Generalitat de Catalunya
Corporació Catalana
de Mitjans Audiovisuals

UPC

Reordenació i agrupament d'imatges resultants d'una cerca de vídeo

Per Mónica Alfaro Vendrell

Tutors:
Xavier Giró
Xavier Vives

Escola d'Enginyeria de Terrassa (EET) – Gener 2011

Resum

La recuperació de vídeo a través de consultes textuals es una practica molt comú en els arxius de radiodifusió. Les paraules clau de les consultes son comparades amb les metadades que s'anoten manualment als *assets* de vídeo pels documentalistes.

A més, les cerques textuals bàsiques generen llistes de resultats planes, on tots els resultats tenen la mateixa importància, ja que, es limita a avaluar binàriament si la paraula de cerca apareix o no entre les metadades associades als continguts. A més, acostumen a mostrar continguts molt similars, donant al usuari una llista ordenada de resultats de poca *diversitat* visual. La redundància en els resultats provoca un malbaratament d'espai a la interfície gràfica d'usuari (GUI) que sovint obliga a l'usuari a interactuar fortament amb la interfície gràfica fins localitzar els resultats *rellevants* per a la seva cerca.

La aportació del present projecte consisteix en la presentació d'una estratègia de reordenació i agrupació per obtenir *keyframes* de major rellevància entre els primers resultats, però al mateix temps mantenir una diversitat d'*assets*. D'aquesta forma, aquestes tècniques permetran millorar els sistemes de visualització d'imatges resultants d'una cerca de vídeo. L'eina global es dissenya per ser integrada en l'entorn del Digition, el gestor de continguts audiovisuals de la Corporació Catalana de Mitjans Audiovisuals.

Agraïments

En primer lloc, vull donar les gràcies a en Xavi Giró, tutor, i en Xavi Vives, co-tutor, per la oportunitat que m'han donat al formar part en aquest treball, pel suport rebut i per haver estat una guia durant aquests mesos, fet que m'ha permès avançar més ràpidament.

En segon lloc, al equip del GPI, en particular a en Manel Martos, Khristina López, Jaume Sastre, per l'ajuda rebuda i la dedicació que han mostrat durant tot el treball, i l'ambient de treball creat.

En tercer lloc, al equip de la CCMA, en concret a en Ramon Salla i la Irene Zeller, per la seva col·laboració, ajuda constant que m'han permès aprendre ràpidament el llenguatge de programació i les comunicacions entre client servidor.

índex

1. Introducció

La recuperació de vídeos i imatges s'ha convertit en una àrea d'interès degut al ràpid i continu creiexement de informació que s'emmagatzema en els repositoris de les empreses relacionades amb la producció audiovisual. Per tal de gestionar i utilitzar els recursos multimèdia els usuaris han de ser capaços de realitzar cerques multimodals (text, imatge i àudio) de forma eficient i eficaç. Aquest escenari ha propiciat la investigació per trobar noves solucions a la cerca de vídeos i imatges.

La recuperació de vídeos s'ha enfocat, tradicionalment, en solucions basades en cerques textuals, on les paraules claus es comparen amb text associat a aquest, com per exemple les transcripcions de la parla, subtítols, reconeixement de text (OCR) i altres dades d'interès: autor, data, format, etc. Però, els descriptors textuals no sempre aporten els resultats més rellevants que espera l'usuari final perquè no són suficients per descriure amb precisió el contingut visual d'una imatge. Per això, en els casos on és difícil explicar amb paraules el concepte que s'està buscant és molt millor expressar la consulta visualment, en forma d'exemple.

A més, les cerques textuals bàsiques generen llistes de resultats planes, on tots els resultats tenen la mateixa importància, ja que, es limita a avaluar binàriament si la paraula de cerca apareix o no entre les metadades associades als continguts. A més, acostumen a mostrar continguts molt similars, donant al usuari una llista ordenada de resultats de poca *diversitat* visual. La redundància en els resultats provoca un malbaratament d'espai a la interfície gràfica d'usuari (GUI) que sovint obliga a l'usuari a interactuar fortament amb la interfície gràfica fins localitzar els resultats *rellevants* per a la seva cerca.

Tècniques com la *reordenació*, que ordena els resultats obtinguts d'una cerca segons uns criteris diferents als de cerca, i *l'agrupament*, que forma conjunts de resultats que tenen coses comunes, permeten millorar la presentació i visualització dels resultats, de forma que a l'usuari li resulta més fàcil trobar resultats rellevants i diversos.

En enquestes realitzades als usuaris professionals de la CCMA en el marc del projecte i3media s'aprecia que dins d'un ventall de requeriments, els usuaris valoren preferentment una presentació intel·ligent dels resultats d'una eina de consulta, *Figura 1*.

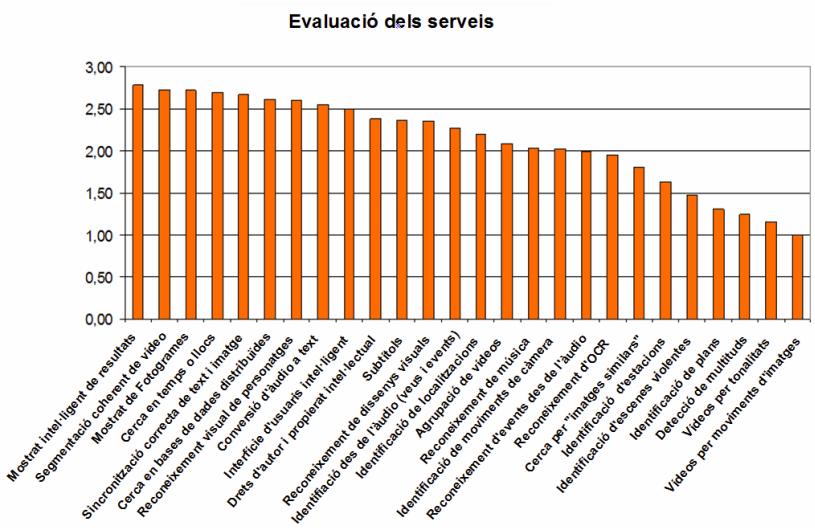


Figura 1. Requeriments més valorats

D'acord amb aquests estudi de les necessitats dels usuaris, el primer objectiu que persegueix aquest Projecte Final de Carrera (PFC) és dissenyar i implementar tècniques per millorar els sistemes de visualització de resultats a través de la reordenació i l'agrupament basats en criteris de similitud visual.

Com a segon objectiu es proposa integrar aquestes noves tècniques dins de la GUI Digition, l'eina amb la que compta la Corporació Catalana de Mitjans audiovisuals[1] (CCMA) i, en concret, la Televisió de Catalunya. Per fer les cerques textuals i visuals del material audiovisual digitalitzat i emmagatzemat al seu repositori de dades.

La CCMA és una empresa catalana puntera en el camp de mitjans audiovisuals que té la missió d'oferir a tos els ciutadans un servei públic audiovisual de qualitat. Aquest grup està format per Televisió de Catalunya, Catalunya Ràdio, CCRTV Interactiva, CCRTV-ASI i Activa Multimèdia Digital. Aquest és un projecte de col·laboració universitat - empresa entre la UPC i la CCMA emmarcat en el projecte estatal Buscamedia[2].

El treball s'ha desenvolupat en dues institucions diferents:

- Per una banda, el Grup de Processament de la imatge de la Universitat Politècnica de Catalunya on s'han dissenyat i implementat els algoritmes de reordenació durant tot el projecte.
- Per l'altre banda, la CCMA on s'ha desenvolupat les millores a nivell de GUI durant els tres últims mesos.

[1] http://www.ccma.cat
[2] http://www.cenitbuscamedia.es/

2. Requeriments

2.1 Requeriments tècnics

L'objectiu general d'aquest treball és millorar la cerca de vídeos a partir d'una consulta textual tot proveint a l'usuari d'una visualització eficient dels resultats en una interfície gràfica accessible des d'un navegador web.

Tot i que el treball tindrà com a base i objectiu el concepte anterior, s'han definit dues tasques ben diferenciades: d'una banda la programació d'un motor de reordenació i agrupament que millori la presentació dels resultats obtinguts donada una cerca, i d'altra banda la integració d'eines que permetin millores en la visualització del Digition.

2.1.1 Descripció dels elements existents

Digition

El Digition és el repositori de la CCMA on es fa la ingesta dels continguts audiovisuals per la seva posterior recuperació a través d'un sistema de gestió de bases de dades. Els seus usuaris són els documentalistes i periodistes de TVC (Televisió de Catalunya)), que l'utilitzen per a produir els continguts de vídeo que genera l'empresa..

Un conjunt de fitxers de vídeo, d'àudio i de metadades relacionades amb el vídeo i amb cadascun dels *keyframes* s'anomena *asset*. Els *keyframes* són les imatges clau que s'extrauen automàticament del vídeo per agilitzar la cerca i poder fer un primer cop d'ull al vídeo. Els *keyframes* són les imatges base sobre les quals es fan les cerques visuals, reduint significativament l'esforç que significaria indexar cadascuna de les imatges que composen un vídeo.

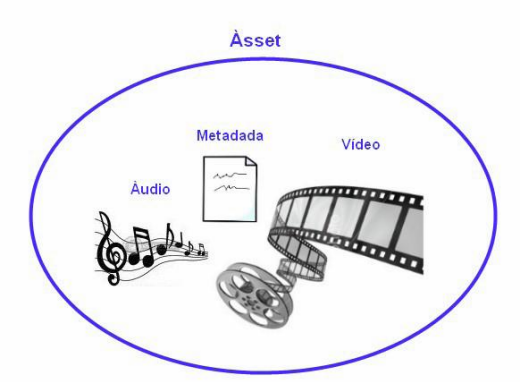


Figura 2. Composició d'un asset

Només un nombre reduït *d'assets* marcats com importants passen pel departament de documentació on s'anoten manualment per tal de poder-los recuperar una vegada arxivats dins de la seva base de dades. Aquestes descripcions poden tenir un caràcter global de tot *l'asset*, com per exemple "partit de futbol: Barça 3 - Athlètic 1" o bé poden referir-se a un segment temporal que incorpora codi de temps d'entrada i de sortida de la descripció "00:05:30 - El president ensopega i cau de la tarima - 00:06:03". Anomenarem a partir d'ara aquestes descripcions estrats. En aquest projecte només es considera les anotacions a nivell

d'asset, per tant, això vol dir que tots els *keyframes* d'un mateix *asset* tenen les mateixes metadades textuals.

La GUI del Digition consta de quatre mòduls:

Figura 3. Mòduls del Digition

- El mòdul de **resultats** ofereix un llistat de resultats ordenats, segons la probabilitat de que *l'asset* resultant sigui d'interès a partir de la cerca textual que s'ha dut a terme inicialment.
- El **visor**, on es pot visualitzar el vídeo amb l'àudio de forma tradicional o bé fer una visualització d'aquest a partir d'un *keyframe* predefinit.
- El **mòdul de *keyframes*** mostra els *keyframes* de *l'àsset* que sigui seleccionat al mòdul de resultats amb els seus corresponents codis de temps.
- El **mòdul de metadades** mostra tota la fitxa de *l'àsset*. Les metadades inclouen la identificació de *l'àsset*, el títol, la durada, la data de creació, descripcions per estrats (fragments de *l'àsset*), l'instant del vídeo al qual correspon cada *keyframe*, etc.

2.1.2 Algoritme de reordenació

L'objectiu que es vol resoldre amb l'algoritme de reordenació és el de la recuperació *d'assets* d'una cerca de text sobre un camp de les seves metadades a nivell *d'asset*, en concret sobre el camp *dTema*.

Dins d'aquest context, la reordenació s'adreça al problema de les anotacions manuals a escala *d'asset* quan aquestes només són rellevants per a un subconjunt dels *keyframes* dins *l'asset*. Aquest és el cas del camp de metadades *dTema* de la CCMA, que s'associa a tots els

keyframes que es troben dins *l'asset* anotat, tant si les imatges representen els conceptes descrits com si no.

Per exemple, si es considera un *asset* d'un programa de noticies on la diversitat de temes és molt gran i les seves anotacions s'han realitzat a nivell *d'asset*, les *metadades* només indicaran que dins *l'asset* existeixen *keyframes* que representen el concepte expressat, però no tots. El problema apareix en el moment de la cerca per text, perquè a banda de poder recuperar els *keyframes* rellevants a la consulta se'n poden obtenir d'altres que no.

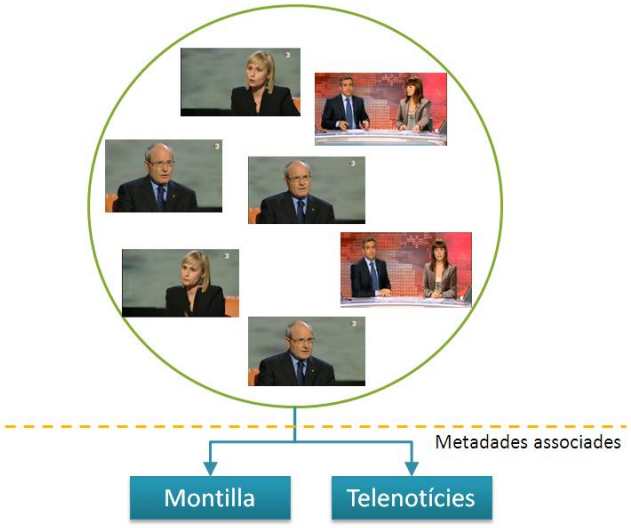


Figura 4. Metadades a escala d'asset

Per altra banda, es demana que la reordenació també consideri la problemàtica de la diversitat *d'assets*, ja que els usuaris de la CCMA estan principalment interessats en recuperar peces de vídeo diferents, no tant imatges concretes.

En conclusió, els reptes que es plantegen són:

1. Donat un conjunt de resultats, prioritzar els *keyframes* rellevants al concepte de la cerca.
2. Donats molts *keyframes*, mostrar diversitat d'*assets* en les primeres posicions, en comptes de mostrar resultats redundants a nivell d'*asset*.

2.1.3 Algoritme d'agrupament

L'objectiu que persegueix l'agrupament és organitzar els *keyframes* obtinguts d'una cerca en conjunts d'imatges similars des del punt de vista visual. Amb aquesta tècnica es preten fer un ús eficient de la superfície de visualització de resultats i la reducció de la redundància visual. Així doncs, es pretén desenvolupar un algoritme capaç d'identificar *keyframes* duplicats o molt similars. Quan es detectin grups de *keyframes* similars també es pretén identificar quin d'ells és el millor prototipus per a representar el conjunt.

Per altra banda, però, es vol mantenir la diversitat d'*assets* entre els primers resultats. Així doncs, l'agrupament d'imatges només s'aplicarà en *keyframes* que pertanyin a un mateix *asset*.

2.1.4 Integració i millores a nivell de GUI

La interfície ha de ser capaç de treure rendiment a les tècniques de reordenació i agrupament descrites anteriorment per tal d'aprofitar de forma eficient l'espai de visualització de la GUI. La interfície haurà de mostrar els resultats prioritzant els *keyframes* en les primeres posicions del sistema de reordenació i, en segon lloc, haurà de permetre a l'usuari explorar els agrupaments de *keyframes* generats. Per últim, la interfície s'haurà d'integrar dins del Digition per ser utilitzada pels documentalistes de TVC.

Durant el desenvolupament d'aquest projecte es va realitzar una reunió amb Imma Rull Perello del departament de Documentació de TV3. Com a usuària professional de la interfície Digition ens va expressar la seva sensació de pèrdua en la cerca per imatges a través del domini temporal quan els *assets* són molt grans. A més, va donar el vist-i-plau a la incorporació d'aquestes tècniques i va aportar la seva opinió sobre el disseny de la GUI per tal de millorar les cerques per imatge.

2.1.5 Avaluació de la diversitat

Degut a que no s'ha trobat cap mesura que s'adaptés al cas que es presenta en aquest projecte, l'estudi de la diversitat d'*assets* d'una llista resultant requereix dissenyar una nova expressió que permeti avaluar-lo. En concret, es demana que el millor valor sigui 1.0 i el pitjor 0.0 i que, a més, penalitzi més la homogeneïtat en les primeres posicions que no pas en les últimes.

2.2 Requeriments d'entorn

La realització de les dues parts del projecte ha comportat treballar en dos entorns de treball diferents, un proposat per la UPC i un segon per la CCMA.

Per una banda, el treball realitzat amb el grup d'imatge de la UPC, (els algoritmes de reordenació i agrupament) s'han implementat en Java[3]. El codi a desenvolupar ha estat integrat en el marc del projecte UPSeek, un sistema de recuperació d'imatges desenvolupat pel grup de recerca. Aquest codi es gestiona a través d'un repositori de control de versions del tipus SVN i combina les contribucions de sis desenvolupadors que treballen en paral·lel. La plataforma de desenvolupament recomanada és l'Eclipse amb el connector de Web Tool Plataform. Per altra banda, les cerques basades en similitud visual s'han generat amb el programa *ranker* de la llibreria ImagePlus desenvolupada pel GPI. En el moment de realitzar aquest PFC, el sistema UPSeek no disposava de cap sistema d'indexació que permetés una cerca ràpida, circumstancia que ha obligat a treballar amb resultats pre-calculats. A nivell de documentació, la UPC també ha demanat a l'estudiant que col·laborés periòdicament en el bloc BitSearch[4] per tal de narrar en anglès els avenços del projecte. Aquests escrits es lliuren també en la present memòria en forma d'annex II.

Per l'altre banda, en el treball realitzat a la CCMA s'ha utilitzat Google Web Toolkit [5](GWT). El GWT és un framework creat per Google que permet ocultar la complexitat de diversos aspectes de la tecnologia AJAX. El seu concepte és bastant senzill, bàsicament tradueix el codi en Java a HTML i Javascript utilitzant un connector que s'afegeix a l' entorn de desenvolupament (IDE) com l'Eclipse.

[3] http://www.oracle.com/technetwork/java/index.html
[4] http://bitsearch.blogspot.com
[5] http://code.google.com/intl/es-ES/webtoolkit/

3. Estat de l'art

Aquest projecte proposa un sistema de reordenació i agrupament per fer front a la problemàtica que es planteja a la introducció. Donada una consulta s'espera:

1. Mostrar les imatges més rellevants a la consulta en les primeres posicions.
2. Aprofitar l'espai de visualització dins de la GUI per mostrar diversitat de resultats.

3.1 Reordenació

La reordenació de vídeo es presenta com una solució per fer front a la gran redundància en els resultats de les consultes a bases de dades de vídeo. La reordenació és la organització automàtica d'un conjunt inicial de resultats amb algun criteri auxiliar diferent al que s'ha utilitzat per executar la cerca inicial.

Classificació d'algoritmes de reordenació

Les tècniques de reordenació d'imatges i vídeo es poden dividir en dues grans famílies segons: basades en pseudo-retroacció de rellevància o basades en similitud visual.

a) Basat en pseudo-retroacció de rellevància

Les tècniques de pseudo-retroacció per rellevància estan inspirades en els sistemes de retroacció. La retroacció de rellevància [1] treu avantatge dels judicis de rellevància del usuari en processos de recuperació de documents. El procediment és el següent:

1. L'usuari formula una consulta simple i obté una llista inicial amb resultats.
2. L'usuari marca els documents rellevants i els que no.
3. El sistema calcula una representació millor de la informació basada en aquesta retroacció
4. Una o més iteracions es duen a terme.

La idea és que l'usuari no sap molt bé el que esta buscant fins que no el veu. La formulació de les consultes pot ser difícil i aquest sistema pot simplificar el problema a traves de la iteració, ja que, facilita el vocabulari i el descobriment del concepte.

En canvi, la pseudo-retroacció de rellevància no requereixen de la interacció de l'usuari per a que aporti informació de rellevància sobre els resultats obtinguts, el sistema suposo que els primers resultats de la llista són rellevants (pseudo-positius) i els darrers no rellevants (pseudo-negatius). Aquesta opció requereix que el llistat inicial estigui ordenat. Els pseudo-resultats serveixen per entrenar classificadors que s'apliquen a cada imatge inicial per generar una nova llista amb la puntuació obtinguda.

b) Basat en similitud:

Els sistemes de reordenació basats en la similitud visual assumeixen que els vídeos amb descriptors visuals o textuals similars tenen també rellevància similar.

Aquestes tècniques, a diferencia de les de pseudo-retroacció per rellevància, no requereixen que els resultats obtinguts estiguin ordenats. Per això, degut a que les cerques

textuals de la CCMA no donen puntuacions inicials l'estudi sobre l'estat de l'art s'ha centrat en les tècniques basades en similitud.

En la ***Figura 5*** es pot veure, a grans trets, un esquema del procés que segueixen aquests sistemes de reordenació.

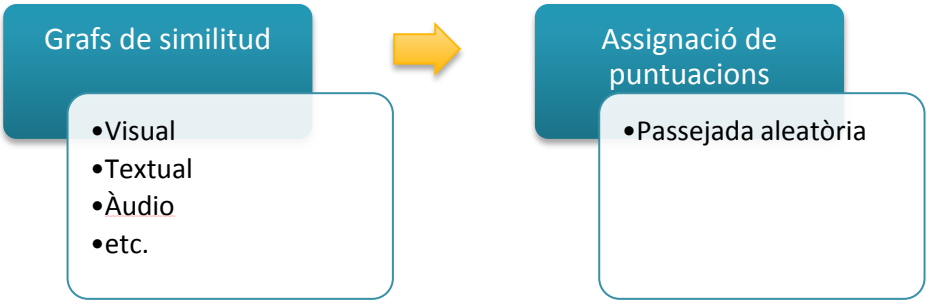


Figura 5. Esquema dels sistemes de reordenació basats en similitud

3.1.1 Grafs de similitud

L'ús de grafs de similitud en reordenació de resultats visuals s' inspirant en l'algoritme PageRank [13] que Google utilitza en el seu motor de cerca per a pàgines webs. Aquest, mesura el grau d'importància d'un document de forma numèrica i permet situar els resultats més rellevants en primer lloc. Bàsicament, reflecteix l'idea de que un document és important si molts altres documents fan referència a ell i, si els documents als quals ell fa referència són importants.

Traslladat al domini visual, la importància d'una imatge es tradueix al nombre d'imatges similars segons el criteri de construcció del graf (visual, textual, etc.).

Per tal de representar la relació entre imatges es genera un graf de similitud on cada node representa una imatge i les arestes la similitud entre elles. Si ens centrem en el cas de similitud visual mostrat a la ***Figura 6***, les connexions entre nodes es construeixen a partir de calcular la distancia visual entre les imatges i fixar un llindar per sota del qual es consideren les imatges "similars".

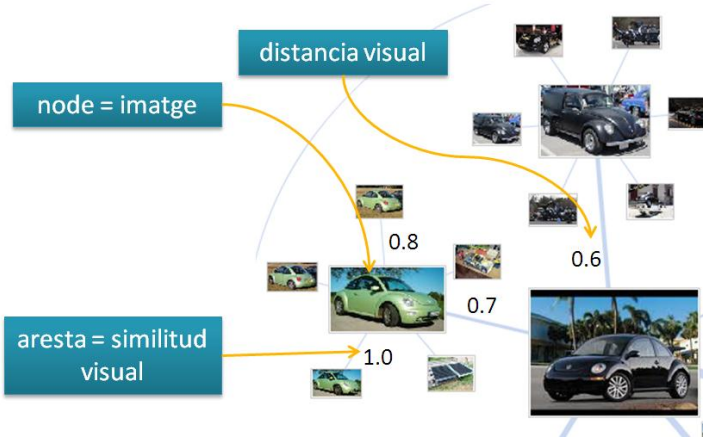


Figura 6. Graf de similitud visual

Per exemple, Jing *et al.* [1] presenten aquest concepte aplicat a les imatges. Considera les imatges com a pàgines web o documents i les seves referències com la similitud visual, a través d'un procés on s'assignen pesos numèrics a cada imatge, es mesura la seva importància relativa amb les altres imatges que s'estan considerant.

3.1.2 Assignació de puntuacions

Després de construir els grafs de similitud cal aprofitar-los per assignar puntuacions als seus nodes. Els nodes amb més puntuacions seran aquells que tinguin més vincles de similitud amb altres nodes i que, a més, aquesta similitud sigui alta.

Una solució molt comuna es resoldre la reordenació a partir d'una passejada aleatòria a través del graf de similitud. Bàsicament, assigna puntuacions als nodes del graf basant-se en la quantitat d'arestes que rep i en els pesos d'aquestes. La puntuació indica la probabilitat d'estar en un node del graf en un instant donat.

Passejada aleatòria

La passejada aleatòria o *random walk* en anglès, és una formulació matemàtica de la trajectòria que resulta de fer successius pasos aleatoris.

Considerant un vianant que es troba a un node del graf de similitud i que cada pas que fa el mou a un dels nodes veïns. La probabilitat d'escollir un veí depèn de la similitud visual, és a dir, una distancia més petita representa una probabilitat alta per a que es mogui cap a ell.

La decisió de moure's de un node a un altre és una iteració infinita, per tant, el vianant es desplaçarà sobre el gràfic per sempre. Sota aquest escenari, la passejada aleatòria calcula la probabilitat de trobar al vianant en cada node del graf en un determinat moment.

Dins del context del graf de similitud, aquesta probabilitat proporciona la puntuació de rellevància per cada imatge considerada en base a les similituds dels resultats recuperats.

Una vegada s'ha descrit i justificat la importància d'aquest algoritme es passa a presentar la seva formulació matemàtica [4].

En primer lloc, es considera un moment de temps k. En aquest instant, la probabilitat de trobar al vianant en un node j es coneix com x(k)(j). Des de la perspectiva de similitud visual, aquesta probabilitat depèn de la probabilitats del veïns en un instant de temps anterior (k-1) i de la probabilitat de transició definida per les arestes d'entrada al node i. D'altra banda, l'expressió proposada introdueix la puntuació prèvia v(j) provinent de resultats previs o obtinguts amb una altre criteri que permeten influenciar en la reordenació final. Aquesta influencia es pot controlar amb el paràmetre alfa (α).

$$x_k(j) = \alpha \sum_{i \epsilon B_j} x_{(k-1)}(i) p_{ij} + (1-\alpha) v(j)$$

on,

$x_{(k)}(j)$: Probabilitat d'estar en un node j en un instant de temps k
v(j): Puntuació obtinguda per la cerca prèvia

p_j: Probabilitat de transició d'un node i a j

α: Factor que controla el pes de la cerca textual i visual

Una vegada que la probabilitat d'estar en un cert node en un instant k, l'expressió es pot estendre al cas estacionari i considerar que totes les probabilitats en el graf s'actualitzen iterativament fins a la convergència. Aquesta probabilitat estacionaria és:

$$x_\pi(j) = \alpha \sum_{i \epsilon B_j} x_\pi(i) p_{ij} + (1-\alpha) v(j)$$

on,

$x_\pi(j)$: Probabilitat estacionaria d'estar en un node j quan convergeixi.

L'equació anteriors pot formular en una matriu considerant un vector que conté la probabilitat estacionaria dels diferents nodes.

$$[x_\pi(1) \cdots x_\pi(n)] = \alpha \cdot [x_\pi(1) \cdots x_\pi(n)] \cdot \begin{bmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{bmatrix} + (1-\alpha) \cdot [v(1) \cdots v(j) \cdots v(n)]$$

$$x_\pi^T = \alpha \cdot x_\pi^T \cdot P + (1-\alpha) v^T$$

on,

x_π: Probabilitat estacionaria $x_\pi^T \equiv [x_\pi(1) \cdots x_\pi(n)]$

P: Matriu de transició $P \equiv [p_{ij}]_{nxn}$

v_π: Puntuacions textuals $v^T \equiv [v(1) \cdots v(n)]$

L'expressió algebraica anterior pot ser manipulada introduint un vector auxiliar d'uns. Com a resultat, l'equació anterior pot ser re formulada com:

$$e \rightarrow vector\ auxiliar\ d'1$$

$$x_\pi^T \cdot e \equiv [x_\pi(1) \cdots x_\pi(n)] \cdot \begin{bmatrix} 1 \\ \dots \\ 1 \end{bmatrix} = \sum_{j=1}^{n} x_\pi(j) = 1$$

$$x_\pi^T = \alpha \cdot x_\pi^T \cdot P + (1-\alpha) x_\pi^T \cdot e \cdot v = \alpha \cdot x_\pi^T \cdot P + (1-\alpha) x_\pi^T \cdot E \quad \text{on,} \quad E = e \cdot v^T$$

$$x_\pi^T = x_\pi^T [\alpha P + (1-\alpha) E] = x_\pi^T [P'] \quad \text{on,} \quad P' = \alpha P + (1-\alpha) E$$

Finalment, l'expressió resultant pot ser manipulada una altre cop per obtenir la expressió exacte que defineix l'autovector. La probabilitat estacionaria és l'autovector principal de la matriu de transicions i hi ha algoritmes per calcular-ho. Però a la literatura [3] es recomana executar iteracions, ja que, és menys costos i, tot i no obtenir el valor exacte, convergeix prou ràpid. En concret, l'algoritme iteratiu fixa una epsilon(ε) a partir de la qual ja no s'itera més.

$$x_\pi^T = x_\pi^T \{P'\} \leftrightarrow (x_\pi^T)^T = (x_\pi^T \{P'\})^T \leftrightarrow x_\pi = \{P'\}^T x_\pi$$

3.1.3 Aplicacions

A continuació, es mostren alguns exemples que utilitzen criteris basats en la similitud multimodal i la passejada aleatòria com a solució per a la reordenació.

