respect to fully-automated solutions. The pipeline receives as input a textual keyword indicating the name of the trademark. The query is expanded by means of a text-based image retrieval component (e.g., Google Images), to identify the different versions of the logos. In this phase, human computing is adopted to assist the process of validation and segmentation of the retrieved images to be used as input to the next component. Then, a content-based image retrieval component (i.e., VLFeat) looks for occurrences of the different versions of the logos in a collection of video clips. Those matches characterized by low confidence are good candidates to be dispatched to the Conflict Manager of the CUBRIK architecture, in such a way to exploit human knowledge.

5. ACKNOWLEDGMENTS

We wish to thank all the people involved in the development of the CUBRIK project.

6. ADDITIONAL AUTHORS

Fausto Giunchiglia, Universita di Trento, fausto@dit.unitn.it, Wolfgang Nejdl, Gottfried Wilhelm Leibniz Universitaet Hannover, surname@l3s.de; Martha Larson, Delft University of Technology, M.A.Larson@tudelft.nl; Ebroul Izquierdo, Queen Mary, University of London, ebroul.izquierdo@elec.qmul.ac.uk; Petros Daras, CERTH, daras@iti.gr; Otto Chrons, Microtasks, otto.chrons@microtask.com; Ralph Traphoener and Bjoern Decker, Attensity, name.surname@attensity.com; John Lomas, Nexture, j.lomas@nexture.it; Patrick Aichroth, Fraunhofer IDMT, ath@idmt.fraunhofer.de; Jasminko Novak, European Institute for Participatory Media, j.novak@eipcm.org; Ghislain Sillaume, Centre Virtuel de la Connaissance sur l'Europe, ghislain.sillaume@cvce.lu; F. Sanchez Figueroa, Homeria, fsfigueroa@homeria.com; Carolina Salas-Parra, Inn. Engineering, leonor.salas-parra@innovationengineering.eu;

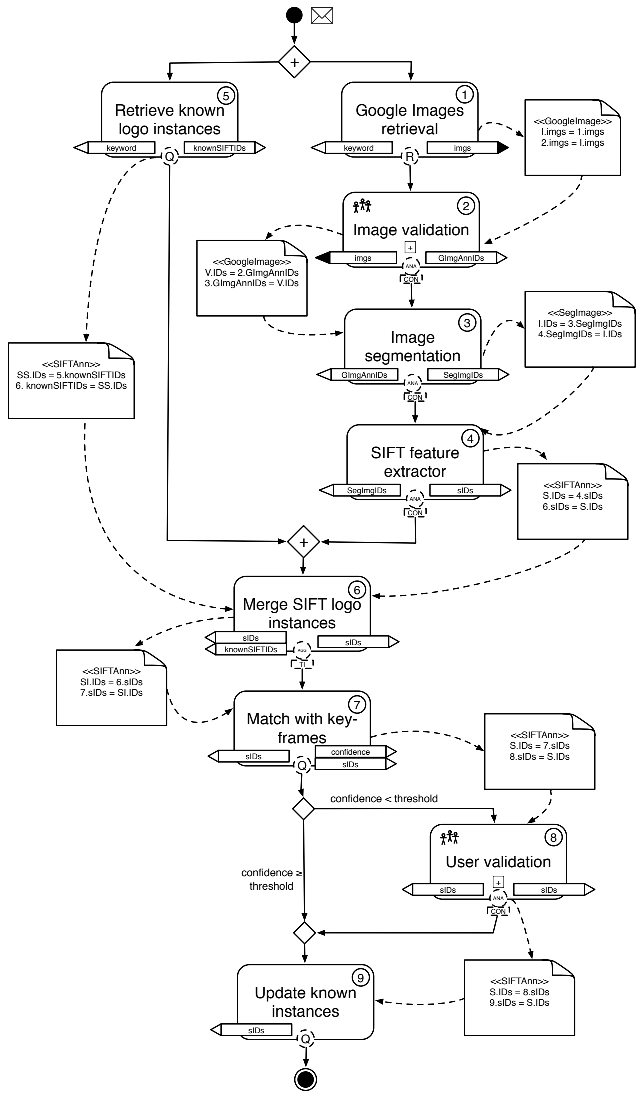

Figure 2: The demo search pipeline

[1] The project will be at month 6 at the time of the Conference.