- Hsu *et al.* [4] proposen la reordenació automàtica de vídeo, la qual classifica els resultats inicials de la cerca textual basant-se en els patrons contextuals més freqüents. Formulen el graf de similitud de forma que les histories dels vídeos són els nodes, les arestes es ponderen per similitud textual. Quan recuperen histories basades en una cerca textuals, els resultats són extesos a altres històries que no eren rellevants per la cera textual inicial a través del graf de similitud.

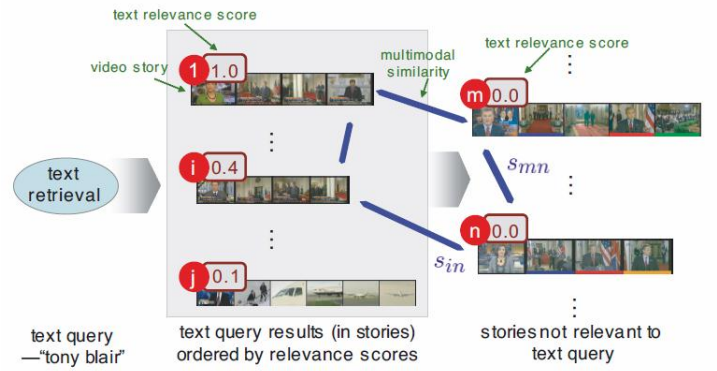


Figura 7. Exemple de cerca de vídeo que es beneficia de la similitud multimodal a nivell de histories en grans base de dades de vídeo.

- En canvi, Richter et al. [5] es centren en seleccionar les imatges més rellevants obtingudes d'una cerca textual en una base de dades comunitària com és la de Flickr, on les imatges sovint s'associen amb diferents metadades generades pels usuaris. Els usuaris d'aquestes comunitats són subjectius i dispars a l'hora d'introduir metadades fent que l'enfoc d'una cerca textual sigui difícil. La innovació que presenta és un filtrat dels graf de similitud per limitar aquesta influencia, finalment realitzen una passejada aleatòria en un graf de similitud multimodal basat en les etiquetes associades a la imatge i en la seva similitud visual.

La majoria de les solucions multimodals adopten una combinació lineal o probabilística. De forma que tracten els grafs de forma independent per després fusionar-los amb un pes associat a cada tipus de descriptor, aquest és el cas dels dos articles anteriors.

- Yao *et al.* [6] proposen una co-reordenació que té com a objectiu la interrelació entre els diferents models per reforçar la combinació visual i textual. Realitzen dues passejades aleatòries per reforçar el intercanvi mutu i la propagació de la rellevància de la informació a través dels dos grafs de similitud. Les probabilitats obtingudes en una modalitat serveixen per inicialitzar la passejada aleatòria en l'altre mode. L'article suggereix que les dues passejades aleatòries per reforçament mutu convergeixen en una mateixa probabilitat. Com a resultat, la reordenació textual i visual aprofiten els avantatges d'un del altre, aconseguint millors resultats.

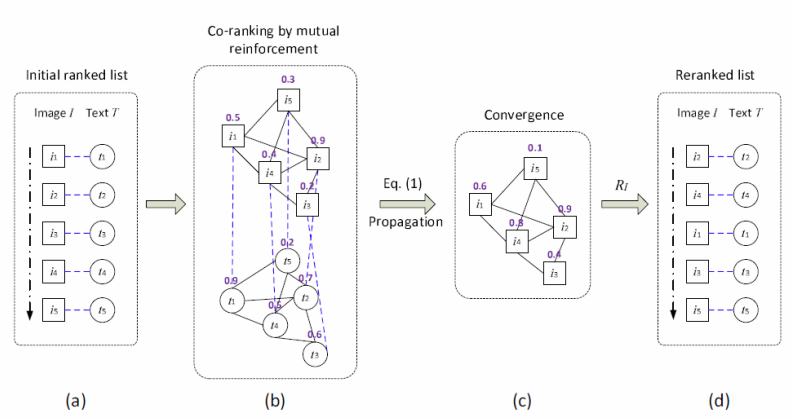


Figura 8. (a) llista de resultats inicials obtinguts de la cerca textual; (b) co-reranking per el reforç mutu dels dos grafs, visual i textual; (c) convergència després de la propagació en (b); (d) Llista reordenada.

3.2 Agrupament

L'agrupament és una estratègia per evitar que es repeteixin molts resultats que són molt similars o fins i tot duplicats. Les tècniques d'agrupament permeten formar grups amb els resultats obtinguts d'una cerca.

S'han estudiat tres tècniques diferents:

	Supervisat	Càlcul	Nous centroides
K-Means	✓	→	✓
Canopy	✗	↓	✗
Quality Threshold	✗	↑↑	✗

Taula 1. Comparació dels algoritmes d'agrupament

3.2.1 Supervisat

Els sistemes supervisats requereix un coneixement previ del número de grups (k) en que es vol dividir el conjunt de dades.

a) K-means

El K-means és un dels algoritmes d'agrupament més simples i coneguts. Tots els punts han de ser representant com un conjunt de característiques numèriques.

Cada punt es representa amb un vector en un espai n-dimensional on *n* és el número de totes les característiques utilitzades per descriure els punts.

La inicialització escull aleatòriament *k* punts que serveixen com centres inicials, centroides, dels grups. A continuació, tots els punts són assignats al centroide que tenen més a prop. Finalment, per a cada grup es determina un nou centroide calculant el promig dels vectors de característiques amb tots els punts assignats a ell i es torna a distribuir els punts als nous centroides més propers. Els processos d'assignació dels punts i recalcul del centroide es repeteix fins que els processos convergeixen o quan s'arriba a un número màxim predeterminat d'iteracions.

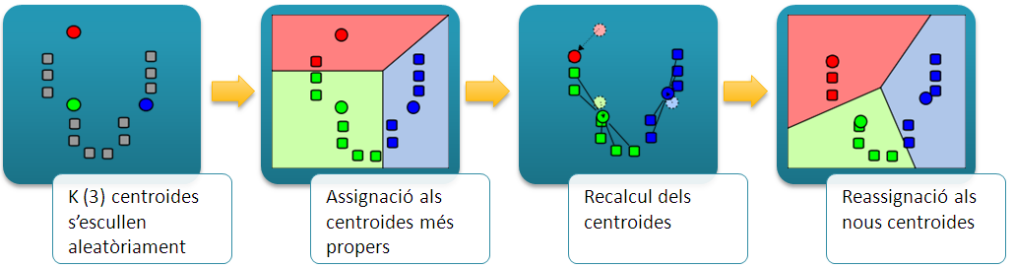


Figura 9. Esquema del algoritme K-means

3.2.2 No supervisat

Els sistemes no supervisats no requereixen el coneixement a priori del número de grups.

a) Canopy [14]

És un mètode molt simple, ràpid i precís que permet agrupar punts en grups. Normalment, s'utilitza com a un primer pas en tècniques d'agrupament més rigoroses com K-means o Mean-Shift, ja que, utilitza una aproximació de les distancies per dividir els punts en subconjunts anomenats canopies i, per tant,

L'algoritme utilitza dos paràmetres per agrupar els punts, aquests poden ser configurats pel usuari:

1. Un llindar aproximat de la mesura de similitud. (T_1)
2. Un llindar més exacte de la mesura de similitud. (T_2 , tal que $T_2 < T_1$)

Donat els dos llindars, els canopies es creen de la següent manera:

1. Es comença amb una llista dels punts en un ordre aleatori
2. S'agafa un punt i es mesura la seva distancia amb tots els altres punts.
3. Es posa tots el punts que estiguin dins del llindar T_1 en el mateix canopy i elimina de la llista tots els punts que es troben dins del llindar T_2 per no ser analitzats un altre vegada.
4. Es torna al punt 2 fins que la llista estigui buida.

A la ***Figura 10*** es mostren quatre grups: A, B, D-C, E. Els punts amb el mateix to de gris pertanyen al mateix grup. El punt A ha estat seleccionat al atzar i forma un canopy amb tots els punts que estan dins del llindar exterior (cercle sòlid). Els punts en el interior del llindar interior (cercle discontinu) estan exclosos de ser el centre, i de la formació de nous canopies. Els canopies D i C comparteixen el mateix grup i almenys tots els canopies estan dins d'un grup. Finalment la distancia més costosa a estat calculada només entre els punts del mateix canopy.

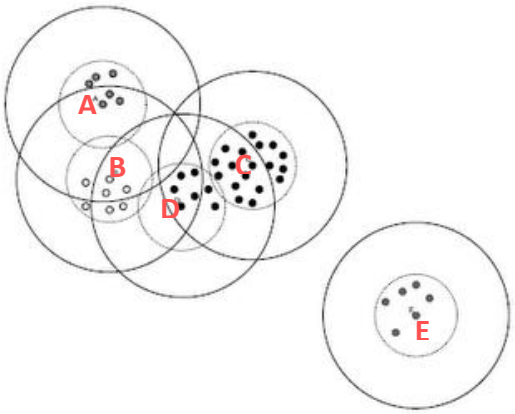


Figura 10. Resultat d'aplicar el Canopy a la primera etapa del algoritme d'agrupament.

Significativament, un punt pot aparèixer en més d'un canopy i, cada punt ha d'aparèixer almenys en un canopy. Els canopies es creen amb la intenció de que els punts que no apareixen en cap canopy comú estan prou allunyats per no estar en el mateix grup. Donat que la mesura de la distancia és aproximada no hi ha moltes garanties de que es compleixi aquesta propietat. Però al permetre que els canopies es sobreposin, per elecció d'un llindar gran, si que es pot garantir en alguns casos.

El seu cost computacional és baix, ja que, al començar amb una primera agrupació el número de mesures de distancia es redueix significativament degut a que s'ignoraren els punts que es troben fora dels canopies inicials.

b) Llindars de qualitat (Quality Treshold) [15]

Requereix més cost computacional que el k-means i sempre retorna el mateix resultats encara que s'executi moltes vegades.

Necessita dos paràmetres de configuració:

1. Màxima distancia entre el centre del grup i qualsevol dels punts assignats a ell.
2. Mínim número de punts que ha d'haver en un grup per a que es consideri como a tal.

L'algoritme segueix els següents passos:

1. Selecció dels dos paràmetres de configuració.
2. Creació dels grups candidats per a cada punt de manera iterativa incloent el punt més proper al grup, fins que el diàmetre de l'agrupació supera el llindar.
3. Creació del grup si el valor de nombre de punts supera el valor mínim i eliminar tots el punts del conjunt de dades.
4. Iteració del procés fins que no es pugui definir un altre grup que compleixi els requisits.

3.3 Mesures d'avaluació

En aquest apartat s'introdueixen les mesures d'avaluació de la rellevància i la diversitat, que s'han estudiat durant el treball. Aquestes mètriques permeten avaluar quines de les opcions estudiades ofereixen millors resultats per als sistemes de recuperació de la informació.

3.3.1 Precisió – Record

Les mesures precisió i record són dos indicadors àmpliament utilitzats per avaluar l'exactitud dels sistemes de recuperació d'informació. Es defineixen de la següent forma:

La **precisió** es defineix com el número de documents rellevants recuperats d'una cerca dividit pel número de documents totals recuperats.

$$Precisió(k) = \frac{número\ de\ docs\ rellevants\ fins\ K}{número\ de\ docs\ totals\ recueprats}$$

El **record** es defineix com el número de documents rellevants recuperats a les primeres K posicions dividit pel número de documents rellevants totals recuperats d'una cerca.

$$Recall\ (K) = \frac{número\ de\ docs\ rellevants\ fins\ K}{número\ de\ docs\ rellevants\ recuperats}$$

Amb aquestes dues mesures normalment es representen en un únic diagrama anomenat la corba precisió-record.

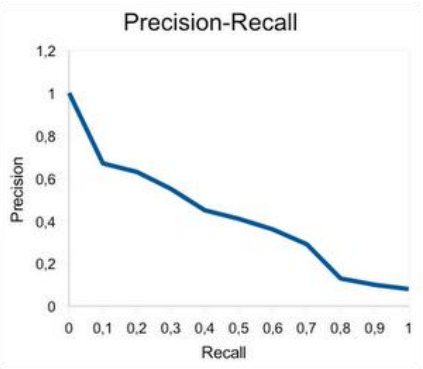


Figura 11. Corba precisió - record

3.3.2 Average Precision (AP)

L'***Average precision*** (AP) també s'utilitza per avaluar la rellevància dels resultats però ofereixen un únic valor que no està vinculat amb cap record concret. Aquesta mesura s'obté promitjant la precisió en els primers *m* valors obtinguts d'una llista resultant.

$$Average\ Precision\ (AP) = \frac{1}{m}\sum_{k=1}^{m} Precision(k)$$

on la Precision (k) és la proporció de *keyframes* rellevants considerant les primeres k posicions de la llista resultant, i la m és el número de *keyframes* rellevants.

3.3.3 Subtopic-recall (S-recall)

S-recall [8][9] s'utilitza per avaluar la diversitat de resultats en els sistemes de recuperació d'informació. Considerant un tema *T* amb *na* subtemes $A_1 \ldots A_{na}$ i una llista $d_1, \ldots, d_m$ de *m* documents. Es defineix l'S-recall fins K com el percentatge de subtemes coberts pels primers *k* documents.

$$S - recall\ at\ k = \frac{\left|\bigcup_{i=1}^{k} subtemes(d_i)\right|}{n_A}$$

on, subtemes(d_i) és el conjunt de subtemes pels quals el document d_i és rellevant.

3.4 GUIs de cercadors d'imatges

En aquest apartat es mostren alguns exemples de GUI que integren algoritmes d'agrupament per mostrar els resultats visuals.

- Google Image Swirl[6] [16]: és una eina de Google Labs. La seva funció es organitzar els resultats de les cerques per imatges en grups i subgrups, des del punt de vista de similitud visual i semàntica, i mostrar-los en una interfície de exploració. Els seus creadors suggereixen que aquest tipus de GUI pot ajudar a resoldre les consultes que generen varis tipus de resultats o que són ambigus. A la actualitat, aquesta funció només esta disponible per 200.000 consultes precalculades.

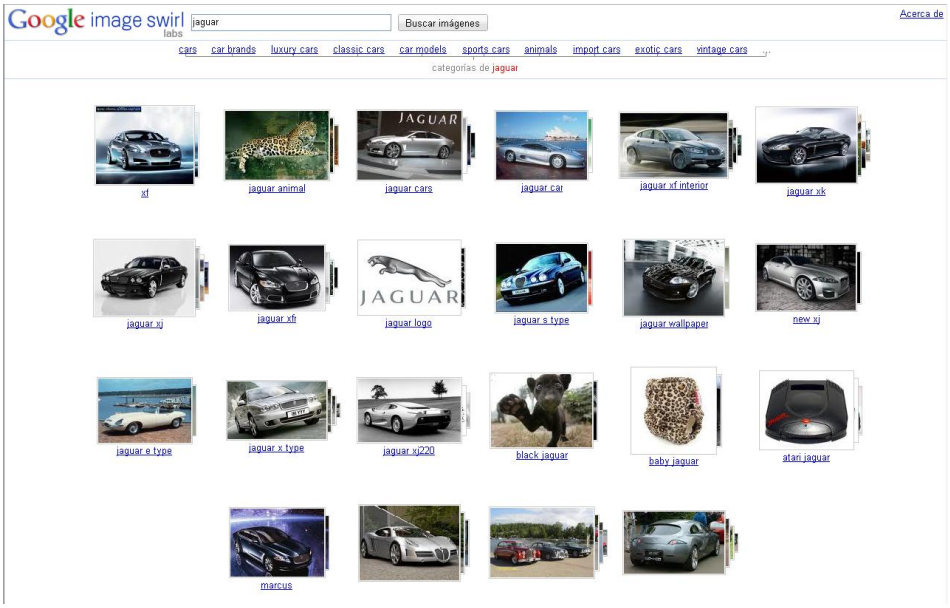


Figura 12. Google Image Swirl

[6] http://image-swirl.googlelabs.com/

Figura 13. Google Image Swirl: exploració d'un grup i dels seus subgrups

- Jing et al. [6] presenten dues propostes per integrar els algoritmes d'agrupament. Una és una visualització en grup, ***Figura 14***, que mostra quatre imatges representants per cada grup i el nom corresponent del grup (grup ID). L'altre vista és molt similar a la resta de visualització de resultats existents a la web, ***Figura 15***. Les dues propostes van ser avaluades per vuit participants i el resultat va ser que la primera vista era millor que la segona. A més van obtenir els comentaris dels participants i la principal raó per preferir la primera era que mostra més informació en menys espai.

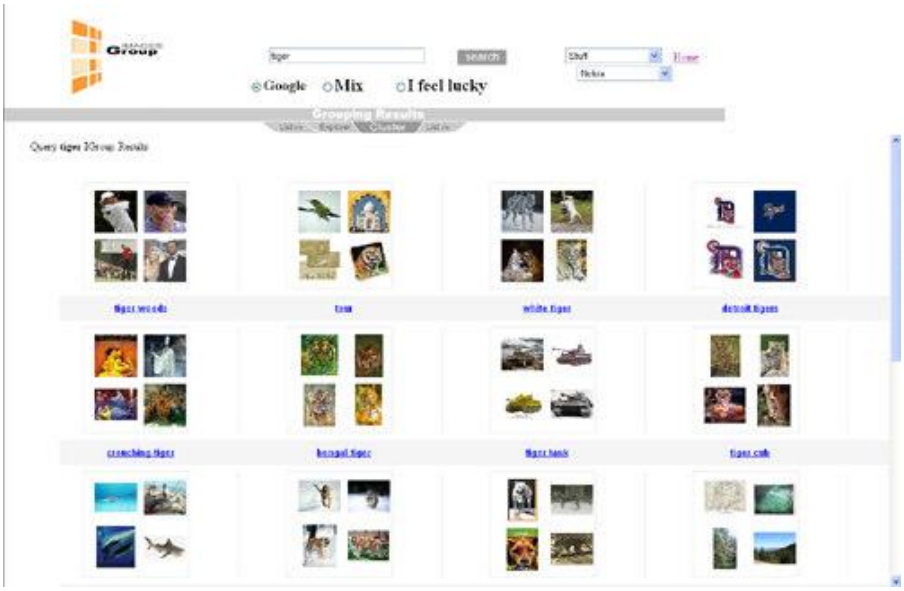


Figura 14. Visualització en grups

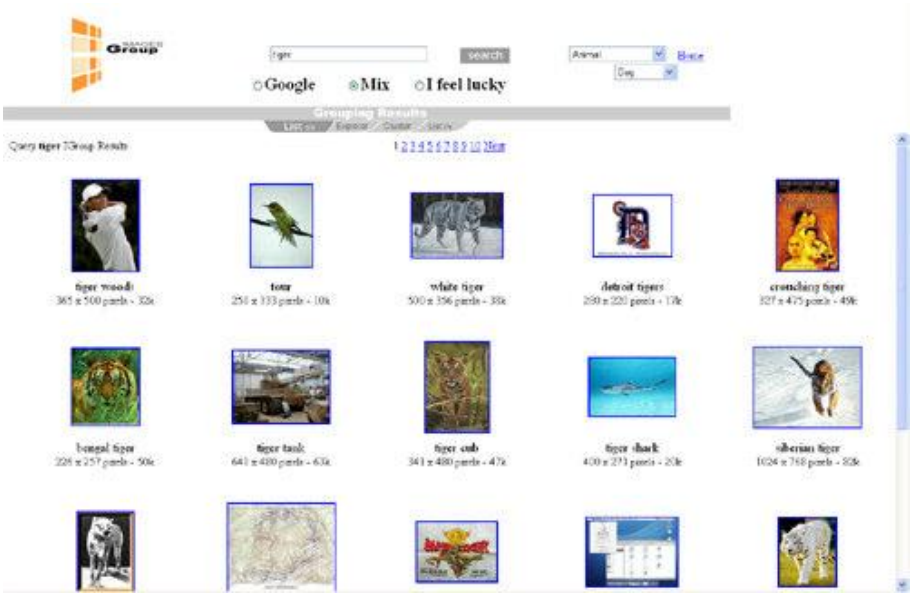


Figura 15. Visualització convencional

- IPhoto: és una aplicació desenvolupada per Apple. Pot importar, organitzar, editar, imprimir i compartir fotos digitals. Una vegada que els fotos són importades, opcionalment, poden ser etiquetades, marcades o organitzades en grups, coneguts com àlbums. Els àlbums es mostren com a la ***Figura 16***. A més, permet explorar les imatges de cada àlbum posicionant el ratolí sobre la imatge representant, des de la mateixa pantalla. La imatge que apareixerà depèn de la posició del ratolí sobre la imatge representant.

Figura 16. Visualització del IPhoto

4. Disseny

En aquest capítol s'exposen les tècniques proposades de reordenació i agrupament per tal de resoldre els problemes descrits a l'apartat de Requeriments. A més, també es presenten el disseny i funcionalitats de la GUI.

4.1 Reordenació

L'algoritme de reordenació té com a entrada una llista de *keyframes*, i com a sortida aquests mateixos resultats ordenats amb unes puntuacions segons el criteri de similitud visual. En la ***Figura 17*** es pot veure que esta format per, bàsicament, quatre blocs: construcció del graf de similitud, filtrat, passejada aleatòria i fusió de probabilitats.

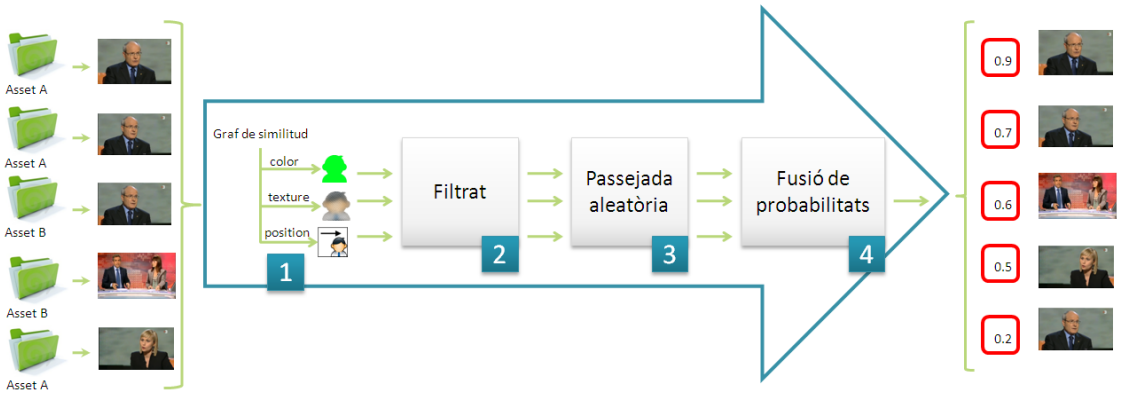


Figura 17. Esquema general del algoritme de reordenació

A continuació es descriuen en detall cadascun d'aquests blocs.

4.1.1 Càlcul dels grafs de similitud visual

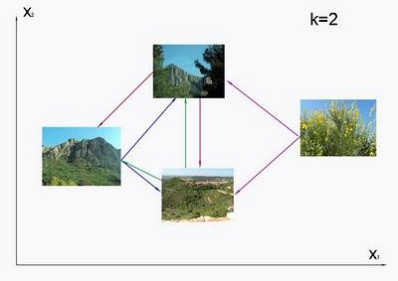
Un dels objectius principals d'aquest projecte és resoldre el problema de l'escala d'anotació manual d'un *asset* de vídeo, la qual es pot referir a un únic *keyframe* o bé a un grup de *keyframes* de l'*asset*. La suposició que es fa és que els *keyframes* rellevants per a la cerca apareixen en els resultats en major proporció que els no rellevants i en *assets* diferents. Per representar aquesta relació de similitud és calcula el graf de similitud.

Tot i que algoritmes revisats [4][5][6] es basen en sistemes multimodals, és a dir, parteixen de diferents tipus de grafs de similitud com el visual i el textual per després fer una combinació de tots ells, aquest projecte proposa combinar quatre tipus de similitud visuals. Aquestes similituds es mesuren amb la distancia visual entre els descriptors visuals associats a cada imatge.

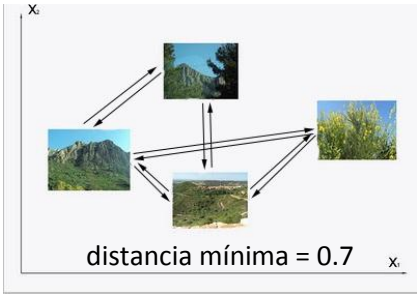
En concret es construeixen quatre grafs de similitud a partir de la llista de *keyframes* d'entrada, un per a cada tipus descriptor visual. Els descriptors visual que s'utilitzen en el nostre sistema es troben dins l'estàndard MPEG-7 [10] i aquests són: color layout, color dominant, color structure, texture edge histogram.

La construcció dels grafs de similitud es pot realitzar de diferents maneres. En aquest projecte s'han estudiat dues opcions:

- Limitant el número màxim d'arestes (k) que es permeten per a cada node. Segons Young Rui [1] el valor recomanable per aquesta variable és el 5% de la base de dades.



- Limitant la distancia visual mínima per a que hi hagi una aresta entre dos nodes. D'aquesta manera s'assegura que tots els nodes que més s'assemblen estan connectats, però en canvi no es controla el número de connexions per node.



Els grafs de similitud es poden utilitzar de dues formes diferents depenent de si es calcula sobre tota la base de dades o sobre els resultats de la cerca.

- **Independent de la consulta:** aquest mètode calcula els grafs de similitud amb tots els *keyframes* de la base de dades. D'aquest manera els grafs de similitud ja estan precalculats quan es realitza la reordenació.

- **Depenent de la consulta:** aquest mètode calcula els grafs de similitud només amb els *keyframes* obtinguts d'una consulta. Per tant, és calcula en el moment de fer la reordenació.

Cada mode pot donar resultats diferents en el cas de la construcció dels grafs de similitud limitant el número d'arestes per nodes, però donen el mateix resultat si es limita la distancia mínima. A la ***Figura 18*** es pot veure els resultats dels dos modes. A l'esquerra es mostra el graf del mode dependent de la consulta i a la dreta, primer es mostra el graf sencer, construït amb tots els *keyframes* de la base de dades, i després el resultat de fer el filtrat dels *keyframes* que responen a la consulta. Els *keyframes* resultants d'una consulta són els marcats en negre. Tal com s'observa, limitant el número d'arestes pels nodes (a) els grafs són diferents i, per tant, s'obtindran resultats diferents. En canvi, limitant la distancia mínima de similitud (b) s'obtenen grafs iguals, degut a que s'inclouen totes les arestes possibles.

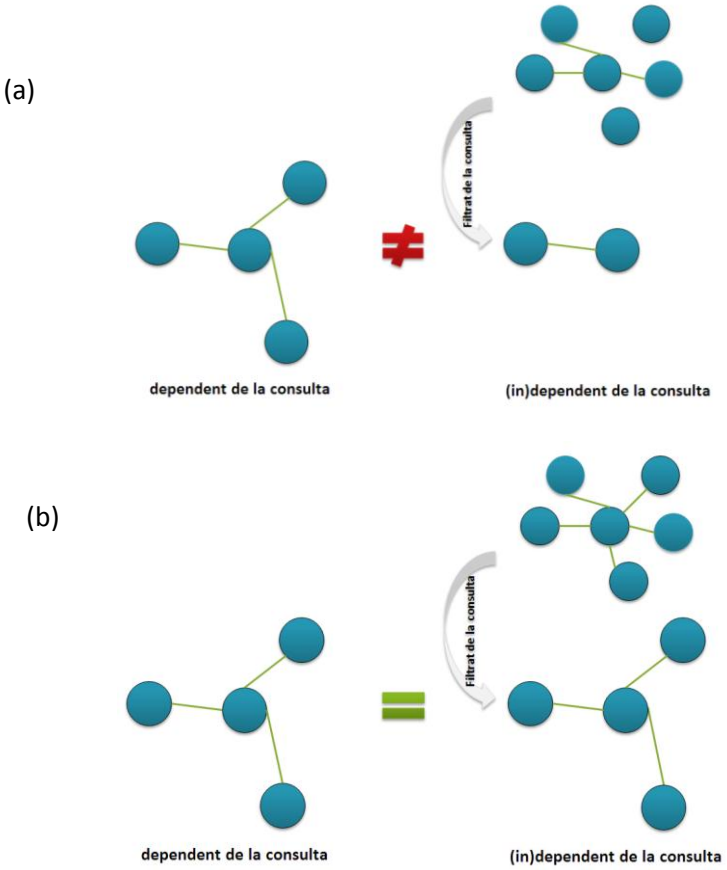


Figura 18. Comparació dels dos modes de funcionament segons la limitació del graf de similitud (a) Grafs de similitud limitats pel nombre de connexions permeses. (b) Graf de similitud limitats per la distancia mínima de semblança.

El càlcul dels grafs de similitud comporta un cost computacional molt gran ja que, requereix calcular la distància de cada imatge respecte cadascuna de la resta d'imatges de la base de dades. Per aquest motiu, aquest projecte es basa en grafs de similituds pre-calculats per a cada descriptor visual i amb tots els *keyframes* de la base de dades.

Donat que es vol obtenir els mateixos resultats per qualsevol mode i, donat que es treballa amb grafs pre-calculats, s'ha optat per la construcció dels grafs limitant la distancia mínima de similitud. Ara bé, els primers resultats experimentals que es van fer vam veure que aquesta solució pot generar massa veïns per a alguns nodes. Per això, s'ha pres com a decisió final una solució híbrida limitant la distancia visual però també el número de màxim d'arestes per no obtenir massa resultats.

Tot i que es va limitar el número màxim d'arestes a 1000 encara s'obtenien massa resultats per a que l'algoritme fos prou ràpid en entorns d'explotació. Però alhora, una limitació dràstica perjudicava els resultats obtinguts a la fase d'experimentació degut a que el número d'arestes per node no eren suficients per treballar en mode independent de la consulta. Per aquest motiu, s'ha reproduït dos grafs de similitud amb diferents limitacions en el número màxim d'arestes per node depenent en quin entorn s'utilitzi.

Solució final:

Experimentació

- Número màxim d'arestes per node: $N_{màx.} = 1000$
- Distancia visual mínima per cada descriptor visual:

Descriptor visual	Distancia mínima
color layout	0.7
color dominant	0.7
color structure	0.5
texture edge histogram	0.7

Explotació

- Número màxim d'arestes per node: $N_{màx.} = 50$
- Distancia visual mínima per cada descriptor visual:

Descriptor visual	Distancia mínima
color layout	0.7
color dominant	0.7
color structure	0.5
texture edge histogram	0.7

4.1.2 Filtrat

L'objectiu d'aquest bloc és eliminar arestes en els grafs de similitud per tal de garantir la diversitat dels *assets* sobre els resultats alhora que es preserven *keyframes* rellevants entre els millors resultats.

Per exemple, si es fa la consulta "President Montilla"a la base de dades es retornaran varis *assets*, i entre ells es podria obtenir aquests dos:

- *Asset* A: entrevista al President Montilla
- *Asset* B: el telenotícies on es comenta l'anterior entrevista.

- **Evitar la influencia dels *assets***

En el context de la CCMA, el contingut i la mida dels *assets* pot provocar efectes adversos en la reordenació. Per exemple, si en un *asset* hi ha molts *keyframes* similars però aquests no s'assemblen a *keyframes* en altres *assets*, vol dir, que aquests *keyframes* no són tan rellevants per a la consulta. El fet de que hi hagi molts *keyframes* similars fa que a l'hora de calcular la passejada aleatòria aquests es puntuïn entre ells i la seva probabilitat augmenti per efecte d'estar fortament connectats a altres *keyframes* del mateix *asset* quan en realitat la seva rellevància es deguda a la influencia que ha exercit el propi *asset*, i no altres *keyframes* de diferents *assets* com hauria de ser.

A la ***Figura 19***(a), es pot veure que el *keyframe* del pla general del plató només esta repetit moltes vegades en un sol *asset* i que surt com a millor resultat. Aquest *keyframes* no és

rellevant per a la consulta però degut a la influencia que han exercit la resta de *keyframes* similars dins del *asset* a fet que ho fos.

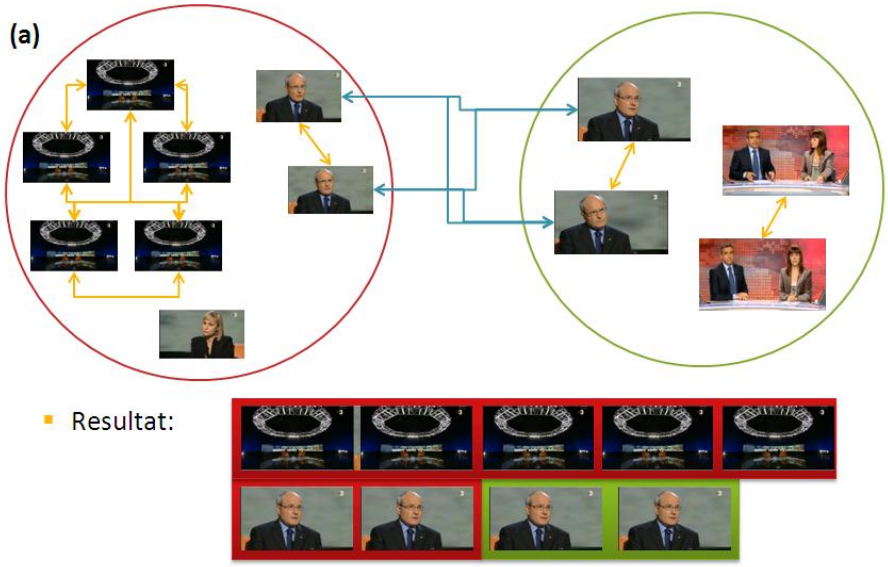


Figura 19(a). Graf de similitud visual sense cap restricció

- **Evitar *keyframes* similars com a millors resultats**

Normalment, quan es realitza una ordenació basant-se en el criteri visual per determinar la rellevància es tendeix a generar resultats molt semblants entre si. Aquest efecte es degut a que la importància d'un *keyframe* augmenta amb el nombre de *keyframes* que mostren continguts similars.

A la ***Figura 20***(b), es pot veure que passaria si no es realitzes aquest filtrat. Les imatges amb la cara del Montilla serien les millors perquè tenen més similitud amb la resta dels *keyframes*.

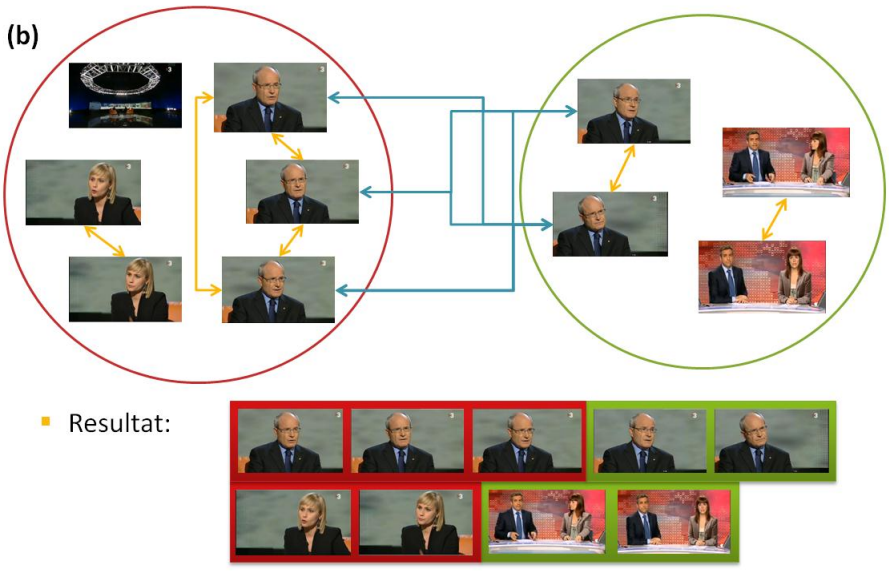


Figura 20 (b). Graf de similitud visual sense cap restricció

Recuperar molts *keyframes* similars dels mateix *asset* no aporta més informació de la que aportarà només una, ja que els usuaris busquen *assets* a través de *keyframes*. Aquesta redundància en els resultats, si no es realitza cap altre processat portarà a una situació de malbaratament d'àrea de la GUI i, en conseqüència, en una cerca menys eficient.

Per evitar la influencia de multiplicitat de *keyframes* similars dins d'un mateix *asset* s'han dissenyat dues restriccions en forma de filtrat d'arestes en el graf de similitud. Aquests filtrats permeten obtenir resultats més diversos mantenint els principis d'estimació de rellevància generada durant la passejada aleatòria.

1. **Filtrat intra-*asset***

Per tal d'evitar que *keyframes* d'un mateix *asset* siguin els responsables del valor de rellevància d'un *keyframe* s'eliminen les arestes entre *keyframes* del mateix *asset*.

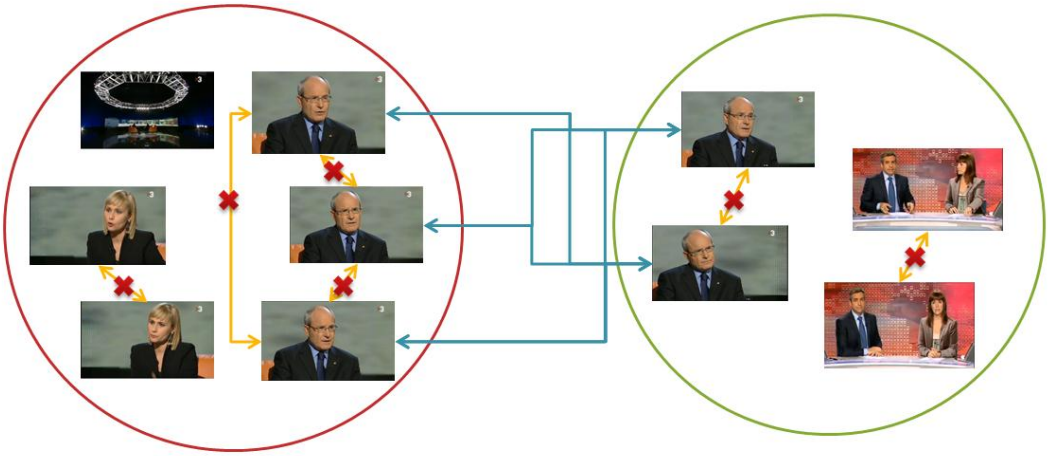


Figura 21. Graf de similitud amb la restricció intra-asset

2. **Filtrat inter-*asset***

Si un *keyframe* d'un *asset* té més d'una aresta d'entrada, és a dir, si rep vots per part de més d'un *keyframe* d'un *asset* en particular, els pesos d'aquestes arestes es normalitzen pel nombre d'arestes d'entrada provinents d'un mateix *asset*. Alternativament, es pot optar per escollir l'aresta de major pes i eliminar la resta d'arestes d'entrada.

Per exemple, en la ***Figura 22*** el *keyframe* marcat amb una rodona groga del *asset* B rep tres arestes d'entrada de tres *keyframes* que pertanyen al mateix *asset*, *asset* A. El filtrat inter-*asset* només permet que el *keyframe* marcat en groc rebi o les tres arestes normalitzades ***Figura 22*** o bé l'aresta amb el major pes ***Figura 23***.

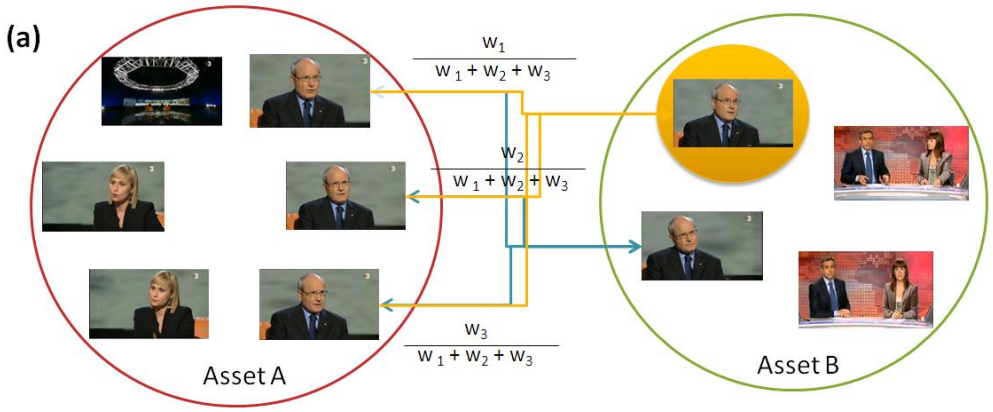


Figura 22. Graf de similitud amb la restricció inter-asset (normalització de les arestes)

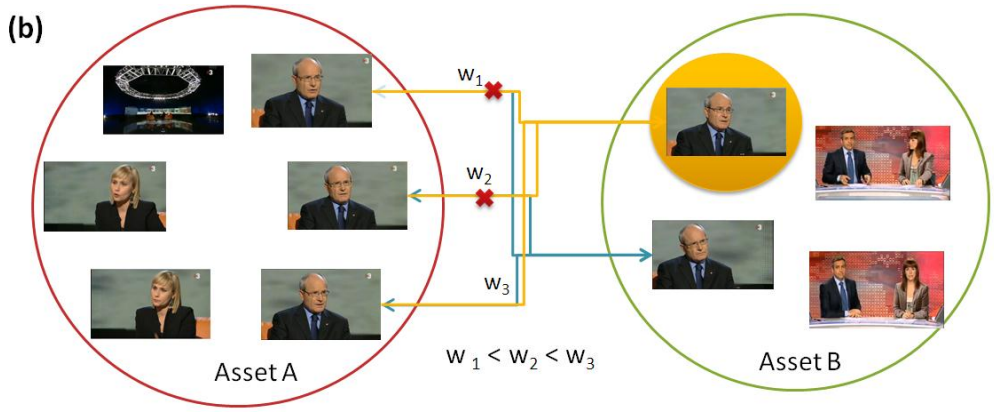


Figura 23. Graf de similitud amb la restricció inter-asset (aresta de major pes)

De les dues opcions possibles aquest projecte s'ha decantat per la implementació de la opció ***Figura 23***.

4.1.3 Passejada aleatòria

La passejada aleatòria assigna puntuacions als *keyframes* del graf de similitud visual basant-se en el número d'arestes i el seu pes visual.

En l'estat de l'art s'ha definit l'equació de la passejada aleatòria i com s'ha dit aquesta expressió introdueix la puntuació de la cerca prèvia v(j) obtinguda per una cerca prèvia.

Molt cercadors donen puntuacions inicials als resultats obtinguts segons la seva fiabilitat, d'aquesta manera s'ordenen els resultats moltes vegades. Per tant, si els resultats tenen puntuacions inicials el vector v(j) s'inicialitza amb aquestes. En canvi, en el cas en que no existeixin s'assignen unes puntuacions uniformes a tots el resultats.

Degut a que el sistemes de cerca de la CCMA utilitzats en aquest projecte no proporcionen aquesta informació, s'assignaran puntuacions uniformes per tots els resultats.

4.1.4 Fusió de resultats

Si s'utilitzen diferents descriptors visuals per construir els grafs de similitud i les passejades aleatòries cal combinar els resultats obtinguts per obtenir una puntuació fusionada. Existeixen diverses opcions com calcular el màxim, el mínim o la mitja. En aquest cas, s'ha seguit l'estratègia emprada a UPseek, que és la combinació lineal amb uns pesos normalitzats assignats a cada descriptor visual.

Molts sistemes realitzen la combinació dels grafs de similitud després de la seva construcció, i per tant, per fer aquesta fusió utilitzen les distancies visuals entre els nodes. El problema està en que el sistema de fusió de distancies normalitzada amb pesos és imperfecte degut a que la normalització és imperfecte. Per exemple, una distància 0.5 en textura no té perquè tenir el mateix significat semàntic que un 0.5 de color.

La formulació de la passejada aleatòria en forma de probabilitats permet evitar la fusió de distancies. Enlloc de primer calcular els diferents grafs de similitud i fusionar-los per distancies per després fer una passejada aleatòria, ***Figura 24***(a), s'ha optat per calcular tantes passejades aleatòries com descriptors visuals i, finalment, fusionar probabilitats, ***Figura 24***(b). Les probabilitats segur que estan ben normalitzades i així s'eviten els problemes de normalització imperfecte entre les distancies definides per cada descriptor.

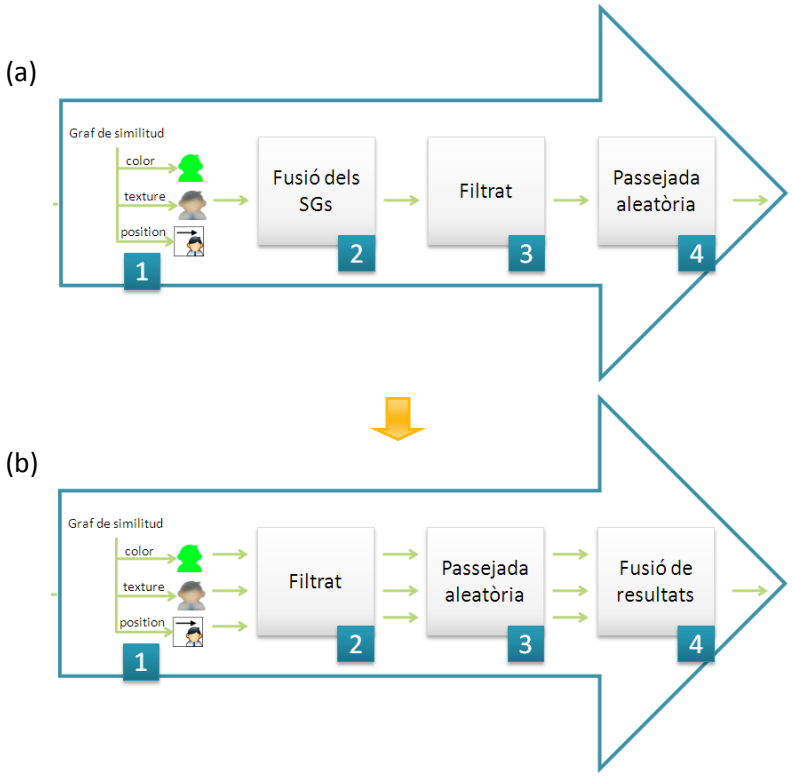


Figura 24. (a) Esquema de la fusió de distancies; (b) Esquema de la fusió de probabilitats.

4.2 Agrupament

L'objectiu del algoritme d'agrupament és mostrar un únic *keyframe* representant per a cada grup de *keyframes* similars d'un mateix *asset*.

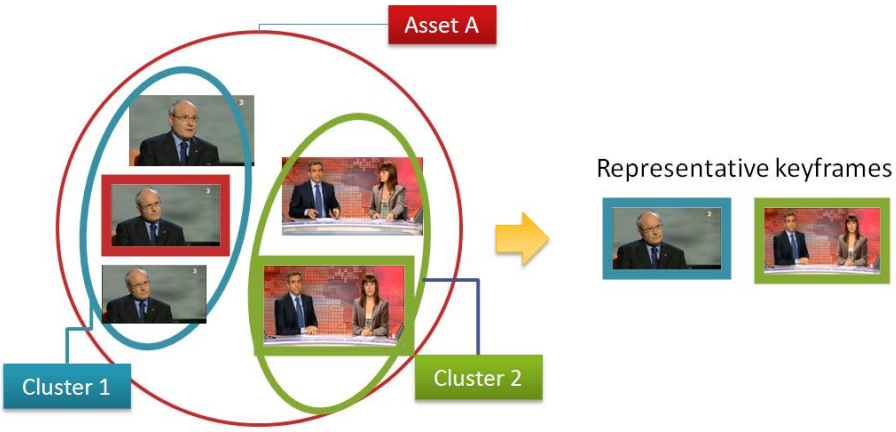


Figura 25. Keyframe representant per a cada grup similar.

Tot i que es realitzin el filtrat intra *asset* a l'algoritme de reordenació per eliminar les arestes dels *keyframes* del mateix *asset*, aquests poden quedar connectats indirectament a traves de les arestes inter *asset* i, per tant, no s'obtindrà un únic *keyframe* representant, com es pot veure a la ***Figura 26***.

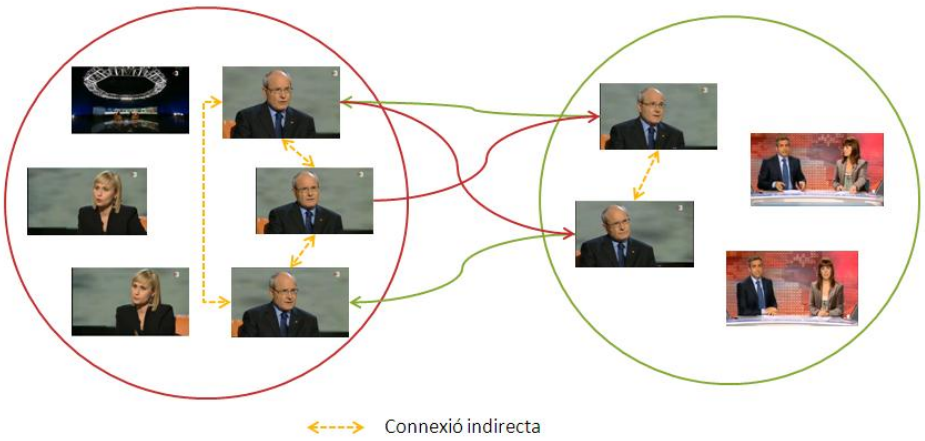


Figura 26. Connexió indirecta dels keyframes del mateix asset

En aquest projecte s'ha escollit un algoritme d'agrupació no supervisat perquè no es saben quants grups es volen generar entre els resultats.

S'ha escollit l'algoritme de Llindar de qualitat (QT) donada la disponibilitat de resultats pre-calculats en forma de grafs de similitud.

4.2.1 Llindar de qualitat (QT)

Finalment, es va decantar per l' utilització d'aquest algoritme per realitzar l'agrupament. El motiu és que el seu principal inconvenient, el cost computacional de totes les distancies, queda resolt perquè ja es compta les distàncies pre-calculades en els grafs de similitud.

L'esquema que segueix l'algoritme d'agrupament és el que es mostra a la ***Figura 27***.

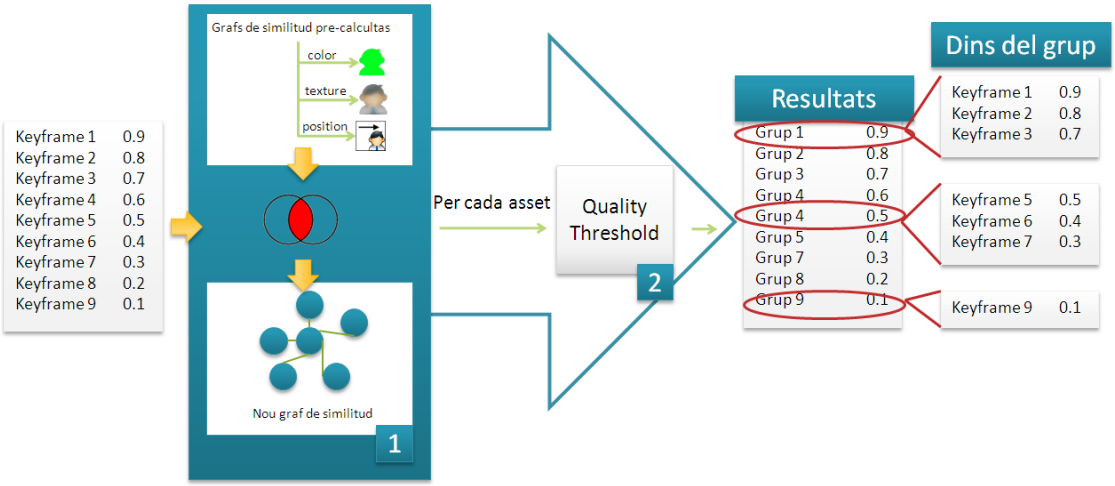


Figura 27. Esquema general del procés d'agrupament.

1. Grafs de similitud

Al igual que en l'algoritme de reordenació es parteix de grafs de similitud pre-calculats a partir de tots els *keyframes* de la base de dades per a cada descriptor visual.

Parteix d'una llista de *keyframes* que pot estar ordenada o no i a partir d'aquesta s'extrauen els nodes dels grafs pre-calculats per obtenir un graf de similitud només format amb *keyframes* que responen a una consulta, exactament igual al algoritme de reordenació. L'única diferencia és que si s'utilitzen més d'un descriptor visual la fusió d'aquests es realitza amb les distancies visuals.

És important tenir en compte que s'utilitzen dos tipus de puntuacions:

1. Les puntuacions de la llista de resultats inicials que representen com de fiables són aquest resultats respecte la consulta. Aquestes són les que s'utilitzen per donar les puntuacions als grups de resultats finals.
2. Les distancies visuals utilitzades en els grafs de similituds. Aquestes s'utilitzen per formar els diferents grups.

2. Quality Threshold

Hi han alguns paràmetres a tenir en compte:

1. El valor màxim del radi dels grups, el qual depèn de la similitud entre els *keyframes*.

2. El mínim número de resultats per poder ser considerat un grup.
3. El valor de la puntuació final del grup. En aquest cas, és el valor màxim referit a la puntuació del llistat inicial de tots els *keyframes* que formen el grup. El fet d'escollir aquest criteri es deu a que d'aquesta manera es conserva els resultats en el mateix ordre que en la llista inicial.

En aquest projecte s'ha dissenyat de forma que es pot escollir els dos valors crítics, mínim nombre de resultats i màxim radi dels grups.

3. Resultats

Com a resultat s'obté una llista amb el *keyframe* representant i la puntuació final del grup. El *keyframe* representant és el centroide que defineix el grup segons l'algoritme de QT. Per cada grup representant s'obté una llista amb tots els *keyframes* que formen el grup i la seva puntuació inicial.

4.3 Mesures d'avaluació

En aquest treball es proposa una nova expressió per calcular la diversitat d'*assets* en una llista ordenada de *keyframes*. Aquesta nova mesura esta basada en l'average precision, el qual dóna valors normalitzats entre 0.0 pel pitjor valor i 1.0 com el millor i que, a més, penalitza més la homogeneïtat en les primeres posicions que no en les últimes.

Es va partir de la mesura S-recall que es defineix com el "record de subtemes" i mesura el percentatge de subtemes apareguts en els primers k documents recuperats. En el nostre context però no és una bona solució perquè per a la cerca textual tots els *keyframes* recuperats ja estan vinculats a un, i només un, *asset* que és rellevant.

En realitat, en aquest projecte no estem tant interessats en el record sinó més aviat en la diversitat, ja que no es divideix pel número total d'*assets* diferents possibles sinó pel número màxim d'*assets* diferents que podrien estar a les primeres k posicions. Per aquest motiu, no es parlarà de recall sinó de diversity.

Diversitat (D) d'*assets*

L'*Asset*-Diversity mesura la diversitat d'*assets* en els primers *k* documents recuperats. L'AD és una mètrica que es comporta de manera similar a l'AP, és a dir, un valor normalitzat on el millor resultat és 1 i el pitjor és 0. A més, penalitza més la homogeneïtat en les primeres posicions que no pas en les últimes.

$$Asset - Diversity\ fins\ k = AD(k) = \frac{d(k) - 1}{k - 1}$$

on *d(k)* correspon al número d'*assets* diferent en les posicions 1...k de la llista resultants.

D'aquesta nova mesura neix l'Average *Asset*-Diversity (AAD) com la segona mètrica per avaluar l'algoritme de reordenació.

$$Average\ Asset - Diversity\ (AAD) = \frac{1}{m-1}\sum_{k=2}^{m} AD(k)$$

on, *m* és el número total d'*assets* diferents.

L'AAD satisfà la propietat de la mesura desitjada: 0 per resultats homogenis i 1 per una diversitat ideal. Si apareix certa homogeneïtat entre els resultats aquesta mesura produirà valors més baixos quan aquesta uniformitat es trobi en les primeres posicions de la llista resultant.

4.4 Interfície gràfica d'usuari GUI

En aquest projecte s'ha implementat un nou mòdul de *keyframes* que permet incorporar les tècniques de reordenació i agrupament sense deixar de banda la visualització actual dels *keyframes*.

S'han dissenyat tres perspectives diferents per visualitzar els resultats:

1. ***Keyframes* recuperats amb reordenació i agrupament**
   L'usuari farà una consulta textual i obtindrà una llista d'*assets* resultants que es mostraran tant textualment al mòdul de resultats, com fins ara, com al mòdul de *keyframes* de forma visual. Per tant, els *keyframes* es mostraran a la pestanya "Imatges" ordenats i agrupats, per *keyframes* similars d'un únic *asset*. Aquest cas només serà visible per un conjunt limitat d'*assets* resultants que pot ser configurable per l'usuari. En els resultats agrupats només es mostraran un número màxim de *keyframes* encara que n'hi hagi més dins el grup.

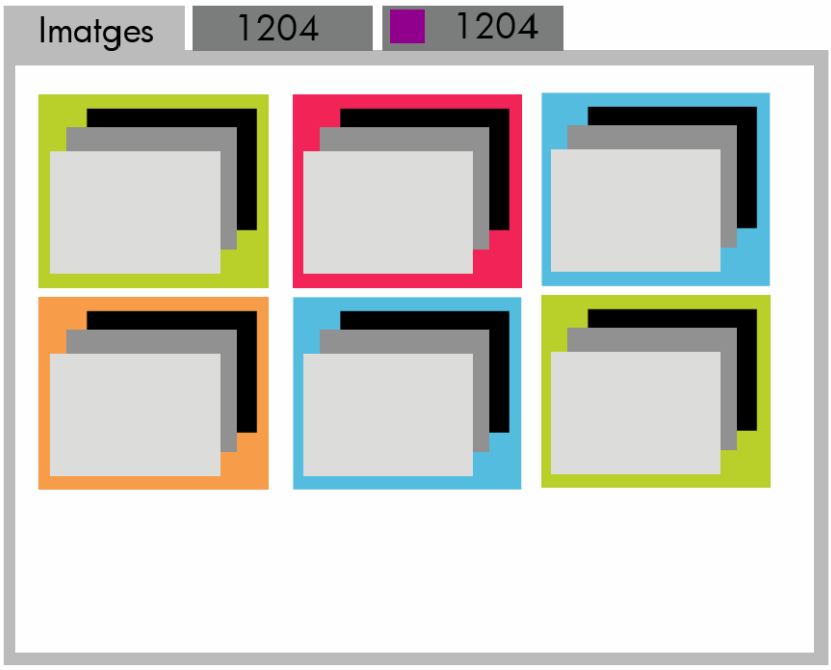


Figura 28. Perspectiva de tots els assets

2. ***Keyframes* d'un mateix *asset* ordenats temporalment**

   Perspectiva clàssica del mòdul de *keyframes*. Es mostren els *keyframes* en ordre temporal. Conté una barra amb botons que indiquen franges temporals, les usuaris poden navegar pel vídeo seleccionant la franja temporal que vulguin.

Figura 29. Perspectiva clàssica

3. ***Keyframes* d'un mateix *asset* amb reordenació i agrupament**

   Els *keyframes* es mostraran agrupats de la mateixa forma que la primera perspectiva.

Figura 30. Persepctiva dels keyframes agrupats d'un únic asset.

4.4.1 Funcionament

En el disseny s'ha utilitzat un sistema de pestanyes per obrir les diferents perspectives. La pestanya "Imatges" utilitza la primera perspectiva i només s'obre una vegada per consulta. En canvi, cada vegada que es desitgi visualitzar els *keyframes* d'un *asset* en concret del mòdul de resultats s'obrirà una nova pestanya amb la vista clàssica, aquestes pestanyes tindran com a títol l'*asset* ID i es podran tancar.

A més, des de la perspectiva d'ordenació temporal es tindrà l'opció de realitzar la reordenació i/o agrupament del *asset* que mostra. Aquesta funció està disponible a través d'un botó a la barra de menú. Els resultats agrupats i/o reordenats es mostraran en una altre pestanya amb la tercera visualització.

4.4.2 Exploració dels *keyframes* agrupats

Per poder explorar els resultats d'un grup s'ha de fer un clic amb el ratolí sobre el grup, aleshores, s'obrirà un panell amb tots els *keyframes* que pertanyen al grup.

Per tancar el panell desplegat es pot fer de dues formes:

1. Des del botó de tancar del propi panell.
2. Clicant al mateix grup que s'ha seleccionat.
3. Clicant a un altre grup. Aquesta acció el que farà serà tancar el panell desplegat i obrir un de nou amb els *keyframes* corresponents al grup seleccionat.

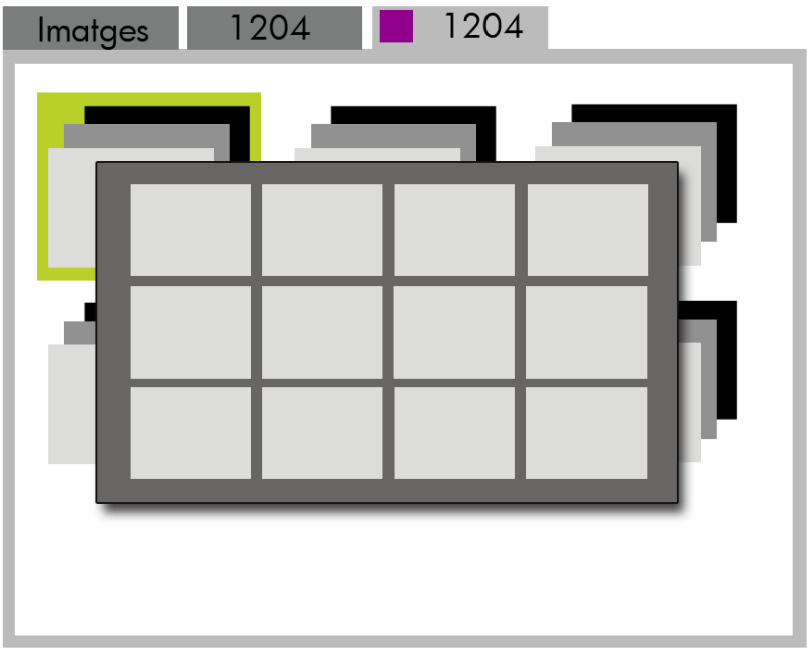


Figura 31. Exploració dels keyframes d'un grup mitjançant un panell desplegable.

4.4.3 Visualització dels *keyframes* mostrats d'un grup

Els *keyframes* agrupats es mostren un darrere el altre, fins a un màxim, i no permet fer-hi una ullada ràpida als que estan darrere. Per a aquest motiu quan el ratolí passi per sobre del *keyframes* del darrere aquests es mostraran al davant.

Figura 32. Vista ràpida dels keyframes d'un grup.

4.5 Comunicació

Els protocols de comunicació implementats han estat una extensió de la feina feta durant el PFC de la Pia Muñoz [11]. Com a millora, s'ha introduït una novetat en l'enviament i recepció de les dades entre els servidors de la CCMA i la UPC.

4.5.1 Arquitectura distribuïda

En els nous models organitzatius les empreses es divideixen en unitats independents i, cada vegada més, les aplicacions tendeixen a incorporar més mòduls al propi client, el qual demana una major autonomia i llibertat per accedir a la informació. Aquests requisits són difícilment realitzables en una arquitectura monolítica de maquinari i programari.

La solució recau en el desenvolupament d'una aplicació dividida en mòduls independents interconnectats entre ells mitjançant xarxes d'alta velocitat. Així, l'usuari de cada màquina pot accedir als recursos remots de la mateixa manera que hi accedeix als locals.

En aquest context, podem situar els dos servidors que alimenten l'aplicació. Cadascun d'ells té una funció clara i diferenciada:

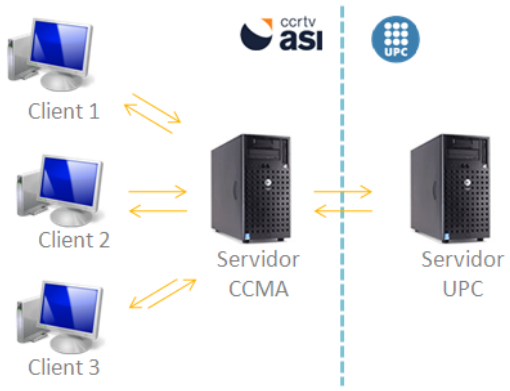


Figura 33. Estructura Client- Servidor

- El **servidor de la CCMA** s'encarrega de mediar entre el client web i la UPC així com de proporcionar a aquest client la informació necessària per poder mostrar els resultats de la cerca.
- El **servidor de la UPC** es comunica únicament amb el servidor de la CCMA, qui li fa arribar les peticions de cerca. La UPC processa aquestes peticions i retorna els resultats al servidor de la CCMA.

Tant la comunicació entre el client i el servidor dins de la pròpia Corporació com la comunicació entre el client de la CCMA i el servidor de la UPC, s'han servit del protocol HTTP per l'intercanvi de dades entre ells. A més, tots dos servidors segueixen una arquitectura del tipus REST servint-se del protocol HTTP entre d'altres.

4.5.2 Protocol HTTP

HTTP és el protocol utilitzat a les transaccions de la World Wild Web. És un protocol orientat a les transaccions entre client – servidor i segueix un esquema petició - resposta. A més, es tracte d'un protocol sense estat, és a dir, no guarda cap informació sobre les connexions anteriors.

En una comunicació HTTP el client, que en el nostre cas és un navegador web, efectua una petició i el servidor respon transmetent la informació demanda, que s'anomena **recurs**. Al recurs se'l identifica mitjançant una URL (Universal Resource Locator) i pot ser des d'un fitxer de text fins a qualsevol format multimèdia.

El tipus de dades que es manipulen en una comunicació HTTP segueix l'estàndard MIME (Multipurpose Internet Mail Extensions), que especifica com han de ser transferits els arxius multimèdia (video, àudio, imatges, etc.). MIME adjunta una capçalera amb informació del tipus d'arxius que s'està tramitant permetent, així, que tant el servidor com el client puguin entendre'l i llegir-lo.

La resposta a la petició d'un recurs s'anomena representació i manté la següent estructura:

1. Codi d'estat indicant si la petició és vàlida.
2. Codi d'error: els errors més típics que es poden trobar són la manca de permisos per accedir a un arxiu, la indisponibilitat de l'arxiu sol·licitat o bé la incorrecció en la petició realitzada. És a dir, una sintaxi o un nombre de paràmetres incorrecte.
3. La informació demanada a la crida.
4. Informació de l'objecte que 'ha retornat.

4.5.2.1 Sintaxi d'una URL per HTTP

En un servei web, la petició d'un recurs s'efectua seguint el model URL. La sintaxi típica d'una URL per HTTP és la següent:

esquema://servidor:port/ruta?paràmetre1=valor1&paràmetre2=valor2#enllaç

- **Esquema** o protocol que s'utilitza en la comunicació. En aquest cas, HTTP.
- **Servidor** o anfitrió és la part amb més pes d'una URL ja que proporciona el nom del domini o DNS i aquest factor no bé donat per defecte en cap navegador, com en el cas de l'esquema, ni en cap protocol, com en el cas del port.
- **Port**: Especifica el número de port TCP, que per defecte serà el 80. En el nostre cas utilitzem un port alternatiu: 8080.
- **Ruta**: Especificada pel servidor. Normalment s'utilitza per especifica la ruta d'un recurs.
- **Consulta**: la porció consulta és opcional i consta d'un o més paràmetres de cerca.
- **Enllaç**: Aquesta part es coneix com a identificador de fragment i es refereix aposicions a dins d'una mateixa pàgina.

4.5.2.2 Mètodes HTTP

A més, HTTP defineix vuit mètodes de petició: head, get, post, put, delete, trace, options i connect. Els més utilitzats són:

- **GET**: S'utilitza per demanar dades al servidor
- **POST**: S'utilitza per enviar dades cap al servidor
- **PUT**: S'utilitza per canviar informació ja existent o bé carregar un recurs
- **DELETE**: S'utilitza per borrar informació ja existent

El client dissenyat en aquest projecte ha requerit dels mètodes **GET** i **POST**. El mètode GET s'ha utilitzat per demanar qualsevol informació ja sigui un recurs del repositori de la Corporació o un resultat d'una cerca de la UPC amb el servidor de la CCMA com a interventor. El mètode POST s'utilitza per realitzar les crides entre el servidor de la CCMA i el de la UPC degut a que es necessari enviar molta informació per realitzar els processats.

4.5.3 Arquitectura Rest

El client web es comunica amb un servidor del tipus REST, el qual, al seu torn, fa ús del protocol HTTP per comunicar-se amb el client. El terme REST (Representational State Transfer) va ser introduït per primera vegada per Roy Fielding a la seva tesi doctoral sobre la web i originàriament feia referència a un conjunt de principis d'arquitectura. Si més no, en la actualitat, aquest terme s'utilitza en un sentit més ampli per a descriure qualsevol interfície web que utilitzi XML i HTTP.

REST no és un estàndard o protocol, sinó que fa referència a una arquitectura. Per aquest motiu, REST s'alimenta d'altres estàndards:

- HTTP
- URL
- Representació del recurs: XML, HTML, GIF, JPEG, etc.
- MIME Types: text (/XML, /HTML), imatge (/GIF, /JPEG), etc.

Algunes de les característiques fonamentals d'un servei web REST són les següents:

- Utilitzen un protocol client - servidor sense estat: El fet que els missatges HTTP continguin tota la informació necessària per fer la petició fa possible un sistema sense memòria i amb independència entre client i servidor. La separació entre aquests dos components redueix la complexitat de la comunicació, millora la efectivitat i augmenta la escalabilitat.

- Un dels punts forts d'aquest tipus de servei és la garantia d'escalabilitat del sistema, per tant, queda assegurada la capacitat del sistema per a canviar de grandària o configuració a fi d'adaptar-se a les circumstàncies canviants sense que això suposi una pèrdua de qualitat en els serveis oferts.

  A més, la manca de memòria fa possible un disseny més simple del servidor. En contraposició, trobem missatges més pesats i costosos ja que es reenvia tota la informació necessària per la cerca en cada petició.

- Sintaxi universal i unequívoca dels recursos: cada recurs té una única URL la qual és, al seu torn, la única informació necessària per accedir al recurs.

- Es poden afegir components intermediaris entre client i servidor com proxys, servidors dedicats o gateways per millorar la seguretat del sistema sense afectar en cap cas la comunicació entre client i servidor.

4.5.4 Model client – servidor dins la CCMA

La comunicació entre el client web i el servidor a dins de la Corporació es produeix entre un indeterminat nombre de clients i un únic servidor, el de la CCMA. Aquest es limita a l'intercanvi de dades del propi repositori de la CCMA i a les peticions d'execució d'un servei extern, provinent de la UPC, amb la consegüent rebuda dels resultats.

Totes les crides que fa el client al servidor són asíncrones, això vol dir, que el client no es queda bloquejat mentre espera la resposta. Els beneficis de fer aquest tipus de crides en contraposició de les transferències síncrones són:

- El client segueix estant receptiu a l'usuari. De forma que l'usuari pot realitzar altres coses mentre espera.
- Els motors JavaScript en els navegadors webs són, generalment, d'un únic subprocés de mode que si la crida al servidor es fes de forma síncrona faria que la pagina web es quedes "penjada" fins que finalitzes la crida. Si la xarxa és lenta o el servidor no respongués la pàgina no es desbloquejaria.
- Es poden fer múltiples crides al servidor al mateix temps.

4.5.5 Model client – servidor entre la CCMA i la UPC

La comunicació entre la CCMA i la UPC funciona de la mateixa manera, considerant la CCMA com a client dels serveis oferts per la UPC.

4.5.5.1 Serveis web dels algoritmes de reordenació i agrupament

En la comunicació entre els servidors de la CCMA i la UPC s'ha introduït una novetat en l'enviament i recepció de les dades. Prèviament, la UPC enviava representacions d'informació com a text pla i la CCMA parsejava la informació i creava les seves estructures de dades.

En aquest cas, la informació que s'ha d'enviar i rebre és molt gran per això es va decidir canviar a l'intercanvi d'objectes Java. Tant el client, el servidor de la CCMA, com el servidor de la UPC tenen una estructura comuna per l'enviament de les dades i per la recepció. El client envia l'objecte amb les dades, el servidor de la UPC rep l'objecte de la petició i el processa. Finalment, genera un nou objecte del mateix tipus per retorna els resultats al client.

Aquesta estructura és la GroupRankSO i els seus camps són:

Camp	Tipus	Definició
succes	boolean	Indica si s'ha realitzat correctament el processat
errorMessage	String	Indica el tipus d'error si hi ha hagut
mode	String	Mode d'execució
results	Vector<RetornUpseek>	Vector de *keyframes*
weights	Vector<Double>	Pesos dels descriptors visuals
rerank	boolean	Indica si es reordenaran els resultats
cluster	boolean	Indica si s'agruparan els resultats

Taula¿?. Estructura de GroupRankSO

L'objecte GroupRankSO conté un vector d'objectes RetornUpseek. Aquest conté tota la informació associada al *keyframe* i es defineix com:

Camp	Tipus	Definició
source	String	Identificador d'*asset*
title	String	Identificador del *keyframe*
score	double	Puntuació
cluster	int	Número d'objectes dins del grup

Taula ¿?. Estructura de RetornUpseek

Petició web

El client haurà d'indicar quin és el tipus de processat, els pesos dels descriptors visuals i quin és el tipus de mode d'execució que es vol utilitzar. Depenent del mode que s'esculli s'haurà d'indicar la informació necessària per completar la petició.

- **mode = *asset***: s'haurà d'afegir al vector ***results*** tants objectes RetornUpseek com número d'*assets* que es vulguin processar. Els objectes RetornUpseek han de tenir omplert el camp ***source*** amb l'identificador de l'*asset*.
- **mode = *keyframe***: s'haurà d'afegir al vector ***results*** tants objectes RetornUpseek com *keyframes* es vulguin processar. Els objectes RetornUpseek han de tenir omplert els camps ***title i source*** amb l'identificador del *keyframe*.
- **mode = score**: aquest mode funciona de la mateixa manera que l'anterior però a més al objecte RetornUpseek s'haurà d'indicar el camp ***score*** amb la puntuació de cada *keyframe*.

Resposta web

La resposta del procés retorna el mateix tipus d'objecte i serà la següent:

- boolean success: indica si s'ha realitzat correctament el processat.
- string errorMessage: indica el tipus d'error si hi ha hagut.

Hi ha dos modes possibles de resposta depenent de la configuració dels processats (reordenació i agrupament) que s'hagi escollit a la petició:

- **Només reordenació:** el vector ***results*** contindrà els *keyframes* reordenats. Dins de cada objecte RetornUpseek, el camp ***cluster*** estarà inicialitzat amb un 0.

- **Reordenació + Agrupament o només agrupament:** el vector ***results*** contindrà tots els *keyframes*. La forma de senyalitzar els grups serà la següent:

  El primer objecte RetornUpseek del vector ***results*** tindrà el camp ***cluster*** inicialitzat amb el número de resultats (X) que s'han agrupat, els altres camps (source, title i score) donaran la informació del centroide, *keyframe* representant del grup. Els següents X objectes del vector contindran els *keyframes* del grup i els seus camps ***cluster*** estaran inicialitzats amb un 0.

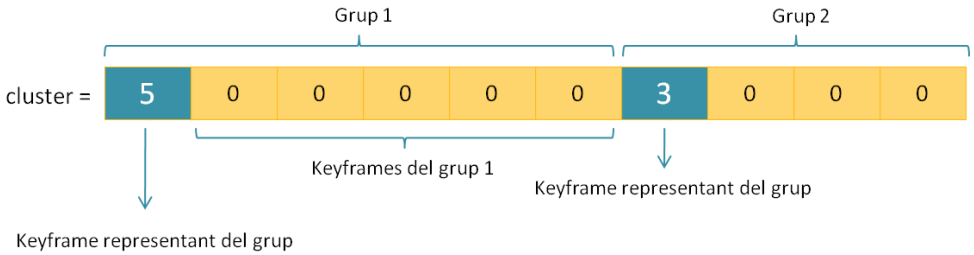


Figura 34. Senyalització dels grups en la resposta web

5. Desenvolupament

5.1 Entorn de desenvolupament

Aquest projecte s'emmarca en dos entorns diferents de treball, per una banda la UPC i per l'altre la CCMA. Per aquest motiu aquest projecte s'han utilitzat una gran quantitat d'eines. En aquest apartat es diferencien els dos entorns per explicar les eines utilitzades en cada cas.

5.1.1 A la UPC i CCMA

Java

Java [7] és un llenguatge de programació orientat a objectes, desenvolupat per Sun Microsystems i actualment impulsat per Oracle. La seva sintaxis deriva, en gran part, de C i C++, però té un model d'objectes més simple i elimina paràmetres de baix nivell, que poden induir a molts errors. També, és multiplataforma, és a dir, permet la execució d'un mateix programa en múltiples sistemes operatius.

Eclipse

Eclipse és un entorn integrat de desenvolupament multiplataforma de codi obert programada principalment en Java. Permet desenvolupar projectes en Java, C, C++, Python, PHP i, molts altres, sempre i quan s'instal·li els connectors corresponents per a cada llenguatge de programació.

L'eclipse va ser desenvolupat originalment per IBM però actualment es desenvolupat per la **Fundació Eclipse**[8], una organització independent que fomenta una comunitat de programari lliure i un conjunt de productes complementaris i serveis.

5.1.2 A la UPC

Control de versions

El **Subversion**[9], conegut com *svn*, és un programari lliure sota la llicencia Apache/BSD de sistemes de control de versions. És molt utilitzat en projectes en equip, on els diferents membres treballen en paral·lel sobre el mateix projecte. El sistema permet tenir emmagatzemada una versió compartida del codi del projecte que els diferents desenvolupadors van actualitzant amb les noves aportacions individuals. D'aquesta manera tot l'equip té accés directe a la darrera versió i pot prendre-la com a punt de partida. Una utilitat important és la detecció de conflictes entre les diferents versions, si dues persones modifiquen la mateixa part de codi sense tenir-se en compte el sistema avisarà incompatibilitats i incoherències quan s'intenti sincronitzar a través del Subversion.

El projecte UPSeek treballa amb aquest tipus de repositori mitjançant el connector Subclipse integrat a l'Eclipse. A més, el GPI també disposa d'uns scripts addicionals que ajuden

[7] http://www.java.com/es/
[8] http://www.eclipse.org/
[9] http://subversion.apache.org/

a compartir el codi. Durant el PFC, fina a set desenvolupadors diferents treballaven en paral·lel sobre la mateixa llibreria.

Cada membre té una branca per desenvolupar el seu codi que esta associada al tronc. El tronc conté la versió compartida del codi del projecte. Els desenvolupadors pugen al tronc les modificacions que han realitzat a les seves branques i els altres membres han d'actualitzar les seves branques per sincronitzar-se amb el tronc, per tal de tenir la mateixa versió del codi del projecte.

A continuació s'explica les operacions bàsiques que s'han utilitzat:

- Des de Subclipse

  commit: és l'acció d'escriure els nous canvis realitzats al codi al repositori.

  update: actualitza els codi amb les modificacions que s'han realitzat al repositori.

  edit conflicts: un conflicte es produeix quan diferents desenvolupadors realitzen canvis en el mateix document i, el sistema és incapaç de fusionar els canvis. Un desenvolupador ha de resoldre el conflicte manualment mitjançant l'operació *edit conflicts.*

- Des del terminal (scripts GPI):

  svn-rebase: actualitza la branca del desenvolupador amb els darreres canvis que s'han integrat al tronc.

  svn-deliver: actualitza el tronc amb les modificacions de la branca del desenvolupador.

Llibreries

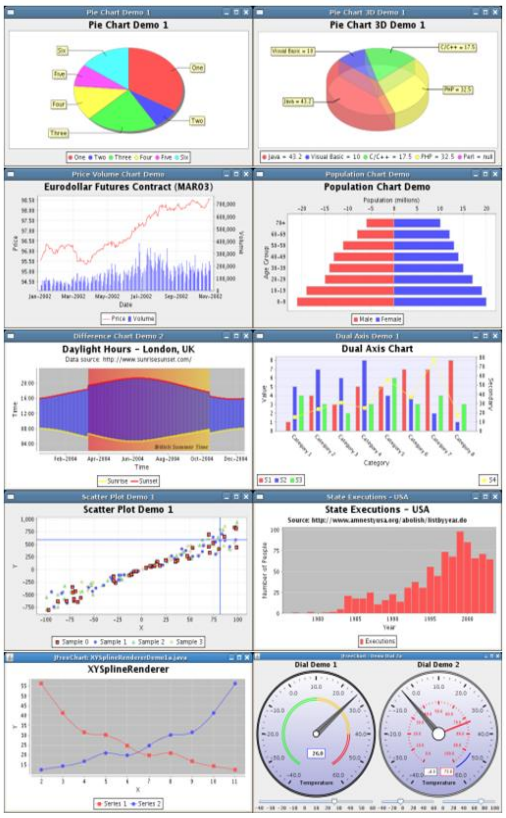
Java Matrix Package (Jama)[10]*:* és un paquet bàsic d'àlgebra lineal. Els seu propòsit es proporcionar unes classes a nivell d'usuari per a la construcció i manipulació de matrius. Aquesta llibreria s'ha utilitzat en el càlcul de la passejada aleatòria per a la manipulació de les probabilitats i la fusió d'aquestes. S'ha utilitzat per gestionar les probabilitats obtingudes mitjançant les passejades aleatòries.

JFreeChart[11]*:* és un entorn de treball de codi obert pel llenguatge de programació Java, que permet la creació de gràfiques complexes d'una forma senzilla. Es distribueix sota les condicions de la GNU Lesser General Public Licence (LGPL). S'ha utilitzat per automatitzar la generació de gràfiques amb els resultats de les avaluacions.

[10] http://math.nist.gov/javanumerics/jama/
[11] http://www.jfree.org/jfreechart/

Suporta els següents tipus de gràfiques:

- Gràfics X-Y
- Gràfics circulars
- Diagrames de Gantt
- Gràfics de barres
- Entre d'altres



JFreeChart dibuixa automàticament les escales dels eixos i les llegendes, tot i que, també permet configurar els eixos i les llegendes per a cada gràfica de forma manual. A més, es possible posar varis marcadors als gràfics.

Suporta diferents tipus de sortides com components Swing, imatges (PNG i JPEG), i formats d'arxius de gràfics vectorials (PDF, EPS i SVG).

5.1.3 A la CCMA

Google Web Toolkit (GWT)

Actualment, la creació d'aplicacions web resulta un procés costos i propens a errors. Els desenvolupadors poden passar-se el 90% del temps estudiants les peculiaritats dels deferents navegadors. Per altre banda, la creació, la reutilització i el manteniment de una gran quantitat de components AJAX i bases de codi JavaScript poden ser tasques complexes.

El **GWT**[12] és un *entorn de treball (framework)* que permet crear aplicacions *AJAX* utilitzant el llenguatge de programació Java. Aquestes classes que son compilades posteriorment pel GWT en codi *JavaScript* que funciona automàticament en els principals navegadors. Tot el codi esta disponible sota la llicencia lliure d'*Apache 2.0*.

El GWT s'integra a qualsevol entorn de desenvolupament (IDE) de Java. Com que en aquest cas s'ha utilitzat l'Eclipse, es va instal·lar el *connector de Google per Eclipse*[13]. Aquest connector permet desenvolupar aplicacions de GWT des del mateix entorn de l'Eclipse.

L'aplicació Showcase del GWT [14]ofereix una visió general i exemples de casos d'ús de les diferents funcions del GWT.

[12] http://code.google.com/intl/es/webtoolkit/
[13] http://code.google.com/intl/es/eclipse/
[14] http://gwt.google.com/samples/Showcase/Showcase.html#!CwCheckBox

- **Modes de funcionament**

Durant el desenvolupament de l'aplicació es poden veure immediatament els canvis realitzats al codi mitjançant el navegador en **mode allotjament** de GWT. No es necessari que es torni a compilar el codi JavaScript ni que s'implementi en el servidor. En el mode allotjament el codi s'executa a la màquina virtual de Java (JVM), per tant, totes les funcions, com la depuració pas a pas i punts d'interrupció del depurador de Java s'aplicaran al codi GWT.

Durant la implementació s'utilitza el **mode web**. GWT compila el codi Java en arxius JavaScript independents sense format i passen a executar-se com JavaScript i HTML al navegador. A més, les aplicacions GWT admeten automàticament els navegadors IE, Firefox, Mozilla, Safari i Opera sense necessitat d'utilitzar un format especial pel codi, ja que, GWT el converteix al format més adequat.

- **Arquitectura**

GWT conté els següents components:

- **Compilador GWT Java a JavaScript**: tradueix el codi desenvolupat en Java al llenguatge JavaScript. S'utilitza quan s'executa en mode web.
- **JRE emulació de llibreries:** contenen les biblioteques més importants de les classes Java.
- **GWT Web UI Class Library**: conté un conjunt d'elements d'interfície d'usuari que permeten la creació d'objectes com textos, caixes de text, imatges i botons.

- **Aplicació del estil**

El GWT proporciona mètodes directament relacionats amb l'estil dels documents HTML generats. L'estil és defineix utilitzant el llenguatge CSS (Cascading Style Sheets). D'aquesta manera cada component té un nom d'estil associat a una regla CSS.

SmartGWT

SmartGWT[15] és un entorn de treball basat en GWT que proporciona una col·lecció de components i suport per a la gestió de dades al servidor. L'aplicació Showcase d'SmartGWT[16] ofereix una visió general i exemples de cassos d'ús de les diferents funcions del SmartGWT.

JSON (JavaScript Object Notation)

El JSON és un format universal per l'intercanvi de dades, independent del llenguatge de dades. D'aquesta manera, es similar a XML. Es basa en la anotació literal dels objectes JavaScript. És un format més lleuger que XML i permet descarregar dades més rapidament.

[15] http://code.google.com/p/smartgwt/
[16] http://www.smartclient.com/smartgwt/showcase/#main

5.2 Estructura del codi

El desenvolupament de codi d'aquest projecte ha inclòs tant el motor de reordenació com la implementació d'un nou client web capaç d'explotar-lo. Tant el motor d'agrupament com les comunicacions webs entre UPC i CCMA han estat desenvolupades per altres membres de l'equip.

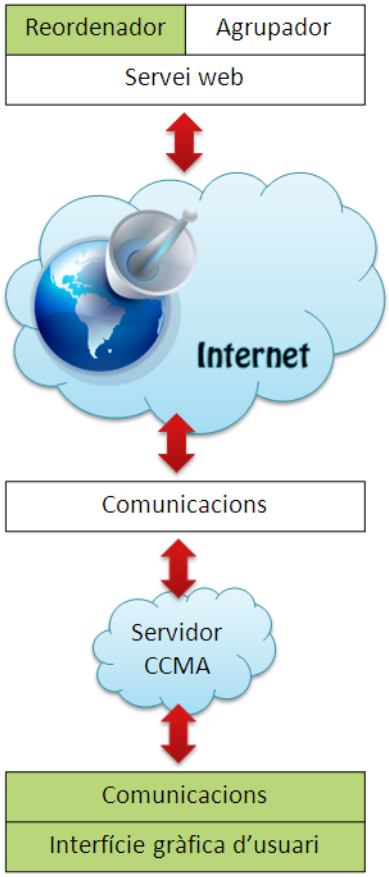


Figura 35. Estructura de les comunicacions

A continuació, s'explica les diferents classes Java que s'han implementat per crear el motor de reordenació i els sistemes d'avaluació de l'algoritme i l'estructura de classes de la GUI.

5.2.1 Motor de reordenació

Reranker.java

Es parteix d'una classe envoltori anomenada Reranker.class. Aquesta és l'encarregada de gestionar les crides a les diferents classes en l'ordre adequat.

La classe Reranker rep com a entrada una llista amb tots els *keyframes* que es volen reordenar i un conjunt de paràmetres de configuració, tal com mostra la ***Figura 36***. Aquests paràmetres s'utilitzen per avaluar l'algoritme i trobar la combinació més optima. Per a l'explotació del servei web aquestes variables estan fixades. Per tant, aquests paràmetres d'entrada només es permeten en l'entorn de desenvolupament.

Figura 36. Esquema de classes de l'algoritme de reordenació

El procés de reordenació és el següent:

1. Realitza el **truncat del graf de similitud pre-calculat** en cas de que no s'hi hagi passat com a paràmetre d'entrada. El truncat del graf només selecciona els nodes que es troben a la llista d'entrada de *keyframes*, per treballar només amb els *keyframes* rellevants a al consulta d'entre tots els resultats pre-calculats de similitud visual.
2. Per cada descriptor visual es realitza el **filtrat del graf de similitud** i la **passejada aleatòria**. Els resultats de la passejada aleatòria es van fusionant a mida que es calculen.

A la ***Taula 2*** es mostra les funcionalitats dels mètodes i classes que gestiona el Reranker.class

	Tipus	Entrada	Sortida	Funció
Truncated (SimilarityGraphsVds.class)	Mètode	RankedList	SimilarityGraphsVds	Seleccionar els nodes corresponents a la llista d'entrada en el grafs de similitud visuals precalculats
Filter.class	Classe	SimilarityGraph Boolean: intra i inter	SimilarityGraph	Filtrat intra i/o inter asset per un graf de similitud
RandomWalker.class	Classe	SimilarityGraphsVds	RankedList	Realitza la passejada aleatòria
Fuser	Mètode			Realitza la fusió dels diferents grafs de similitud visual.

Taula 2. Resum de les classes i mètodes que utilitza el Reranker.class

SimilarityGraphVds.java

Abans de fer aquest projecte ja existia la classe SimilarityGraph, però no era prou flexible. Per això es van crear i generalitzar les classes SimilarityGraph i SimilarityGraphsVds per a qualsevol cas.

Un objecte SimilarityGraphsVds conté un vector d'objectes SimilarityGraph on cada SimilarityGraph correspon a un graf de similitud visual per a un descriptor visual determinat.

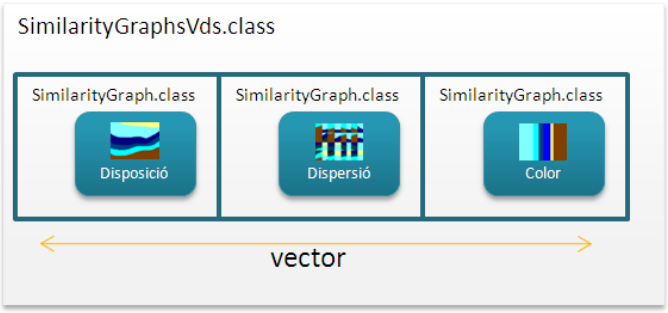


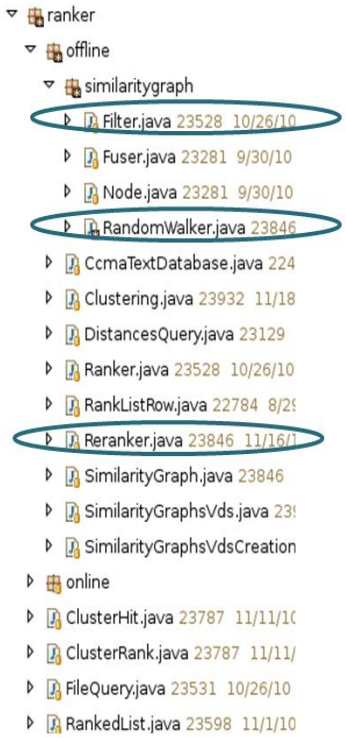
Figura 37. Estructura de les classes SimilarityGraphsVds i SimilarityGraph[17]

Aquesta classe conté el mètode **truncated** que permet seleccionar els nodes corresponents a la llista d'entrada en el grafs de similitud pre-calculats.

L'algoritme de reordenació pot gestionar el truncat dels grafs però per permetre la integració amb un servei web que combina la reordenació amb el motor d'agrupament, aquest truncat el pot realitza una classe externa. Així doncs, la classe Reranker també permet rebre un objecte amb el graf de similitud ja truncat.

Filter.class

Segons el mode de filtrat que s'hagi especificat s'executaran els mètodes intra i inter.



- **Mètode intra:** per a cada node del graf de similitud d'entrada elimina els veïns que pertanyen al mateix *asset*.
- **Mètode inter:** per a cada node del graf de similitud d'entrada repassa la llista dels seus veïns deixant només un *keyframe* per *asset* diferents. S'assegura que s'escull el *keyframe* de major similitud, ja que, aquesta llista esta ordenada de més a menys segons la seva similitud visual amb el node.

En cas en que s'executin els dos modes de filtrat, primer es portarà a terme el filtrat intra i després el inter, ja que, el mode intra elimina moltes més arestes que el inter i, per tant, serà més eficient.

RandomWalker.class

Per executar un objecte del tipus RandomWalker només és necessari cridar al seu constructor, que crea i inicialitza el objecte, i a continuació al mètode *run* que executa l'algoritme de la passejada aleatòria.

El seu constructor rep dos doubles amb el factor epsilon i l'alpha, i un vector amb les puntuacions inicials. En aquest projecte no es contempla l'ús de les puntuacions inicials al

[17] Icones de descriptors visual extretes del [100]

càlcul de la passejada aleatòria perquè els cercadors per text no les faciliten, tot i així aquesta classe esta preparada per poder fer-ho en un futur.

Al mètode run, se li introdueix un objecte SimilarityGraph associat a un descriptor visual i:

1. Es crea i s'inicialitza la probabilitat de transició.
2. Es calcula la passejada aleatòria pel objecte SimilarityGraph per tantes iteracions com:

$$Núm.iteracions = N \cdot \log_{10} T$$

on,

N: factor d'epsilon

T: número de nodes en el graf de similitud

En resultats experimentals es va observar que el llindar havia d'estar relacionat amb el nombre de nodes de cada graf perquè, per un llindar fixa, un graf amb pocs nodes convergia més que per un amb molts nodes. Com a conseqüència, pels grafs amb molt nodes no s'iterava el prou per determinar quins d'ells eren rellevants o no.

3. Es fusionen les probabilitats obtingudes ponderades amb el pes del seu descriptor visual associat.

5.2.2 Sistemes d'avaluació

Per avaluar l'algoritme de reordenació i trobar els valors òptims dels paràmetres de configuració s'ha implementat cinc noves classes pel sistema UPSeek: una que conté els experiments que s'han realitzat i, les altres quatre implementen les mesures que s'han utilitzat per avaluar l'algoritme.

CcmaReranker.class

Conté totes les combinacions dels paràmetres de configuració per realitzar els experiments. Aquesta classe només crida a la classe Reranker amb unes variables d'entrada determinades.

MAP.class i AP.class

Les classes MAP i AP s'utilitzen per calcular el Mean Average Precision i Average Precision, respectivament. La seva finalitat és calcular la rellevància dels *keyframes* d'una llista.

MAAD.class i AAD.class

Les classes MAAD i AAD s'utilitzen per calcular el Mean Average A-Diversity i Average-Diversity, respectivament. La seva finalitat és calcular al diversitat d'*assets* en una llista de *keyframes*.

5.2.3 Interfície gràfica d'usuari

5.2.3.1 Estructura del client web

Tot el mòdul de *keyframes* esta implementat dins d'un contenidor anomenat ModulKey. Aquesta classe conté un objecte TabSet definit per smartGWT per tenir el sistema de pestanyes. A més, s'encarrega d'inicialitzar les pestanyes segons la informació que es vulgui mostrar a la pantalla. En concret, s'ha implementat tres classes diferents que corresponen a tres vistes diferents:

1. **PlanaClassica**: aquesta classe crea la visualització clàssica del mòdul de *keyframes* amb els botons dels timecode dels *keyframes*.
2. **PlanaNoTimeCode**: aquesta classe crea una visualització d'imatges convencional, com la de molts cercadors d'imatges. Esta enfocada per l'ús del algoritme de reordenació o d'altres on la ordenació per temps deixa d'estar present.
3. **PlanaGroup**: aquesta classe crea una vista en grups dels *keyframes*, ja que, esta pensada per utilitzar-la amb l'algoritme d'agrupament.

A diferencia de les dues primeres planes, que carreguen les imatges directament per mostrar els resultats a pantalla, la PlanaGroup requereix un processat previ dels *keyframes* per mostrar-los. La representació visual dels *keyframes* agrupats es defineix a la classe TileKey.

La classe TileKey és un contenidor que té dues classes internes associades (nested classes). Aquestes són ImgCluster i ImgSerie les quals corresponen a la vista en grup dels *keyframes* i al panell que s'obre per visualitzar el grup sencer, respectivament. Els objectes TileKey que es carreguen a la PlanaGroup s'inicialitzen com objectes ImgCluster per defecte. Quan l'usuari clica sobre el contenidor es carrega l'objecte ImgSerie amb la informació associada al contenidor.

Alguns objectes incorporen algunes funcionalitats per a que l'usuari pugui interactuar amb ells a través de *handlers*. Aquest és el cas dels objectes TileKey, ImgCluster, el botó per demanar els serveis a la UPC i les pestanyes.

- **TileKey**: conté un *handler* del tipus *addClickHandler(ClickHandler*) per gestionar l'obertura i el tancament dels objectes *ImgSerie*, degut a que només es permet tenir un objecte ImgSerie desplegat.
- **ImgCluster**: totes les imatges que el formen tenen un *handler* del tipus *addMouseOverHandler(MouseOverHandler)* per tal de que la imatge aparegui davant de la resta del grup quan el ratolí esta sobre d'ella i torni cap enrere quan el ratolí no estigui a sobre.
- **Botó UPC**: té un *handler* del tipus *addClickHandler(ClickHandler)* que obre una finestra per configurar els serveis que es volen demanar a la UPC. Una vegada configurat realitza la crida.
- **Pestanyes del objecte TabSet:** s'ha incorporat un menú a les pestanyes que s'obre mitjançant el botó dret del ratolí. En el menú es té l'opció de tancar la pestanya actual o bé totes les altres.

5.2.3.2 Model client – servidor de la CCMA

El GWT proporciona diferents maneres per comunicar-se amb el servidor. L'estratègia que s'utilitzi dependrà del servidor amb el qual el client s'hagi de comunicar.

L'estratègia que s'ha emprat pel client web desenvolupat és recuperar dades JSON a través de HTTP. El GWT proporciona classes genèriques HTTP que es poden utilitzar per construir les peticions i, classes JSON i XML per processar la resposta. A més, es poden utilitzar els "overlay types" per convertir els objectes JavaScript a objectes Java per ser manipulats pel client.

Per realitzar una petició al servidor s'ha de crear una instancia del objecte *RequestBuilder* que facilita el mòdul de GWT. A aquest objecte s'ha d'especificar el mètode HTTP que és vol fer (GET, POST, ...) i la URL del recurs. Per realitzar la crida s'utilitza el mètode *sendRequest(String, RequestCallback)*. El paràmetre RequestCallback que se li passa s'encarregarà de rebre la resposta a través del seu mètode *onResponseRecieved(Request, Response)*, el qual es crida si la crida HTTP ha finalitzat correctament. Pel contrari, si al crida falla, per exemple, perquè el servidor no respon, es crida al mètode *onError(Request, Throwable)*.

```
final String url = "http://127.0.0.1:8888/proxy/digiproxy?server=192.168.163.17:8080/" +
"rest/getKeyframes&assetid="+searchText+"&tractament=none&filtrats=N";

RequestBuilder req = new RequestBuilder(RequestBuilder.GET, URL.encode(url));
try {
        Request request = req.sendRequest(null, new RequestCallback(){
            @Override
            public void onResponseReceived(Request request, Response response) {
                if (200 == response.getStatusCode()) {
                    //Resposta rebuda correctament
                } else {
                    //No s'ha pogut rebre la resposta correctament
                }
            }
            @Override
            public void onError(Request request, Throwable exception) {
                    //No s'ha pogut rebre la resposta correctament
            }
        });

} catch (RequestException e) {
        e.printStackTrace();
}
```

Manipulació de les dades amb JSON

El client rep dades en format JSON des del servidor. Les dades tenen la següent forma:


```
{  "error":null,
    "valid":true,
    "obj":[
                { "timeCode":"00:00:00:00",
                  "descriptionText":[],
                  "assetId":2704,
                  "imageUrl":"http://SSFOTOS/KEYFRAMES/2008/8/12/6398998/00_00_00_00.JPEG",
                  "videoUrl":"",
                  "descriptonColor":null,
                  "score":"",
                  "imageString":null,
                  "bpt_pos":null,
                  "bpt_neg":null,
                  "imageWeb":"",
                  "image":null
                },

                { "timeCode":"00:00:00:12", "descriptionText":[],
                  "assetId":2704,
                  "imageUrl":"http://SSFOTOS/KEYFRAMES/2008/8/12/6398998/00_00_00_12.JPEG",
                  "videoUrl":"",
                  "descriptonColor":null,
                  "score":"",
                  "imageString":null,
                  "bpt_pos":null,"bpt_neg":null,
                  "imageWeb":"",
                  "image":null
                }
           ]
}
```


A continuació, es necessari transformar el text JSON a objectes JavaScript. La forma més senzilla és utilitzant la funció eval() que incorpora JavaScript, la qual pot analitzar correctament el text JSON i generar l'objecte corresponent.

```
public static final native ResultsJSON GetResultJSON(String json) /*-{
                    return eval('(' + json + ')');
}-*/;
```

Però no només es vol accedir als objectes JSON, a més, es vol treballar amb ells. El GWT permet utilitzar els "overlay types" per transformar aquests objectes a objectes Java i, així el client podrà manipular aquesta informació. A continuació es mostra un exemple d'un overlay type implementat al client web:

```
public class ResultsJSON extends JavaScriptObject{

        protected ResultsJSON(){
        }

        public final native String getError()/*-{
                return this.error;
        }-*/;

        public final native boolean getValid()/*-{
                return this.valid;
        }-*/;

        public final native JsArray<ClipInfo> getObj() /*-{
                return this.obj;
        }-*/;
}
```

Normes d'implementació:

- Els "overlay types" declarats són una subclasse de JavaScriptObject. Els JavaScriptObject tenen un tracte especial al compilador de GWT, el seu propòsit és proporcionar una representació d'objectes JavaScript en codi Java.
- Han de tenir un constructor sense paràmetres i amb accés protected.
- Normalment els mètodes són del tipus JSNI (JavaScript Native Interface). Aquests accedeixen directament als camps JSON existents. Per disseny, tots els mètodes estan marcats com final i private.
- No obstant no tots els seus mètodes han de ser JSNI.

Tant en el mètode eval() com en els overlay types s'utilitza JSNI. Amb JSNI es pot cridar a mètodes JavaScript en el mòdul de GWT. El mètodes JSNI es declaren com native i contenen codi en JavaScript en un bloc de comentari /*-{ codi JavaScript }-*/. Aquest mètodes es poden cridar com si fossin mètodes Java.

6. Resultats

En aquest apartat es presenten els experiments que s'han portat a terme, els sistemes d'avaluació que s'han seguit i els resultats obtinguts pel algoritme de reordenació. En segon lloc es mostra, el resultat de la integració dels algoritmes de reordenació i agrupament a la interfície gràfica d'usuari.

6.1 Avaluació de l'algoritme de reordenació

6.1.1 Experiments

Els experiments s'han basat en un corpus extret dels arxius de la CCMA. El primer pas va ser seleccionar un conjunt de consultes de text d'una llista de termes utilitzats pels documentalistes que anoten manualment els *assets*. Tot seguit, es va simular un cercador textual per recuperar una llista de *keyframes* associats als *assets* que a les seves metadades hi havia el terme de la consulta.

Per poder avaluar la rellevància a nivell de *keyframe* dels resultats obtinguts, per cada consulta, es van anotar manualment tots els *keyframes* recuperats com rellevants o irrellevants per a la consulta amb el GAT (Graphical Annotation Tool)[12]. El criteri per anotar si eren rellevants va ser si el *keyframe,* per ell sol, mostrava el contingut de la consulta.

A la ***Taula 3*** es mostra les cinc consultes que es van seleccionar amb el número total de d'*assets* i *keyframes* i els *keyframes* anotats manualment com a rellevants.

Consulta	*#assets*	#KFs
Tennis de taula	3	1.116
Formula 1	6	3.441
Parlament	12	2.8416
Accident	8	66
Futbol	16	416

Taula 3. Consultes seleccionades

En aquest projecte s'ha estudiat la influencia dels diferents modes de filtratge en els resultats obtinguts després de la reordenació. Aquest impacte s'ha avaluat tenint en compte la rellevància a nivell de *keyframes*, que s'ha anotat prèviament, i la diversitat d'*assets* en les primeres posicions.

Per a cada consulta textual s'ha realitzat una reordenació amb els diferents modes de configuració de filtratge: intra-*asset*, inter-*asset*, intra&inter-*asset* i sense filtratge, només amb la passejada aleatòria. A més, s'ha avaluat els resultats inicials obtinguts de la cerca textual sense aplicar cap processat, baseline. Bàsicament, en aquest darrer cas la llista de resultats era la seqüència d'imatges recuperades per la cerca textual, ordenades per *asset* i, dins cada *asset*, segons l'ordre temporal.

Les diferents variables del mòdul de la passejada aleatòria (l'alpha epsilon i puntuacions inicials) no han estat estudiades en aquest treball i s'han fixat empíricament amb els següents valors:

Variable	Definició	Valor predeterminat
alpha (α)	Indica com influeix les puntuacions inicials en les probabilitats de la passejada aleatòria.	α= 0.8
epsilon (ε)	Indica el número d'iteracions que ha de fer la passejada aleatòria.	$\varepsilon = N \cdot \log_{10} Tamany$
puntuacions inicials (v(j))	Puntuacions que tenen els *keyframes* abans de realitzar la reordenació.	Les puntuacions inicials són uniformes per tots els *keyframes* de la mateixa consulta.

Taula 4. Variables fixades del mòdul de la passejada aleatòria.

6.1.2 Sistemes d'avaluació

L'algoritme de reordenació s'avalua segons la rellevància dels *keyframes* així com la diversitat d'*assets*. Per mesurar aquestes dues qualitats s'han utilitzat: l'*average precision* i *l'average asset diversity*.

Les dues mesures AP i AD es calculen per a cada consulta textual i els seus valors s'han promitjat entre totes les consultes per obtenir el Mean Average Precision(MAP) i el Mean Average *Asset*-Diversity (MAD).

6.1.3 Resultats

Les figures ***Figura 38*** i ***Figura 39*** mostren el MAP i MAD. A simple vista es pot veure que el fet d'aplicar la passejada aleatòria augmenta la rellevància i la diversitat dels resultats, ja que, per les dues figures els valors de Baseline són menors que els altres modes.

El filtratge dels grafs de similitud té poc impacte en el MAP, amb una lleugera disminució quan s'aplica el filtrat inter-*asset*. Això es degut a que el mode inter-*asset* realitza una operació contraria al principi d'estimació de la rellevància del PageRank: el més rellevant és el que té més arestes i, la eliminació d'aquestes arestes provoca que *keyframes* molt rellevants baixin la seva importància i que d'altres no tan rellevants augmentin la seva.

A la ***Figura 39*** demostra que el filtratge sí que augmenta la diversitat d'*assets* en els resultats. També es pot veure que el filtrat inter-*asset* disminueix significativament el MAD degut a que es creen illes de *keyframes* rellevants on les seves puntuacions disminueixen a favor dels *keyframes* del mateix *asset*. Tot i així, els millors resultats s'obtenen quan es combinen els modes inter i intra-*asset*.

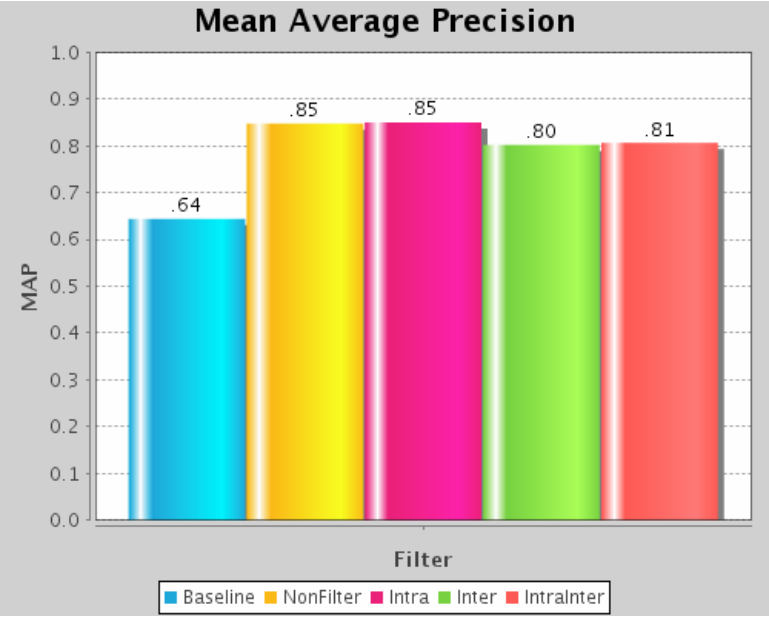


Figura 38. Mean Average Precision

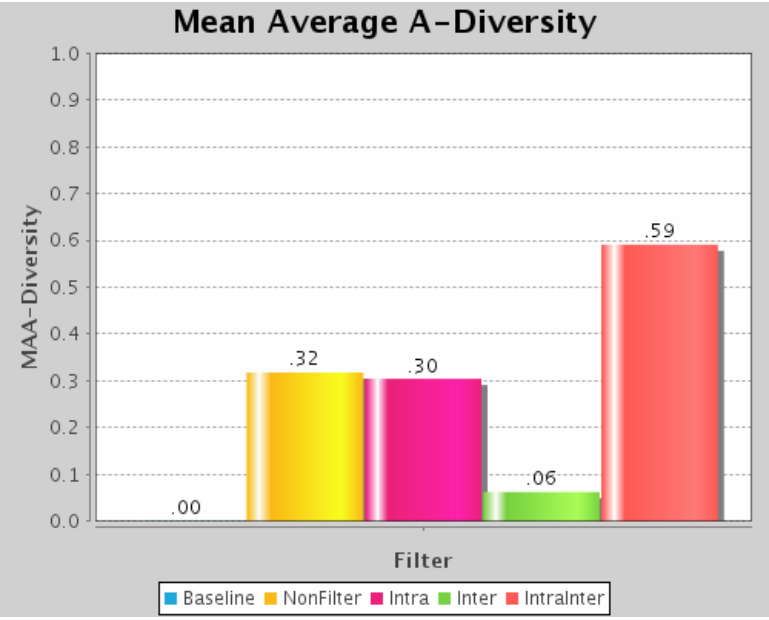


Figura 39. Mean Average A-Diversity

Els resultats per cada concepte es mostren a les figures ***Figura 40*** i ***Figura 41***. La primera conclusió que es pot extreure és que no tots els conceptes presenten el mateix comportament descrits pel MAP i el MAD. Per exemple, el filtrat intra&inter-*asset* no és la millor solució pels conceptes "Formula 1" i "Accident" en termes de diversitat, AD.

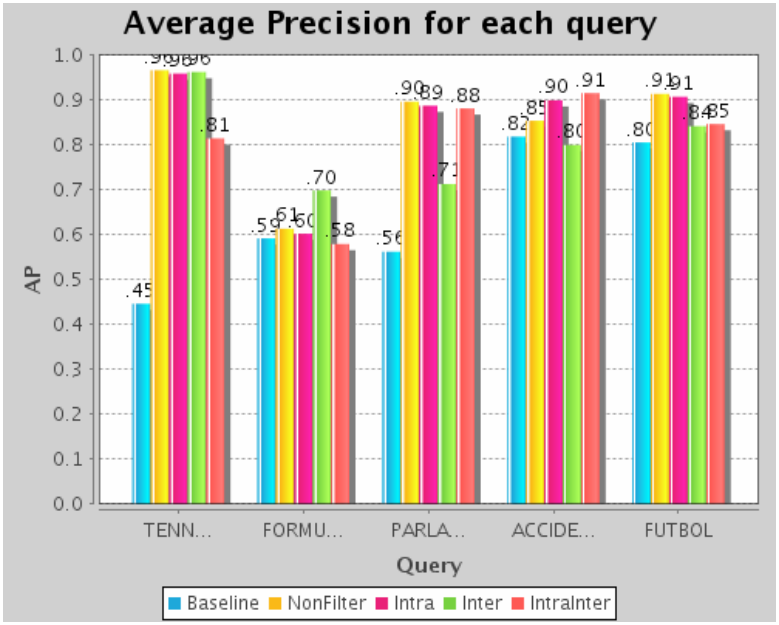


Figura 40. Average Precision per cada consulta

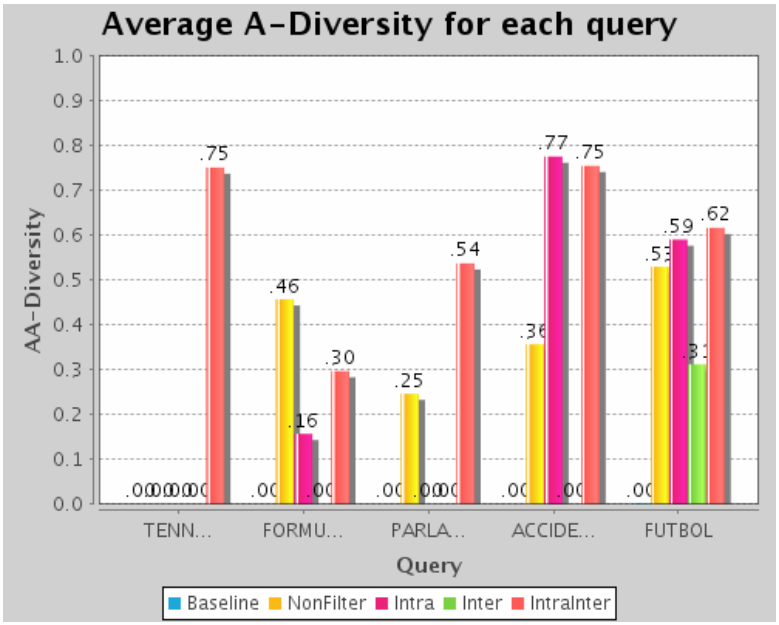


Figura 41. Average A-Diversity per cada consulta

Les estratègies de filtratge requereixen d'un esforç computacional per eliminar les arestes dels grafs de similitud però, per l'altre banda, també simplifiquen les iteracions del procés de la passejada aleatòria. Les mesures experimentals mostrades a la ***Taula 5*** indiquen que no hi ha una conclusió genèrica sobre el impacte del filtratge. També s'observa que en el cas de la combinació del filtratge intra i inter *asset* és aconsellable realitzar el filtratge intra-*asset* primer perquè aquest pas, normalment, elimina moltes més arestes que l'inter-*asset*. Quan s'aplica primer, aquest redueix el nombre d'arestes que el cas inter-*asset* ha de repassar i, per tant, redueix el cost computacional.

Consulta	NonFilter	Intra	Inter	IntraInter
Tennis de taula	80.943	95.649	97.154	109.971
Formula 1	495.578	819.981	610.358	855.534
Parlament	575.740	946.400	724.752	977.864
Accident	1.277	806	734	737
Futbol	12.382	6.148	5.284	6.341

Taula 5. Cost computacional de l'algoritme de reordenació (ms)

6.2 Interfície gràfica d'usuari

El segon objectiu d'aquest projecte era el disseny i desenvolupament d'un nou mòdul de *keyframes* al Digition que integres els sistemes de reordenació i agrupament, però que mantingues la seva perspectiva original.

6.2.1 Estructura

A continuació, s'explica el disseny del nou mòdul de *keyframes* i la funció que ha realitzat la part de simulació durant el desenvolupament de la GUI, ***Figura 42***.

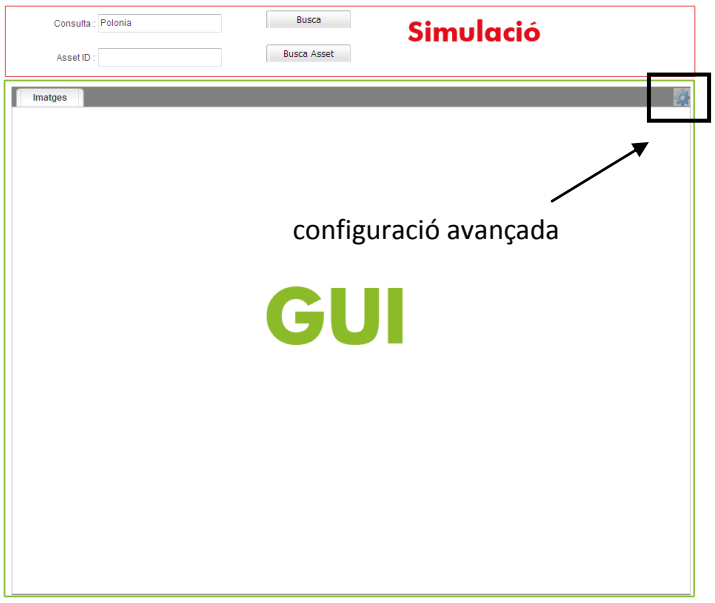


Figura 42. Estructura utilitzada durant el desenvolupament

Simulació

En aquest projecte s'ha simulat un sistema que permet realitzar els mateixos passos que els usuaris fan al Digition per recuperar els *assets* d'una cerca textual durant el desenvolupament de la nova GUI.

- **Consulta:** es considera que un usuari fa una cerca textual amb la paraula clau "Polonia". Tots els *assets* resultants es mostraran, en forma de text, al mòdul de resultats del Digition. Aquesta simulació s'ha implementat per ser utilitzada a la pestanya "Imatges" perquè els resultats de la cerca textual també es mostrarien visualment.
- ***Asset* ID:** Normalment, l'usuari selecciona l'*asset* que més li interessa entre els obtinguts per veure els seus *keyframes*. Aquesta acció es simulada introduint un valor per l'"AssetID" que s'utilitza per generar la resta de perspectives que s'han desenvolupat.

GUI

La nova GUI que es presenta esta formada per un sistema de pestanyes on cada una mostra una perspectiva diferent per a cada resultat. Totes les perspectives que fan referència a un únic *asset* i comparteixen tres característiques principals:

1. Es poden tancar
2. Es mostra un menú quan es clica amb el botó dret del ratolí. Aquest menú dóna les opcions de tancar la pròpia pestanya o les demés.

Figura 43. Menú de les pestanyes

3. Títol de la pestanya: el títol de la pestanya esta format per la primera lletra de la tècnica que s'aplica i pel *asset* ID.

A més, es mostra el botó de configuració avançada amb el que l'usuari pot escollir, a través d'una finestra, entre aplicar la reordenació i/o agrupament als *keyframes* d'una perspectiva. Aquest botó esta actiu a les perspectives clàssiques i desactivat a la resta.

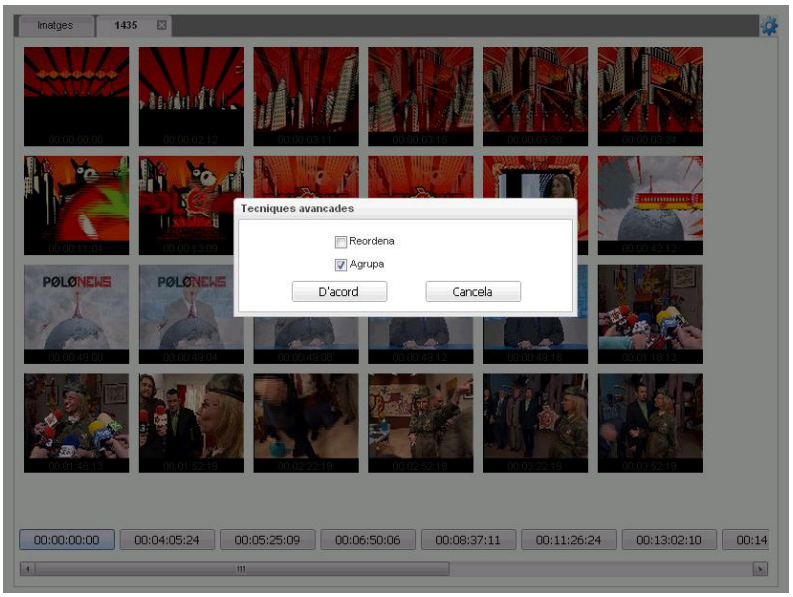


Figura 44. Finestra: Tècniques avançades

6.2.2 Perspectives

Els *keyframes* es poden mostrar amb quatre perspectives diferents segons si s'han aplicat les tècniques de reordenació i/o agrupament o no. Aquestes són:

1. **Perspectiva clàssica: *keyframes* ordenats temporalment**

   Els usuaris poden navegar pels *keyframes* a través de la fila de botons temporals. Cada botó indica el timeCode del primer *keyframe* dels resultats que mostra. El número de resultats que es mostren no esta fixat sinó que depèn de la mida de la GUI i l'espai entre les imatges.

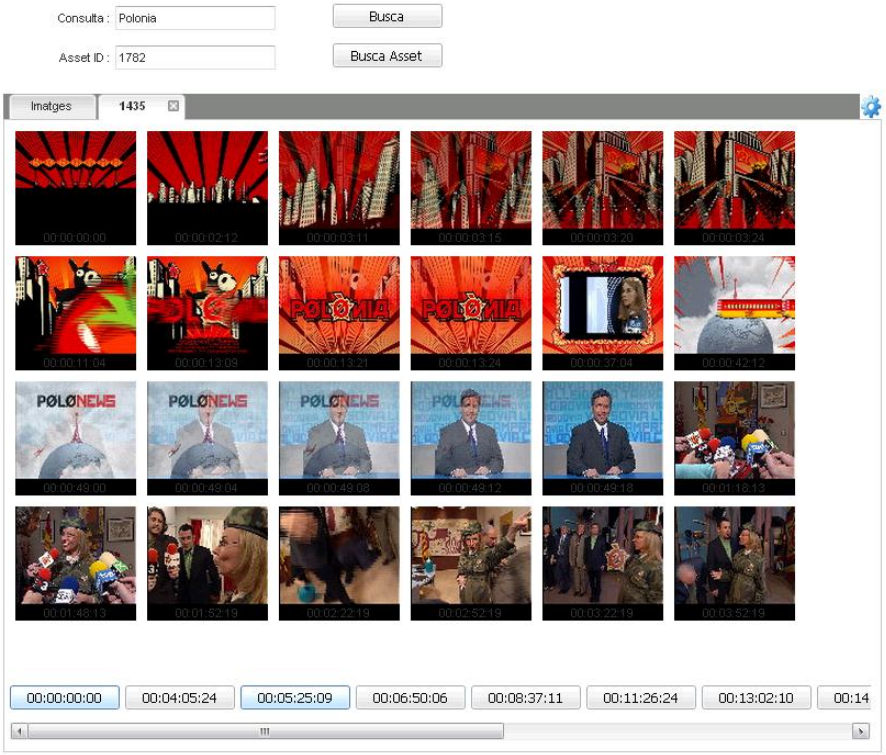


Figura 45. Perspectiva clàssica

Des d'aquesta perspectiva es té l'opció d'aplicar les tècniques de reordenació i agrupament a través del botó de configuració avançada.

2. *Keyframes* amb reordenació

A la perspectiva de la ***Figura 46*** es mostren els *keyframes* reordenats. Aquesta és la perspectiva convencional de molts cercadors d'imatge. A diferencia de l'anterior, l'eix temporal es perd degut a que els criteris de ordenació han estat uns altres.

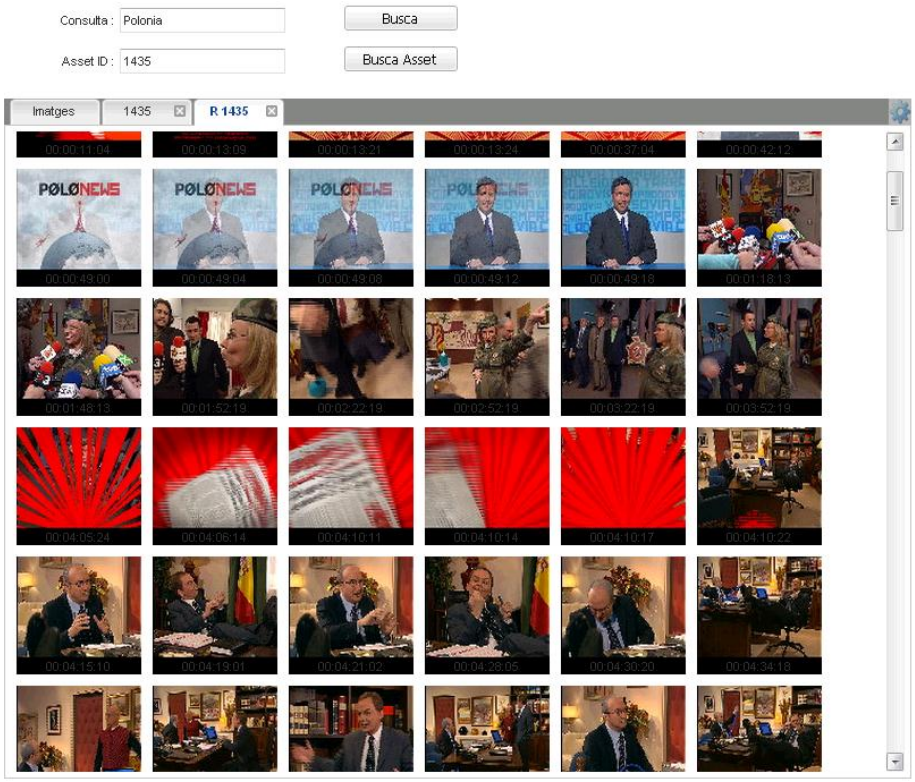


Figura 46. Perspectiva: keyframes amb reordenació

3. *Keyframes* amb agrupament o reordenació i agrupament

Aquesta perspectiva s'utilitza per qualsevol configuració on una de les tècniques escollides sigui agrupament. Els *keyframes* representants del grup es mostren agrupats en escala i un darrera l'altre. Per tal de tenir una idea general del tipus d'imatges que es troben dins del grup. El desenvolupador pot configurar el número de *keyframes* representants. A la ***Figura 47*** s'ha fixat a quatre.

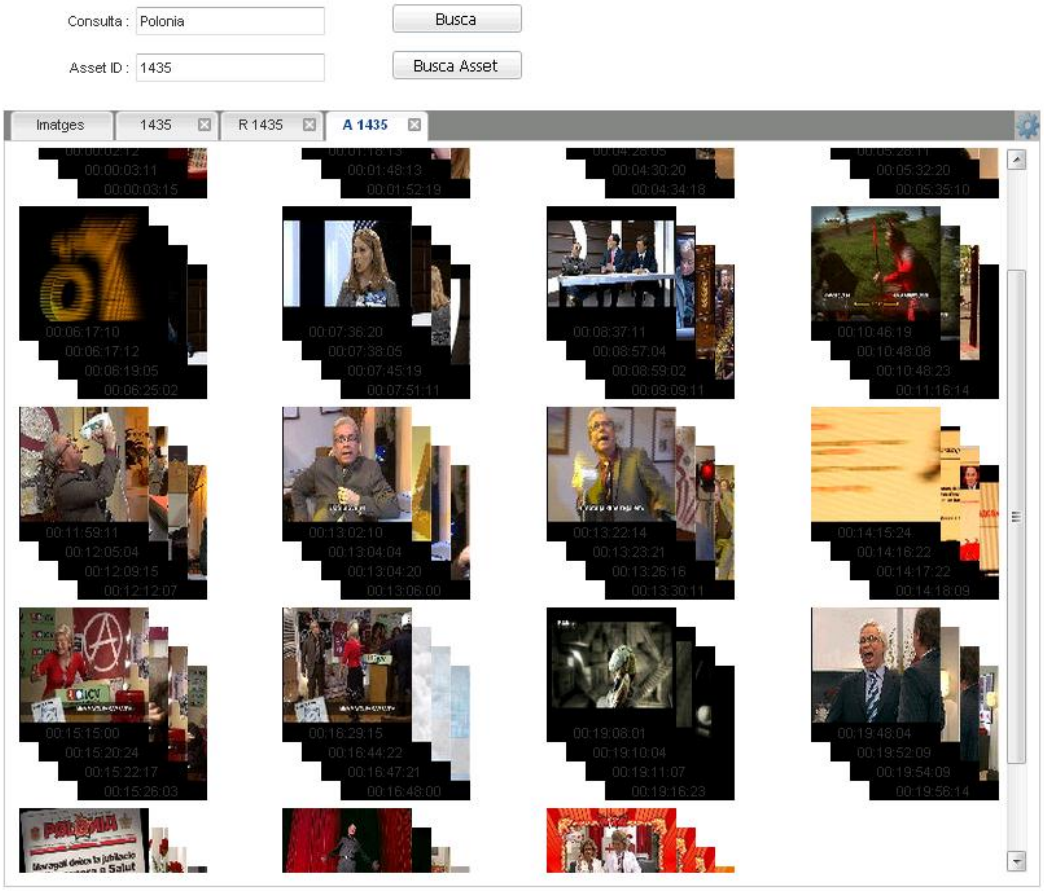


Figura 47. Perspectiva: keyframes amb agrupament o reordenació i agrupament

Exploració dels grups:

L'usuari pot navegar pels grups de dues formes diferents, la primera d'elles esta dirigida per fer vistes ràpides dels *keyframes* representatius i, la segona a l'exploració de tots els *keyframes* del grup.

- La funció **vista ràpida** permet al usuari fer una ullada als *keyframes* representants que hi ha al darrere i, que per tant, no es veuen sencers. L'usuari pot accedir a aquesta funció col·locant el ratolí sobre el *keyframe* que vol veure i, automàticament, aquest es posicionarà davant de la resta de *keyframes* del grup, ***Figura 48***. Quan es retira el ratolí, torna a la posició inicial.

Figura 48. Vista ràpida dels keyframes representants

- L'exploració de tots els *keyframes* que formen un grup es porta a terme mitjançant un clic al grup d'interès. Immediatament, s'obrirà un panell on es mostraran tots els *keyframes*,***Figura 49***. Aquest panell s'ha implementat amb una restricció i és que només es permet tenir-ne un d'obert. Per altra banda, l'usuari podrà tancar-lo o bé utilitzant la icona de tancar del propi panell o tornant a cliclar al grup. Per contra, si el que vol es tancar-lo per obrir un altre el que pot fer és cliclar directament al altre grup d'interès. Automàticament es tancarà el que estigui obert i es tornarà a obrir amb els *keyframes* del nou grup seleccionat.

  Un altre de les característiques d'aquest panell és la seva posició en la GUI. Un dels requeriments en el disseny era que sempre s'ha de mantenir dins de la superfície definida i que s'hauria de posicionar segons la posició del grup. Per defecte, el panell s'obre de dreta a baix, de forma que el grup queda situat a la punta esquerra superior del panell, ***Figura 49***. Però els comportaments dels grups de més a la dreta i els de sota canvia per a que no sobresurti de la superfície.

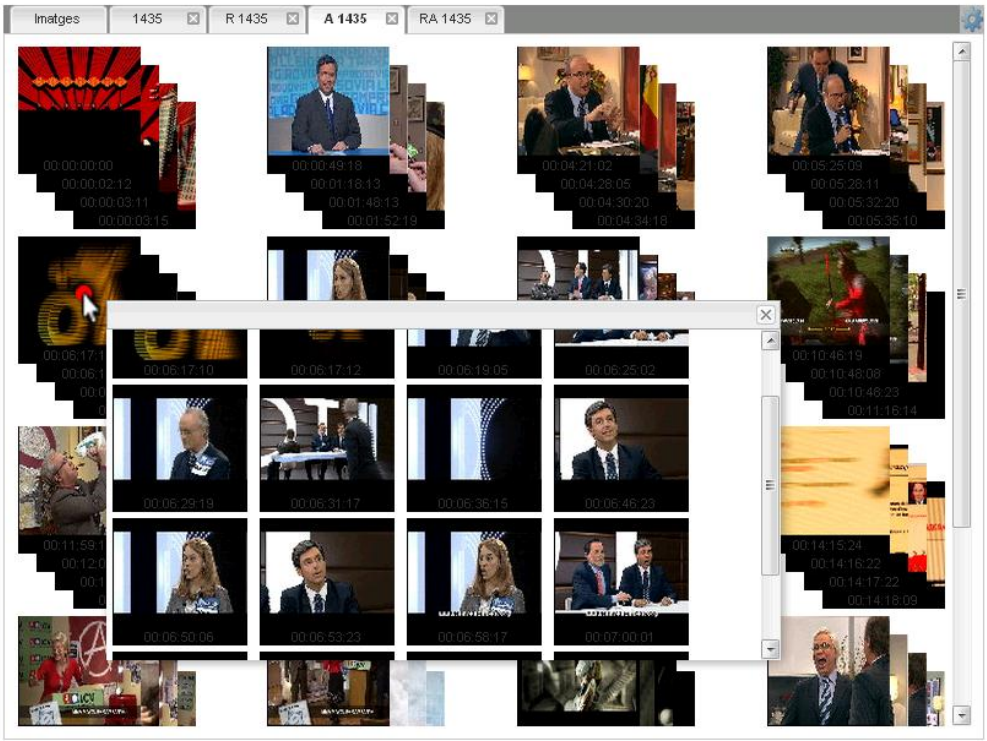


Figura 49. Exploració de tots els keyframes d'un grup. (grup superior esquerre)

A la ***Figura 50***, el panell s'obre de dreta a dalt perquè la superfície de visualització de la GUI no permet que s'obri cap a baix.

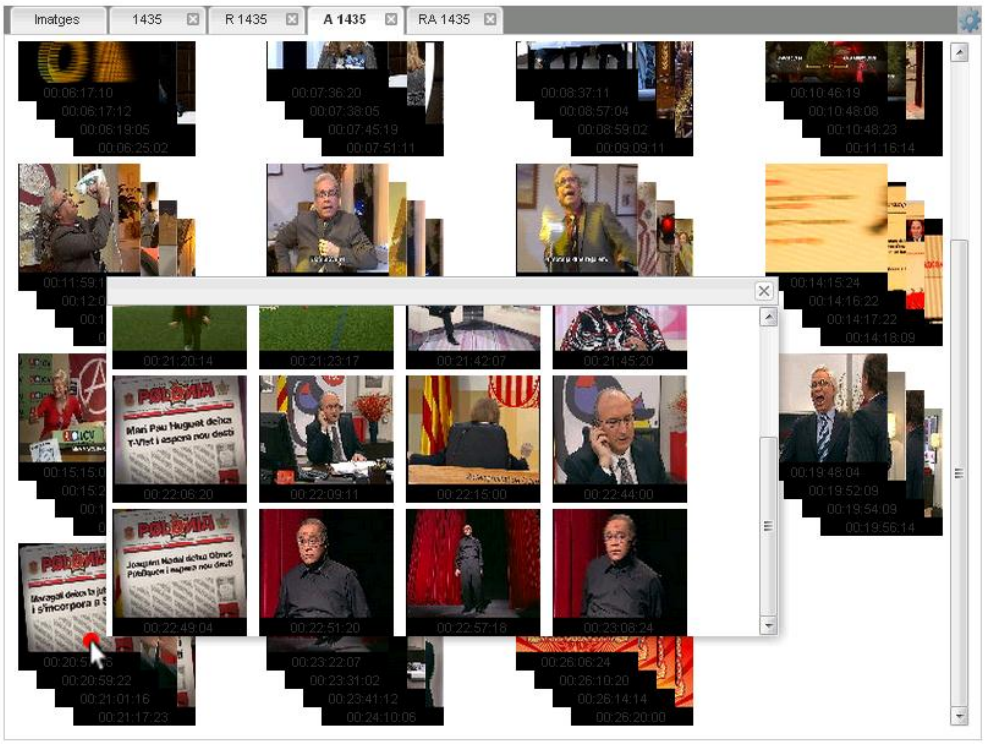


Figura 50. Exploració de tots els keyframes d'un grup. (gurp inferior esquerre)

A la ***Figura 51***, el panell s'obre d'esquerre a dalt, ja que els marges inferior i dret de la GUI no permeten que s'obri amb el seu comportament per defecte.

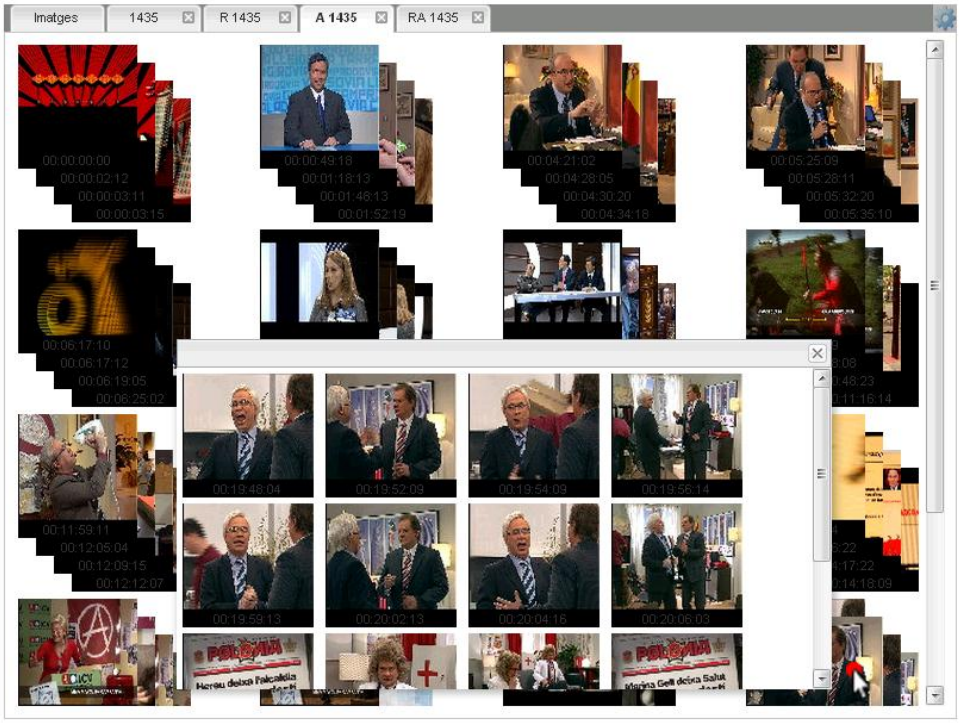


Figura 51. Exploració de tots els keyframes d'un grup (grup inferior dret)

4. **Perspectiva: “Imatges”**

   A diferencia de les altres perspectives, aquesta conté tots els *keyframes* de tots els *assets* resultants de la cerca textual ordenats i agrupats. Segueix la mateixa estructura que la perspectiva anterior però aplicada per més d’un *asset*.

   No s’han pogut generar captures d’aquesta perspectiva degut a que els serveis web no s’han integrat a la GUI i no ha estat possible simular-lo per la seva complexitat.

7. Conclusions

7.1 Assoliment dels requeriments.

El projecte inicial s'ha dividit en dos parts ben diferenciades: per una banda el disseny dels motors de reordenació i agrupament i, per l'altra la implementació d'una interfície de cerca que integrés les dues tècniques anteriors. Mentre que la primera part s'ha desenvolupat des de l'entorn de treball del UPSeek i la segona des de l'entorn de treball de la CCMA; fets que demostren la complexitat i extensió del projecte.

En aquest apartat es vol fer un petit resum del resultat final del projecte. Per tant, les conclusions es poden encabir en dos parts:

7.1.1 Reordenació

En aquest treball s'ha presentat un sistema per implementar una solució de classificació visual basada en el càlcul del algoritme de la passejada aleatòria sobre un graf de similitud. El graf de similitud depenent de la consulta ha sigut filtrat amb diferents estratègies, intra i inter *asset*, amb la finalitat de solucionar els problemes de diversitat de les tècniques bàsiques per recuperar informació en el corpus de la CCMA.

La evolució ha demostrat la validesa del enfoc de la passejada aleatòria per detectar la rellevància dels *keyframes* en l'anotació manual a nivell d'*asset*. La repetició de determinats *keyframes* en múltiples *assets* pot ser interpretat com una anotació implícita del contingut en termes de rellevància. Aquest fet s'explota mitjançant l'algoritme de la passejada aleatòria per determinar els *keyframes* que son més representatius per a la consulta.

Els resultats presentats demostren que el filtrat del graf de similitud és una tècnica vàlida per millorar la diversitat d'*assets* en els resultats basats en *keyframes*. La experimentació suggereix que el filtrat intra-*asset* augmenta la diversitat per si sola, i que el filtrat inter-*asset* només té un impacte significant quan es combina amb el filtrat intra-*asset*.

El pas de filtratge augmenta significativament la diversitat d'*assets* amb un petit impacte en l'estimació de la rellevància. Les observacions també demostren que el tipus de concepte i el conjunt dels *assets* tenen un impacte en el rendiment global de l'algoritme, amb gran desigualtat si s'utilitza els filtrats intra i inter *asset* per separat. No obstant, quan les dues estratègies es combinen els resultats són més estables i generalment millors.

Des del punt de vista computacional, la introducció de les estratègies de filtrat poden augmentar els esforços computacionals, així com la disminució en la construcció de grafs de similitud més simples. Els resultats suggereixen que el rendiment pot disminuir per grans conjunts de dades però millorar pels petits.

La classificació de la diversitat pot ser entesa com un mètode per a la explotació del contingut implícit de la anotació. En primer lloc, s'utilitza la repetició del contingut a la base de dades per detectar quins *keyframes* són els més rellevants. Aquest principi esta basat en la suposició de que el material rellevant és el que més sovint es reutilitzat pels editors de vídeo al generar més material pel repositori. En segon lloc, la organització dels *keyframes* dins dels *assets* és un segon tipus d'organització que es genera pel extractor de *keyframes*. La

explotació, de forma implícita, de les dades generades és presenta com una línea d'investigació prometedora, ja que, permet la millora dels resultats recuperats sense cap esforç addicional per part dels documentalistes per anotar el seu contingut.

7.1.2 Integració CCMA-UPSeek

La interfície dissenyada s'ha implementat amb èxit, de forma que integra els algoritmes de reordenació i, especialment, l'agrupament. Però en el moment de tancar la redacció d'aquesta memòria no compta amb els resultats agrupats i/o reordenats procedents del servei web de la UPC. Aquesta eina permetrà mostrar a l'usuari més resultats en menys espai i, a més, que aquests es presentin ordenats. Tot plegat permetrà a l'usuari fer cerques visuals més ràpides i intuïtives.

Pel que fa a les comunicacions, un dels requeriments de la CCMA en aquest tema és que les respostes del servidor de la UPC siguin prou ràpides per a que els usuaris rebin informació sense que s'hagin d'esperar.

Durant el desenvolupament del algoritme de reordenació se'ns van plantejar un gran repte degut al compromís existent entre velocitat en el càlcul de l'algoritme i memòria disponible. Els experiments de l'algoritme de reordenació s'han portat a terme amb una gran quantitat de *keyframes*, fet que portava a que l'algoritme es quedés sense memòria durant el seu processat. Com a resultat es va optar per una solució de pre-càlcul a disc que feia que l'algoritme fos més lent. Aquesta solució va fer que les respostes per part del servidor de la UPC no fossin prou ràpides per a que el client obtingues els resultats processats a temps. Finalment, es va optar per dos corpus de treball diferents per experimentació i explotació. En el primer amb un número màxim de veïns per cada node del graf de similitud, i el segon amb un número més reduït. Com a conseqüència, el temps de càlcul de l'algoritme es va reduir per l'explotació.

Als últims dies de la realització d'aquest projecte es va aconseguir respostes en menys d'un minut. En concret, per una petició de 1000 *keyframes* amb reordenació o agrupament el servidor de la CCMA rep la resposta en 20 segons. El fet que aquesta fita s'hagi assolit els últims dies no ha permès integrar el servei web al client. Degut a això, a l'actualitat la interfície treballa amb un simulador, també implementat durant el treball, que genera grups de *keyframes*.

Tampoc ha arribat a temps la integració del nou mòdul de *keyframes* a l'eina de la Corporació, el Digition, tot i que era un dels objectius per assolir. No obstant, l'eina desenvolupada podrà ser aprofitada per ser presentada a un usuari final més endavant.

7.2 Treball futur

El treball futur, per part de la UPC, inclou l'experimentació de les tècniques de reordenació pel cas de les consultes visuals, el que proporcionarà una puntuació inicial que determinaran les puntuacions de la passejada aleatòria. Així com, la realització d'experiments per trobar el paràmetre epsilon òptim procedent de la passejada aleatòria.

A més, tot i que la tècnica de reordenació ha demostrat ser una solució vàlida s'obtindrien millors resultats si es combina amb altres estratègies, com l'agrupació per augmentar la diversitat, la retroacció per rellevància per aprendre els pesos dels descriptors visuals, les preferències dels usuaris o el historial del repositori. Un altre línia d'investigació, és explotar la cooperació entre els diferents grafs de similitud construïts per cada descriptor visual.

El treball futur per part de la CCMA seria integrar el servei web al client i incloure la interfície al Digition. A més de realitzar les proves d'usuaris pertinents per tal de poder millorar aspectes relacionats a les funcionalitats de la interfície.

7.3 Conclusions personals

Personalment, aquest projecte m'ha aportat una experiència en el treball en equip, en concret, en dos entorns diferents, amb formes de treballar diferents. El fet de voler realitzar el meu projecte en col·laboració amb una empresa, en aquest cas la CCMA m'ha permès entrar en contacte amb el món industrial i completar així la formació obtinguda a la universitat.

El fet d'haver treballat activament en els dos equips m'ha aportat una visió general del procés necessari per crear una eina com aquesta. Des del disseny i implementació d'una nova tècnica fins al desenvolupament d'una interfície per tal de portar als usuaris finals les tècniques explicades anteriorment. Passant pels requeriments de respostes ràpides per part de la CCMA i els problemes que han portat, ja que com a desenvolupadora del UPSeek era un tema que també havia de solucionar.

Val a dir, que els meus coneixements dels llenguatges emprats, així com les comunicacions entre clients i servidors eren pràcticament nuls en el moment d'iniciar el projecte. En definitiva, aquest projecte ha estat una experiència positiva, que m'ha enriquit professionalment i personalment, i un bon punt final per a l'etapa que deixo enrere.

8. Referències

Bibliografia

[1] Yong Rui Huang, T.S. Ortega, M. Mehrotra, S. Beckman Relevance feedback: a power tool for interactive content-based image retrieval 2002

[2] Yushi Jing i Shumeet Baluja: PageRank for product image search. International World Wide Web Conference 2008

[3] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub. Extrapolation methods for accelerating pagerank computations. In Proceedings of the 12th international conference on World Wide Web, WWW '03, pages 261{270, New York, NY, USA, 2003. ACM.

[4] Winston H. Hsu, Lyndon S. Kennedy i Shih-Fu Chang: Video search reranking through random walk over document-level context graph. MM'07

[5] Fabian Richter, Stefan Romberg, Eva Hörster i Rainer Lienhart: Multimodal Ranking for image search on community databases. MIR'10

[6] Ting Yao, Tao Mei i Chong-Wah Ngo: Co-reranking by Mutual Reinforcement for Image Search. CIVR'10

[7] Feng Jing, Changhu Wang, Yuhuan Yao, Kefeng Deng, Lei Zhang I Wei-Ying Ma: IGroup: Web image search results clustering. MM'06

[8] ChengXiang Zhai, William W. Cohen, John Lafferty: Beyond Independent Relevance: Methods and Evaluation Metrics for Subtopic Retrieval. SIGIR'03

[9] Liangliang Cao, Andrey Del Pozo, Xin Jin, Jiebo Luo, Jiawei Han, Thomas S. Huang: RankCompete. Simulataneos ranking and clustering of web photos. WWW'10

[10] Manjunath, B. S.; Salembier, Philippe; Sikora, Thomas (abril de 2002). 101 Interfície de cerca basada en regions Introduction to MPEG-7: Multimedia Content Description Interface. Wiley & Sons. ISBN 0-471-48678-7.

[11] Pia Muñoz Trallero, "Extensió d'una interfície de cerca d'imatges a les consultes amb regions" CCMA ASI./ EET, Esplugues de Llobregat. Febrer-Juny 2010.

[12] Xavier Giro-i-Nieto, Neus Camps, Ferran Marques, GAT, a Graphical Annotation Tool for semantic regions, Multimedia Tools and Applications, 2009, (doi:10.1007/s11042-009-0389-2).

Articles en línia

[13] Google: PageRank [ref de Juny 2010] http://www.google.com/corporate/tech.html

[14] Mahout Wiki: Canopy Clustering [ref. de 19/09/2010] https://cwiki.apache.org/MAHOUT/canopy-clustering.html

[15] QT (Quality Threshold) Clustering. [ref. de 19/09/2010] http://www.chem.agilent.com/cag/bsp/products/gsgx/Downloads/pdf/qt_clustering.pdf

[16] Blog de Google [ref. de 3/01/2011] http://googleblog.blogspot.com/2009/11/explore-images-with-google-image-swirl.html

Annexos

Es presenten dos annexes al final del document:

Annex I mostra la proposta pel congrés ICMR 2011

Annex II mostra el conjunt d'escrits realitzats al bloc d'investigació BitSearch on s'ha anat descrivint el progrés del present projecte durant la seva realització.

Annex I *Comunicació enviada a l'ICMR 2011*

Diversity Ranking for Video Retrieval from a Broadcaster Archive

Xavier Giro-i-Nieto, Monica Alfaro, and Ferran Marques
Technical University of Catalonia (UPC), Barcelona, Catalonia / Spain
{xavier.giro, ferran.marques}@upc.edu

ABSTRACT

Video retrieval through text queries is a very common practice in broadcaster archives. The query keywords are compared to metadata manually annotated on video assets by documentalists. This paper focuses in 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. Previous solutions based on a random walked over a visual similarity graph have been modified to increase the asset diversity by filtering the edges between keyframes depending on their asset. The random walk algorithm is applied separately for ever visual feature to avoid any normalization issue between visual similarity metrics. Finally, this work also introduces the Average Diversity metric to evaluate the system, a complementary measure to the relevance estimation offered by the Average Precision.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: [Retrieval models]; I.4.10 [**Image Processing and Computer Vision**]: Image Representation multidimensional

General Terms

Algorithms,Measurement,Average Diversity

Keywords

Image Ranking, Video Retrieval, Similarity Graph

1. MOTIVATION

Television broadcasters store nowadays large and growing amounts of video data in their local archives that require efficient retrieval techniques. The traditional text-based queries and indexing strategies are being complemented with new descriptors automatically extracted from the visual and audio content. These new type of features open the door to promising possibilities in the video indexing and retrieval domains, such as query by example or high-level concept detectors.

Most video retrieval systems present the search results as a set of keyframes displayed on a graphical user interface. Keyframes become the representation units of the video assets as the user can obtain relevant information about the video by looking at its related keyframes. Keyframes offer many advantages as a representation unit when compared to textual metadata because humans can quickly interpret them and the amount of information per pixel they contain is normally far over the one coded in text.

The documentalists that work in a broadcaster archive usually produce textual metadata at the video asset scale, providing little or none information at the keyframe level. For example, in an interview to a popular character, the textual metadata of the video asset will typically contain the name of this person, although the asset may also contain keyframes where only the interviewer appears. As a result, the manual annotation of the video asset will not apply to all keyframes and, in extension, not all keyframes retrieved by a text-based search will be relevant for the query.

The scale mismatch between video annotation and keyframes display poses a question when choosing which keyframes are to be shown to the user. One extreme option would be to show all keyframes for every video asset matching the query. As discussed in the previous paragraph, this solution would probably produce the selection of several irrelevant keyframes. Moreover, the resulting set may include many similar images because in a TV broadcaster archive an important proportion of the assets are generated by a set of fixed cameras (studio, sports events, soap opera...), a configuration that produces several nearly duplicate keyframes. As a result, choosing all keyframes from the retrieved assets would probably result into an inefficient use of the GUI, that will be populated with many repetitive and irrelevant keyframes.

On the other extreme, selecting one keyframe per video asset may imply several problems as well. Firstly, it is necessary to correctly select this representative keyframe, a challenging task as no manual annotation is available at the keyframe scale. Moreover, the assumption that all the relevance of the video asset can be fully expressed with a single keyframe may proof wrong in many cases. Consider for example a soccer match where several goals are scored. All these moments are relevant for that match and, in fact, they are normally include in every game highlights and posterior analysis. The most realistic approach is to consider that

multiple keyframes may be relevant in an asset and their detection will requery exploiting other techniques that will complement the manually generated annotation.

Choosing relevant keyframes is not the only design criterion to be taken into account. In the TV broadcast domain, the typical user that accesses the archives is usually interested in retrieving different video assets. The retrieved material may be used, for example, to produce new assets from previously broadcasted content or to learn about the previous treatment of a story or topic in the TV station. In general, archive users will value more the variety of video assets than retrieving several relevant keyframes from the same asset. In this context, the diversity of assets is also a requirement. Once a video asset is found, its keyframes can always be further explored in detail by applying some temporal or relevance criteria.

This paper addresses the problem of building a relevant ranked list of keyframes that belong from a diversity of assets from the archive of Catalan Broadcasting Corporation (CCMA) [1], the public TV broadcaster in Catalonia. The technique considers the visual domain as the additional modality that allows an estimation of the keyframes relevance. The proposed solution uses the visual features of the keyframes because in many cases relevant images also present similar visual features. While manual annotation is costly, these additional visual descriptors are easy to obtain because they can be automatically extracted and processed with no need of human interaction. The organization of keyframes in assets will be considered to guarantee a diversity of assets among the top hits of the results. In order to evaluate this later requirement, this paper introduces a new metric called *Asset Diversity*.

The remain of this paper is structured as follows. Section 2 reviews some of the previous works that have inspired the proposed algorithm. The proposed technique is described in Section 3, while Section 4 provides experimental results on a set of data extracted from a TV archive. Finally, section 5 draws the conclusions and future work.

2. RELATED WORK

Ranking techniques offer solutions for the automatic sorting of an initial set of results according to some auxiliary criterion different from the one used at query time.

A first family of techniques are inspired in the relevance feedback techniques. Relevance feedback systems require the user interaction to determine which of the initially retrieved results are relevant for the query. In order to avoid the user presence and obtain an automatic solution, the pseudo-relevance feedback techniques [7] consider the first results in the ranked list as relevant and the latter as non-relevant. By simulating the user interaction, the original ranked list is modified according to a relevance feedback technique and the reranked list is obtained. This type of solutions requires that the initial query generates a ranked list, a feature that may not be available in basic text-based search engines.

A second strategy for ranking is based on assessing the similarity between the elements in the initial set of results. This similarity measure can be defined on a type of feature (textual, visual, audio, social, multimodal...) and offers a wide range of possibilities for their combination. These measures are typically represented under the form of a *Similarity Graph (SG)* [3], where each node corresponds to a document in the database and each edge is weighted according to the similarity between the two connecting nodes. Once the SG is built, an estimation of the relevance of every node is obtained by considering that those nodes with more connections to other highly connected nodes are the most relevant for the query. This approach is inspired by the web search engines that sort their results according to the links that connect the web documents, such as the PageRank [5]. In the web search case, those sites referred by other important sites are considered important and are ranked first after a text query.

Once the graph-based representation of the items is generated, the structure is exploited to estimate the relevance score for every node. A popular approach to solve this problem is the *random walk*, an algorithm that identifies the document relevance with the probability of finding a traveller at the node if that traveller randomly jumps from node to node with a probability of taking a path proportional to the edge weight. Despite there is an exact solution for the algorithm, its computation requires the inversion of a very large matrix, a complex problem that is normally avoided by applying an iterative estimation according to the Power Method [4]. If available, the scores in the initial ranked lists can be naturally combined with those obtained through the random walk process by adjusting a leverage factor α. In the web search domain, the random walk is executed offline over the whole set of indexed documents so that, at query time, the retrieved ranked list can be quickly build according to these precomputed query-independent scores.

Previous work has applied the presented or similar principles for image or video retrieval. The work by Jing and Baluja [3] applied PageRank to rerank the image results obtained after a text query on the Google image search engine. Their conclusions report an increase on user satisfaction and a decrease on the amount of irrelevant results among the top hits. Hsu et al. [2] applied the random walk solution in a news broadcast archive that expanded the results obtained by a textual query to new assets connected through the SG ("context graph" in their paper). The edges on this graph were weighted by a multimodal similarity measure computed as a linear combination of textual (ASR and transcripts) and visual (salient points) similarities. In their work they studied two options to generate the SGs: consider the whole database (Full Ranking) or only those documents retrieved by the initial query (Partial Ranking). Their experimental results clearly show that the Partial Ranking solution is a much better option because the relevance of every node is clearly query-dependent. Moreover, their research also studied two connectivity options when building the SG: full connectivity and a reduced connectivity limited to the K-nearest neighbours that reduces the computation effort when solving the random walk. The reported results conclude that connectivity reduction is advisable for efficiency but if K is too small it may have a negative impact in precision performance. Another multimodal approach proposed by Richter

[1] http://www.ccma.cat

et al. [6] combined textual and visual features to build a SG which was later filtered to reduced the impact of similar images coming from the same contributor in the context of community databases (eg. Flickr). Finally, Yao et al. [8] used a different SG for every modality (visual and textual) whose values are iteratively propagated to the other SG to initialize a new random walk. This was, the two modalities are combined through a mutual and iterative exchange of information. Cao et al [1] proposed an algorithm that simultaneously reranks and clusters in two classes a collection of images. A single SG holds two random walks that compete for each node by iteratively estimating the probability scores and then assigning each node to the random walk label with whom it obtained a higher score.

The related work offers several hints about how to develop a ranking solution for the broadcast domain but, up to the author's knowledge, none of them has combined them in a suitable way. The random walk has been reported as a valid method to estimate relevance, but several doubts arise about how to build the SG. In addition, the related cases have used SGs to fuse modalities, but non of them has worked with several SGs in the same modality to combine, for example, different types of visual descriptors. Finally, the filtering of edges as a antidote to avoid a bias to a specific user in social network can be adapted to any content database organised in sets where a diversity of these sets is desirable among the top ranked results.

3. PROPOSED SOLUTION

The presented study focuses on a broadcaster archive whose contents are stored under the form of video assets. Every asset consists of a video file, textual metadata and a set of automatically extracted keyframes. The search engine considered for this work is based on textual metadata that describes the complete asset. The broadcaster's retrieval system presents the search results on a graphical user interface based on keyframes. As previously presented in Section 1, the goal of this work is to generate a ranked list of the keyframes associated to the video assets that have been retrieved after a textual query. This ranking must be built accord to the relevance of the keyframes as well as providing a diversity of video assets among the top hits.

Figure 1 shows the architecture elements of the system in a case where the initial text query *President Montilla* has retrieved two video assets, one from an interview (asset A) and a second one from the news program reporting about the interview (asset B). The keyframes associated to the assets are processed to create a SG for each type of available visual descriptor, for example, color and texture histograms. At the next stage, each of these SGs may be filtered to reduce some undesirable effects produced by the repetition and unbalanced amount of similar keyframes among the considered assets. The random walk algorithm is applied on every resulting SG to obtain a probability score for every image in every considered visual descriptor. Finally, these scores are fused to generate the final ranked list.

Our work introduces two contributions in the state of the art. Firstly, the fusion of diverse visual similarities on the random walk scores instead of directly on the visual metrics, a way of avoiding any scaling problem between types of visual descriptors. Secondly, we explore the filtering strategies of some edges in the SGs to boost the diversity of assets in balance with the relevance of the top ranked keyframes.

3.1 Similarity Graphs Computation

In this paper, the nodes of the SGs represent a keyframe in the archive and the edges are weighted according to the visual similarity between two nodes. The computation of a SG poses three basic questions in our study case: (i) when to compute the distances between the keyframes, (ii) what degree of connectivity is required and, (iii) how to deal with multiple measures of visual similarity.

The calculation of visual distances is typically a computation-intensive effort that most systems perform offline. The process requires to evaluate the similarity between every pair of keyframes that are to be considered in the SG. Notice, though, that the exact topology of the SG is not known until query time, so the only SG that can be computed offline is a SG considering all keyframes in the database. This query-independent SG is read at query time to quickly build the SG that only considers those nodes contained among the initial query results. The process can be understood as a pruning of the full query-independent SG to build a query-dependent SG that keeps only the nodes and vertices retrieved in the initial query

The next question that arises is the degree of connectivity between nodes. Considering a full connectivity would result into a majority of very low weighted edges that would require an important computation effort during the random walk. In general, every keyframe in the database is considered similar to only a very small subset of other keyframes, so the full connectivity option is not necessary. Another option is setting a predefined amount of connections for every node as in the k-NN case of [2]. This approach would involve considering that all keyframes have the same amount of similar neighbours in the SG, an assumption which is wrong as visual similarity is content-depending. Besides, using a fix connectivity to all nodes would also imply serious architectural problems when reducing the query-independent SG to generate the query-dependent one, as during the pruning process the connectivity of each node will vary depending on the query. In order to solve this issue, the connectivity of the nodes is defined by setting a threshold on the similarity measure, that is, only establishing links between those nodes whose score is over a certain predefined threshold.

The final issue to be solved is how to combine different visual metrics defined for different types of features. Many retrieval systems use different types of features from the same or different modalities. In our case, four different visual descriptors were considered but the situation is analogous to combining features from different modalities. One option is to compute the similarity metric by fusing the scores obtained for each feature. This solution presents an important problem of combining values coming from different metrics, whose interpretation may be feature-dependent or, in other words, we may be mixing apples with pears. Another option is to delay the fusion of feature scores after the random walk, a process that generates probability scores for every node. This strategy avoids choosing any normalization strategy for the similarity metrics to safely combine probabilities. For

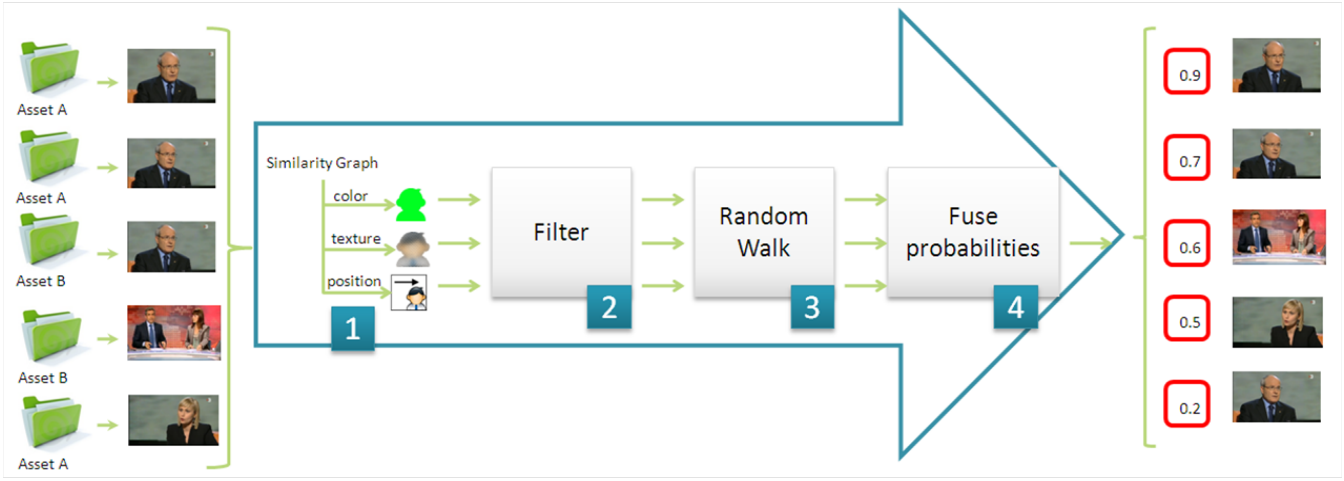


Figure 1: System architecture.

this reason, the second option was chosen and the obtained values are fused after the random walk through an averaging operation.

3.2 Edge Filtering

Given a SG, the random walk algorithm assigns a score to each of its nodes to estimate the relevance of each keyframe. The algorithm produces higher scores for highly connected nodes, especially if their neighbouring nodes are also highly connected. Applying this approach in the context of the described broadcaster archive may produce some undesirable results that can be corrected with a previous filtering of some of the edges. This section describes which problems may arise and proposes two filtering strategies to minimize them.

The goal of the designed ranking algorithm is to boost the relevance of the keyframes and the diversity of the video assets among the top hits in the results. The main assumption is that, given a set of video assets retrieved after the text query, relevant shots will be repeated in multiple different assets. The assumption of *repetition* must be decreased to *nearly duplicate* due to the different behaviour that even the same keyframe extractor may present when dealing with edited versions of the same content appearing in different video assets.

If no post-processing is applied to a SG, the desired requirements may not be achieved due to two basic problems. Firstly, whenever an asset contains a proportional large amount of similar keyframes, these keyframes will tend to be highly scored due to their intra-connectivity, even if they do not appear in any other of the retrieved assets. For example, Figure 2 shows the expected results in the case of the two video assets introduced in Figure 1. In this example, the keyframes showing the TV set may occupy the top hits in the ranked list despite not being the most relevant for the query. Notice that these keyframes were not even included in the news program, a proof of its irrelevance. The repetition of nearly-duplicate keyframes for the same asset is a common situation because the keyframe extractors are normally designed to generate keyframes to summarize the

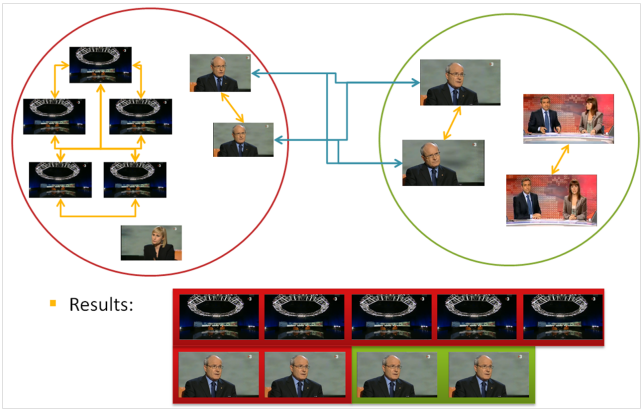


Figure 2: Irrelevant keyframes among the top hits.

asset contents along the temporal dimension.

Even if the most repeated keyframes in the asset are the most relevant ones, a second problem may occur in terms of diversity. If there are many similar relevant keyframes in every asset, the top hits will be composed of blocks of relevant keyframes grouped by video assets. This situation will harm the diversity of video assets among the first positions of the list and, for this reason, should be also treated. This scenario is reflected in Figure 3, where the three first top hits include nearly-duplicate keyframes from the same asset and the second relevant asset does not appear until the four hit when, ideally, this should be the second one.

In order to reduce the impact of these two situations, two strategies have been defined: *intra-* and *inter*-asset filtering of the SGs. These filtering operations aim at increasing the asset diversity by preserving at the same time keyframe relevance estimated by the randown walk.

3.2.1 Intra-asset filtering

As the main assumption of the algorithm is that a keyframe is relevant when itself or one of its nearly-duplicates appears in different assets, the first strategy that can be applied is

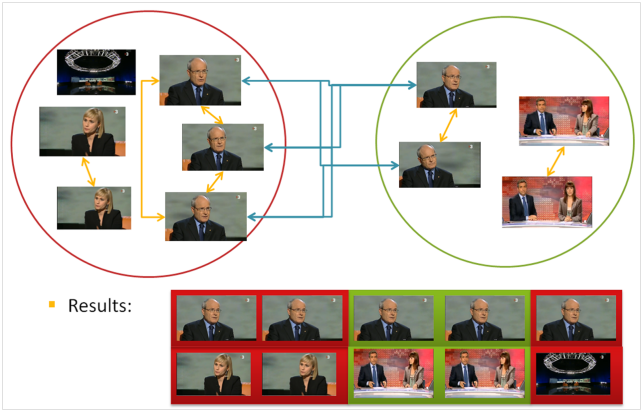


Figure 3: Blocks of assets among the top hits.

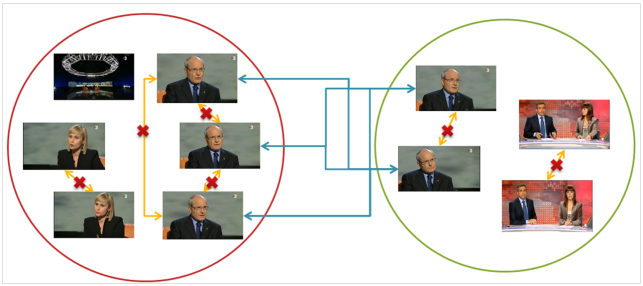


Figure 4: Intra-asset filtering.

to decrease the scores of those keyframes whose relevance is obtained from the same asset by deleting the edges between nodes in the same asset. This way, the only connections that a node can receive will come from an external asset, a topology that satisfies the proposed relevance definition. This operation is shown in 4.

3.2.2 *Inter-asset filtering*

Applying an intra-asset filtering may not be enough in some cases. Even by removing the intra-asset edges in the case shown in Figure 3, several keyframes from both assets would still occupy the first positions of the ranked lists in blocks, as their relevance would be increased by multiple edges originated in the same external asset. In other words, a near-duplicate in a second asset is a good sign of relevance, but several near-duplicates in this second asset should not increase the relevance. The next degree of contribution should come from a near-duplicate from a third asset. In order to avoid excessive relevance boost from a single external asset, the amount of edges connecting every node to nodes in another asset is limited to one. Other options in the literature propose to normalize the edge weights [6].

4. EVALUATION

The presented techniques for filtering the SGs at the intra- and inter-asset level were tested on a representative dataset from CCMA, the public national broadcaster in Catalonia. The obtained ranked lists were evaluated in terms of keyframe relevance and asset diversity for different query topics and the generated results compared.

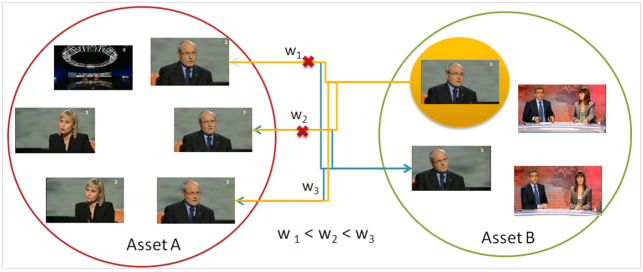


Figure 5: Inter-asset filtering.

Table 1: Test dataset

Query	# assets	# KFs
Table tennis	3	1,116
Formula 1	6	3,441
Parliament	12	2,816
Accident	8	66
Football	16	416

4.1 Experimental set up

The reported experiments were based on real data corpus extracted from the broadcaster archive. The first step was the selection of a set of text queries from a list of controlled terms used by the documentalists that manually annotate the video assets. For every query, all retrieved keyframes were individually annotated as relevant or not-relevant for the query. The criterion to establish the relevance was to consider if the keyframe may be tagged with the textual query by an annotator who would only access the keyframe, that is, with no knowledge about the rest of the data in the asset.

This annotation provided a ground truth over which the relevance of the retrieved keyframes was evaluated. Table 1 describes the annotation, composed of seven text queries, six of them on generic concepts and the last one referring to the title of a specific TV show. The table includes the amount of different assets, total amount of keyframes and amount of that was manually annotated as relevant for the textual query.

On the other hand, the whole set of keyframes retrieved by the textual query was reranked using four different MPEG-7 visual descriptors: Color Structure, Dominant Color, Color Layout and Texture Edge Histogram, which basically describe the color and texture distribution and spatial location of every keyframe. The random walk algorithm was initialized with uniform score for all nodes. Results were generated by comparing the generated ranked list for every filtering option and the ground truth contained in the manual annotation.

4.2 Metrics

This paper pursues the generation of results that accomplish two basic properties: relevant keyframes and diversity of assets. Two different metrics were used to evaluate these two qualities: the average precision and the average asset recall.

The *Average Precision (AP)* is broadly used by the retrieval community when evaluating the relevance of the retrieved results. This measure is obtained by averaging the first m precision values that can be obtained from the resulting ranked list as

$$Averaged\ Precision(AP) \equiv \frac{1}{m}\sum_{k=1}^{m} Precision(k) \quad (1)$$

where the $Precision(k)$ is the proportion of relevant keyframes when considering the first k positions in the ranked list.

The diversity of video assets has been measured with a new proposed metric inspired by the *S-recall* proposed in [9]. The S-recall stands for "subtopic recall" and measures the percentage of subtopics covered by the set of first K retrieved documents. In our study, the goal was to design a metric that behaved similarly to the AP, that is, a normalized value whose best output were the unit and that would introduce a larger penalization to the non-diverse results when they occur among the earliest positions than when they occur in the latest ones. In a first approach, the *Diversity at k* would measure the variety of the results as

$$Diversity\ at\ k \equiv D(k) = \frac{d(k)-1}{k-1} \quad (2)$$

where $d(k)$ corresponds to the amount of different video assets contained in the positions $1 \ldots k$ of the ranked list. Notice that this metric is only defined for $k \geq 2$ as the diversity can only be evaluated on a set of multiple items.

Combining the concept of Equation 1 with the diversity measure introduced in Equation 3, we propose the *Averaged Diversity (AD)* as the second metric to evaluate any system where the diversity is among its specifications. The expression in 3 combines the Diversity at the k first positions, starting on 2 and going on until m, where m represents the total amount of different assets that are relevant to the query.

$$AveragedDiversity(AD) \equiv \frac{1}{m-1}\sum_{k=2}^{m} D(k) \quad (3)$$

The AD satisfies the quality of measuring 0 for homogeneous results, 1 for perfect diversity and, if a certain uniformity appears among the results, produces lower values when this uniformity is located among the first positions of the ranked list.

Both AP and AD were calculated for every text query and their values averaged among all topics to obtain the *Mean Average Precision (MAP)* and *Mean Averaged asset-Diversity (MAD)*.

4.3 Results

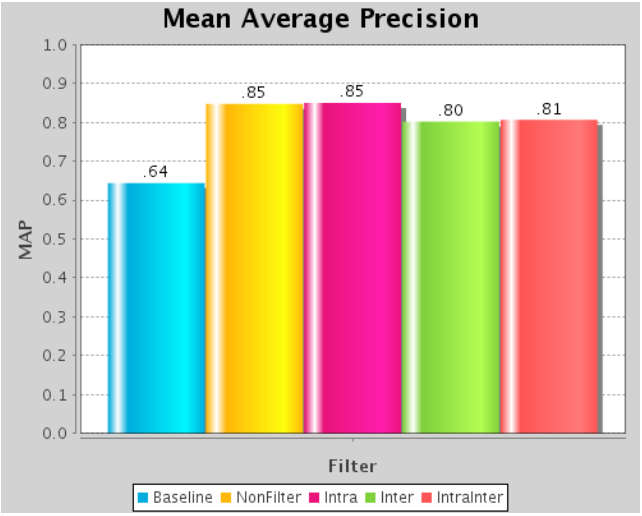


Figure 6: Mean Average Precision.

The previous solutions for the filtering of the SGs have been evaluated by considering the four presented options for ranking: non filter (only random walk), intra-asset filtering, inter-asset filtering and both types of filtering. An additional baseline case was considered by using the results list obtained after the text search, with no further processing.

The Mean Average Precision and Diversity represented in Figures 6 and 7 clearly show the relevance increase introduced by the random walk. The filtering of the SG has little impact in the MAP, with a slight decrease when the inter filtering is introduced. The decrease is reasonable as any filtering operation is an action against the principles of relevance estimation in the SG: the more relevant are the more connected and, by removing connections, there is a loss in the data used to estimate relevance. In compensation, Figure 7 proves that the filtering strategies increase the diversity of assets in the results. The removal of only inter-asset connections significantly decreases the MAP as it isolates groups of relevant keyframes whose score decreases in favour of other keyframes from their same asset. Nevertheless, the best results are obtained when the inter-asset filtering is combined with the intra-asset solution.

The results per query concept are presented in Figures 8 and 9 . The first conclusion from these figures is that the domain of application of the filtering techniques has an impact on the obtained results. While the general conclusion drawn from the MAP and MAD analysis apply, not all query concepts present the exact same behaviour. For example, the intra+inter filtering does not present the best AD in the "Formula 1" and "Accident" domains, although in general its behaviour is the most regular in terms of diversity.

The filtering stages require a computation effort to delete the edges in the SG but, on the other hand, they also simplify the computation of the iterative process of the random walk. The experimental measures shown in Table 2 point out that there is no generic conclusion about what the final impact of the filtering is. It has also been observed that, in case of combining both intra- and inter-filtering it

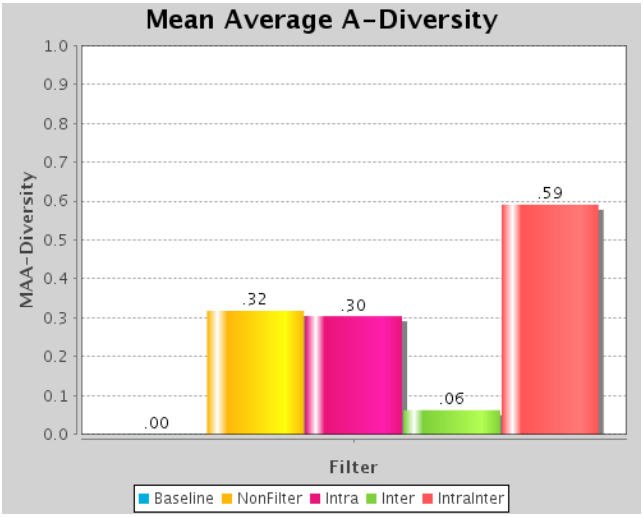


Figure 7: Mean Average Diversity.

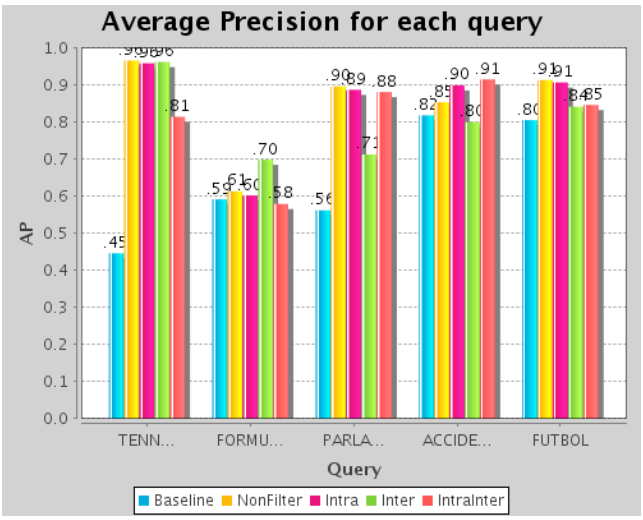


Figure 8: Average Precision for each query concept.

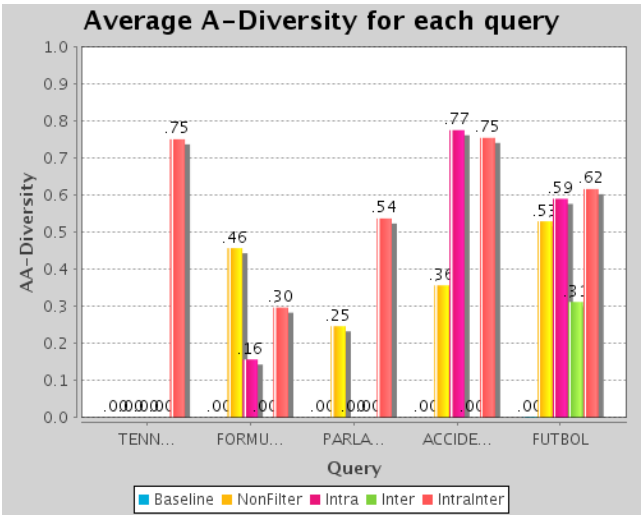


Figure 9: Average Diversity for each query concept.

Table 2: Computation time (in ms)

Query	NonFilter	Intra	Inter	InterIntra
T. Tennis	80,943	95,649	97,154	109,971
Formula 1	495,578	819,981	610,358	855,534
Parliament	575,740	846,400	724,752	977,864
Accident	1,277	806	734	737
Football	12,382	6,148	5,284	6,431

is advisable to perform the intra-case first because this step normally deletes more edges than the inter-case. When applied first, the intra-case reduces the amount of edges that must be checked for inter- and reduces the computation effort. Also for this reason, the obtained times for the inter & intra case do not correspond to the sum of times when applied separately.

5. CONCLUSIONS

This paper has presented a system architecture to implement a visual ranking solution based on the computation of the random walk algorithm over a SG. The query-dependent similarity graph has been filtered following to different strategies, intra- and inter-asset, in order to solve diversity problems of the basic technique when applied on a broadcaster archive.

The evaluation has proved the validity of the random walk approach to detect the relevance of the keyframes when the manual annotation is at the highest scale of the video assets. The repetition of certain shots in multiple assets of the archive can also be interpreted as an implicit annotation of the content in terms of relevance. This fact is exploited by the random walk algorithm to determine which keyframes are the most representative for the query.

The presented results prove that the filtering of the similarity graph is a valid technique to improve the asset diversity of the keyframe-based results. The experimentation suggests that the intra-asset filtering increases the diversity by itself, but that the inter-asset filtering only has a significant impact when combined with the intra-asset filtering. The filtering steps significantly increase the asset diversity with little impact on the relevance estimated by the random walk. The observations also show that the concept type and collection of assets have an impact on the overall performance, with great disparity if using intra or inter filtering separately. Nevertheless, when both filtering strategies are combined, results are more stable and generally better.

From the computation point of view, the introduction of the filtering stages may increase the required effort as well as decrease it by building a simpler SG, this is a query-dependent behaviour. Results suggest that the computation performance may decrease for large datasets but improve in small ones.

The ranking technique has been proven as a valid solution in the domain of the considered TV broadcaster. Nevertheless, best results are to be obtained when this technique is combined with other strategies such as keyframe clustering to increase diversity, relevance feedback to learn the visual descriptors weights, user preferences or repository history.

Diversity ranking can be understood as a method for exploiting the implicit content annotation. Firstly, it uses the repetition of content in the database to detect which keyframes are the most relevant. This principle is based on the assumption that relevant material will be more often reused by the video editors generating new material for the repository. Secondly, the organization of keyframes in assets is a second type of keyframes organization that is naturally generated by the keyframe extractor. The exploitation of implicitly generated data is a promising research line because it allows the improvement of the retrieval results without any further extra effort from the documentalists annotating the content.

Future work includes testing the technique in the case of visual queries, which will provide an initial score that will condition the random walk scores. Another promising direction is to explore the co-ranking by mutual re-enforcement between the different SGs build for every visual descriptor.

6. ACKNOWLEDGEMENTS

All images used in this paper belong to TVC, Televisió de Catalunya, and are copyright protected. They have been provided by TVC with the only goal of research under the framework of the BuscaMedia project.

This work was partially founded by the Catalan Broadcasting Corporation (CCMA) through the Spanish project CENIT-2009-1026 BuscaMedia:"Towards a Semantic Adaptation of Multinetworkd and Multiterminal Digital Media", and by the project of the Spanish Government TEC2010-18094 MuViPro:"Multicamera Video Processing using Scene Information: Applications to Sports Events, Visual Interaction and 3DTV".

7. REFERENCES

[1] L. Cao, A. Del Pozo, X. Jin, J. Luo, J. Han, and T. S. Huang. Rankcompete: simultaneous ranking and clustering of web photos. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 1071–1072, New York, NY, USA, 2010. ACM.

[2] W. H. Hsu, L. S. Kennedy, and S.-F. Chang. Video search reranking through random walk over document-level context graph. In *Proceedings of the 15th international conference on Multimedia*, MULTIMEDIA '07, pages 971–980, New York, NY, USA, 2007. ACM.

[3] Y. Jing and S. Baluja. Pagerank for product image search. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 307–316, New York, NY, USA, 2008. ACM.

[4] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub. Extrapolation methods for accelerating pagerank computations. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 261–270, New York, NY, USA, 2003. ACM.

[5] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the webs. *Stanford Digital Library Technologies Project*, November 1998.

[6] F. Richter, S. Romberg, E. Hörster, and R. Lienhart. Multimodal ranking for image search on community databases. In *Proceedings of the international conference on Multimedia information retrieval*, MIR '10, pages 63–72, New York, NY, USA, 2010. ACM.

[7] R. Yan, A. Hauptmann, and R. Jin. Multimedia search with pseudo-relevance feedback. In E. Bakker, M. Lew, T. Huang, N. Sebe, and X. Zhou, editors, *Image and Video Retrieval*, volume 2728 of *Lecture Notes in Computer Science*, pages 649–654. Springer Berlin / Heidelberg, 2003.

[8] T. Yao, T. Mei, and C.-W. Ngo. Co-reranking by mutual reinforcement for image search. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, CIVR '10, pages 34–41, New York, NY, USA, 2010. ACM.

[9] C. X. Zhai, W. W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 10–17, New York, NY, USA, 2003. ACM.

Annex II: *Contribucions en anglès al bloc BitSearch*

Monday, June 28, 2010

PageRank

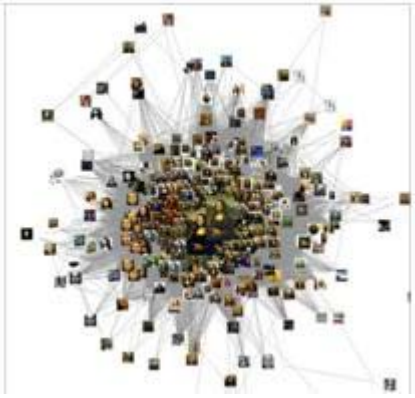
Hi! I'm Monica, Xavi introduced me on his last post. During this summer I will prepare for start my Bachelor Thesis and it will developed in a joint with CCMA, in September.

The PageRank algorithm, used by Google Internet search engine, assigns a numerical weighting to each element of a hyperlinked set of documents with the purpose of measuring its relative importance within the set.

Basically, it reflects the idea that a document or a web page ir importantif there many pages linking to it, and those pages are important themselves

It also considers how important is the web page which casts a vote because weigh more heavily and help to make other pages important.

There are many algorisms that are based on this concept, for example:



VisualRank: It considers images like web pages and its similarities like visual links. Trough an iterative procedure a numerical weight is assigned to each image for measure its relative importance to the other images being considered.

RankComplet: It's a new algorithm that generalizes the PageRank for the task of simultaneous ranking and clustering.

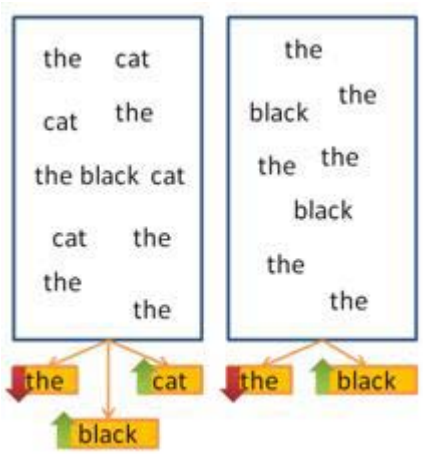
VisualRank is still not perfect because it doesn't consider the visual diversity of the retrieval results. Therefore, if two images are similar they will share the visual links themselves. And, if an image is ranked high its near duplicated images will also be ranked high, giving the user a subset of images with limited visual diversity.RankComplete can summarize the diversified images by performing the clustering techniques.

Tuesday, June 29, 2010

TF-IDF

The tf-idf *(term frequency-inverse document frequency)* is a weight often used to evaluate how important is a word to a document in a collection.

On one hand, the frequency a term appears in a document called **term frequency.** On the other hand, the **inverse document frequency** factor diminishes the weight of terms that occurs very frequently in the collections and increases the weight of terms occur rarely.



For example, suppose we have a set of text documents and we want to determinate which documents is most relevant to the query "the black cat". In order to distinguish them, we might sum the number of times each term ("the", "black" and "cat") occurs in each document.

But as the term "the" is so common, is not a good keyword to distinguish relevant and non-relevant documents. However, terms like "black" and "cat" that appear rarely are good keyword to distinguih important documents. Hence an IDF factor diminishes the weight of the term "the" and increases the weight of terms "black" and "cat".

Recently, tf-idf weighting has proven to be a very successful approach for image and particular object retrieval.And now I will explain some thesis where this weighting is applied.

Tag Ranking: The tags associated with an image generally are in a random order without any importance or relevance information. It propose a ranking scheme that classifies automatically the tags associated with a particular image based on its relevance to the content of the image.
To estimate tag's relevance are based on probability density estimation, using the IDF factor to penalize the tags that appears too frequently in the dataset, and then perform a random walk over the tags graph with the aim to boost the performance of tag ranking using relationships among tags.

Clustering the Topics using TF-IDF for model fusion:
It proposes a novel data fusion technique for combining the results os different information retrieval strategies according to user's queries.

It chooses the tf-idf weighting that allows them to cluster the training queries/topics from a collection. Then, they select the best weighting schemes for each cluster as a combination of the best scheme for each of the topics from the cluster. Later on it uses this feature to classify test topics into the appropiate cluster and run the corresponding weighting scheme.

Sunday, July 18, 2010

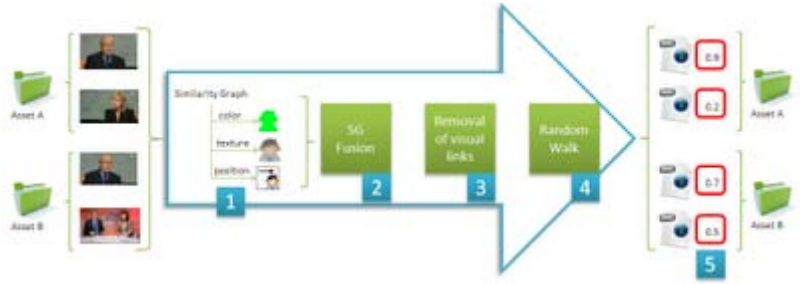
Scheme of reranking results for video search

I met Xavi and Ramon from CCMA-ASI to present my proposal for my thesis last Tuesday, I'll define my aim and structure of the algorithm that I'm working these days.

One of my objectives of my work focuses on **reranking results for video search based on visual similarity graph.**But, first of all I want to explain some concepts for you can understand this post:

- **Asset:** a set of video and audio. They are annoted at the level of asset therefore all frames of each asset have the same metadata
- **Keyframe:** representative images of the assets. So an asset is reprsentated by a set of keyframes.

As I said before, I'll explain the structure of the algorithm in general. So only I will show the three main blocks and I will describe them in detail in my next entries.



Step 1. I don't consider that this block form part of my thesis but is important to explain. Aida built an image similarity graph for each visual descriptor where vertices represent keyframes and edge weights are proportional to the visual similairty between two keyframes.

Step 2. Similarity Graph Fusion:

- Goal: merge the different similairty graphs to work only with a single graph.
- In: one similarity graph for each visual descriptor + weights for each visual descriptor.
- Out: only one similiarity graph.

Step 3. Removal of visual links:

- Goal: remove links in similarity graph in order to avoid visual similar keyframes among the best results.
- Normally, when an image search based in visual similarity is done, we can see in the best results only imges very similar. If we focus on the case of CCMA it's more interesting suggest this filtering based on the paper Multimodal Ranking for Image Search on Community Databases to avoid this effect.
- In: a similarity graph
- Out: a limited similarity graph.

Step 4. Random Walk:

- Goal: assign scores to (nodes) keyframes of similarity graph based on tamount of links and their visual distances. If you want to know more about this algorithm take a look at Xavi's post.

- In: a limited similarity graph + convergence threshold
- Out: score for each keyframe.

Two methods can be used to apply a random walk on the similarity graph.

- *QUERY-(IN)DEPENDENT method* attempt to measure the estimated importance of a keyframe on the similarity graph, independent of any consideration of how well it matches the specfic query. So I have to consider all of the keyframes in the database.
- *QUERY-DEPENDENT method* attempt to measure the estimated importance of a keyframe on the similarity graph based on the degree to which a keyframe matches a specific query. So I have to consider a limited number of keyframes which matches with the query.

We decided to implement the two methods for compare them empirically. But, nowadays I'm working in the first method, query-independent.

Step 5. Finally, I get an xml file for each keyframe with the final score.

That's all! During these weeks I'll program *Similarity Graph Fusion* and *Random Walk* blocks. In my next posts I'll tell you my progress with more details.

Thursday, August 5, 2010

Filtering similarity graph for reranking video search

A few weeks ago I explained the Scheme of reranking results for video search without going into details. For that reason, I would like to invite you to read my last post if you have still not done it and I'll explain one of the five blocks more extensively today, in particular the third block: *Removal of visual links*.

Step 3. Removal of visual links:

- Goal: remove links in similarity graph in order to avoid visual similar keyframes among the best results.

As I said this block is based on the algorithm found in the paper Multimodal Ranking for Image Search on Community Databases.

In this work they focus on the goal of selecting relevant images given a query term on a large-scale community database such as Flickr where images are often associated with different types of user generated metadata. Therefore, they select relevant images based on user's influence and multimodal similarity graph.

The importance of an image is assumed to be proportional to the number of images showing similar content. This concept is used by PageRank.

To represent the relationship between visual images they generate a visual similarity graph where each node represents an image and edge the similarity. Each link from one image to another represents a vote for the other image's relevance.

As they target the search in community databases where users upload their images, it is necessary to take special care to avoid artifacts introduced by users. In order to limit a user's influence, they apply two restrictions in the multimodal similarity graph.

Let me to show you the two restrictions of video archive from a TV broadcaster.

We address the problem of manual annotations at the asset scale when these annotations only refer to one or a few keyframes in the asset. For example, consider an asset of a news program where diversity of topics is diverse and their annotations has been performed at the asset scale.

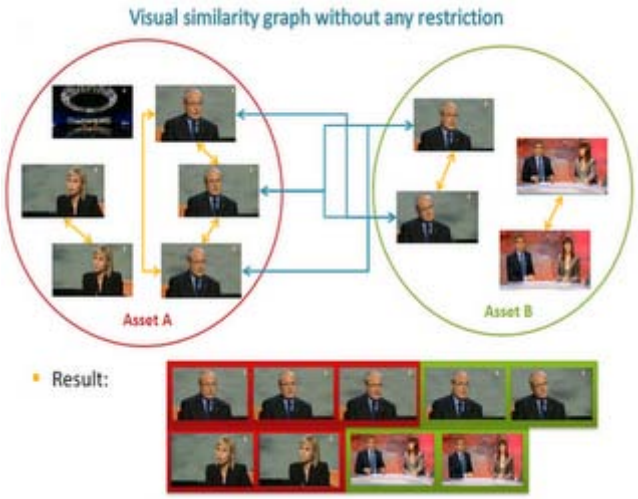
What would happen if any restriction hasn't been applied?

When an image search based on visual similarity is done, we can see in the best results only images very similar.

For example, if we make a query, for instance "Montilla" who is the current president of the Generalitat de Catalunya, in the database we will retrieve different assets, among them these two:

- Asset A: an interview with President Montilla, from the Catalan government.
- Asset B: two TV anchors talking about the interview with Montilla in the news. Therefore in the same asset appears duplicates from asset A.

If we base on a keyframe is important/relevance if they have a lot of keyframes showing similar content and we apply Random Walk without filter the similarity graph associated to all keyframes from assets which are annotated as "Montilla", in the top of ranked list will appear similar keyframes where appears "Montilla".



The problem is that have many similar keyframes from the same assset they don't give more information that could give only one keyframe so they are redundant and take up space on graphical interface. But, the users of archive look for assets and, therefore, they are interested in having results from different assets in the first positions of ranked list.

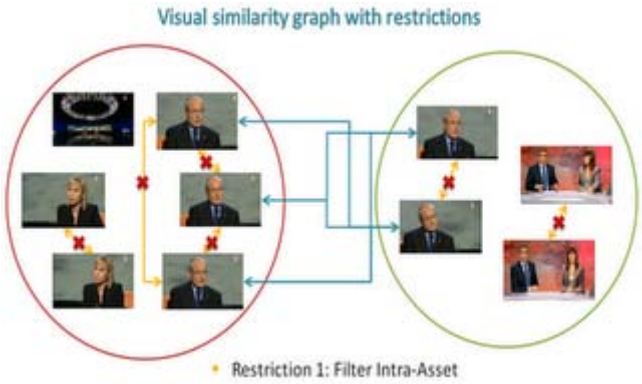
Moreover, the same asset can have more than one occurrence of the same concept and, in this case, we want to have a keyframe for each occurence in the best results.

What would happen if two restrictions have been applied?

It's time to propose the solution for the problem. That is the two types of filter, intra and inter asset.

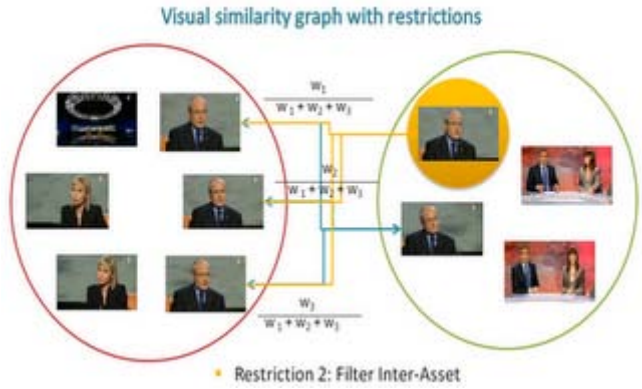
- **Filter intra asset**

An asset may not vote for any of his own keyframes. That is, no links between keyframes of the same asset are allowed as this would make the ranking vulnerable to manipulation by a single asset.

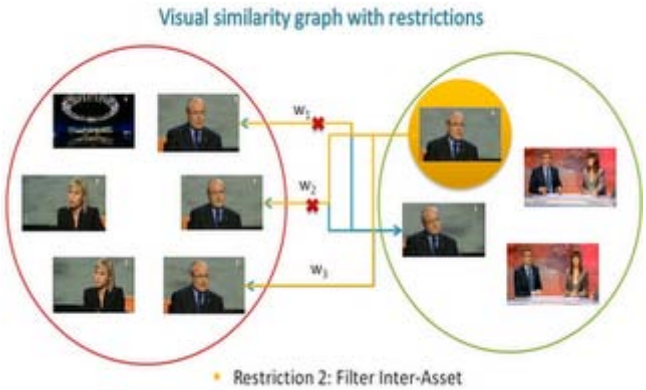


- **Filter inter asset**

If a keyframe, for example the keyframe from asset B with a yellow circle, has an incoming link from more than one keyframe of a particular asset, in this case from asset A, the respective link weights are normalized by the number of incoming links originating from that assets keyframes.



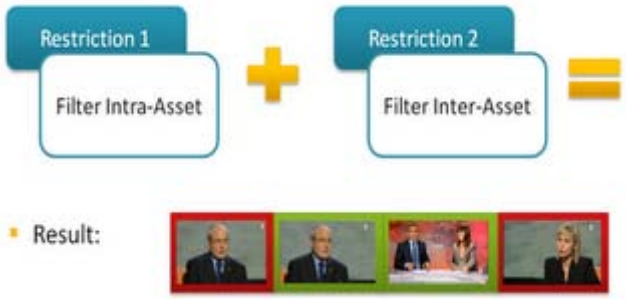
Alternatively we could keep only the in-link from the best matching keyframe.



Ideally, there won't be similar images of the same asset and it will be reduced the number of near duplicated images from different assets.

In addition, the images will be reranker according its relevance without any influence by reducing the visual similarity graph with this technique.

This could be the result for the query "Montilla".



Wednesday, August 18, 2010

Evaluation of a reranking video search algorithm

Today, I will talk about how I am going to evaluate the results from the Reranking video search algorithm These results are keyframes which has been found after a text-query. The keyframes are annotated at the level of video asset, therefore all frames in the same video have the same metadata.

What do we expect from the reranking algorithm?

We would like to achieve two things:

- That it includes keyframes from many different assets early in the ranking instead of include many keyframes that redundantly cover the same assets. So we expect diversity assets in the best results.

- That the visual content of the keyframes, in the top ranked results, will be relevance for the text-query.

Measures

I base on the methods of evaluation from the papers I mentioned in my last posts to choose the right measure to evaluate it.

Since we want to evaluate these two things, asset diversity and relevance keyframe, we have two families of measures:

1. S-recall for *asset diversity*

S-recall was used by Cao et al (Urbana-Champaign / Kodak) for evaluating the performance of the RankCompete algorithm in their experiments of visual reranking on ImageClef2008 dataset. This measure had been defined by Zhai, Cohen and Lafferty (Urban-Champaign / Carnegie Mellon) for evaluating the diversity of subtopics in text document retrieval.

Definition: consider a topic *T* with *na* subtopics *A1, ... , Ana* and a ranking *d1, ... , dm* of documents. Let *subtopics(di)* be the set of subtopics to which *di* is relevant. It defines the subtopic recall (S-recall) at rank K as the percentage of subtopics covered by one of the first K documents:

$$S - recall\ at\ K = \frac{\left|\bigcup_{i=1}^{K} subtopics(d_i)\right|}{n_A}$$

In our case a subtopic correspond to a video asset and one keyframe is always associated to one and only one asset.

2. Precision and Recall *for keyframes relevance*

These two concepts were previously defined in by Laura in this post. In order to calculate them, we will need to generate an annotation of the dataset at the keyframe scale. We will combine the annotations at the video scale with the manual annotation tools provided by GAT to perform this task. Once we have generated the ground truth, we will be able to measure the Mean Average Precision (MAP) as an evaluation of the retrieved keyframes relevance.

Experiments

A few weeks ago, we defined the experiments to examine the influence of several parameters in our keyframe reranking.

These are the goals which we set for the evaluation:

- Study the influence of the number of nearest neighbours *(k)* used to establish the link structure in similarity graph.

- The vector which has the scores obtained by intra-asset random walk, allows assigning some nodes a higher importance prior to the actual ranking procedure in the Random Walk block. Thus we examine two different vector settings in our experiment:

1. A uniform vector where we assign the same value to all nodes.
2. A non-uniform vector where the values correspond to a simple initial estimate of the importance of each keyframe.

- Study the influence of the weight factor leveraging visual and textual search and the convergence threshold which are the number of iterations in random walk.

- Compare filter intra-asset and/or inter-asset filtering. This point is the most important and novel of the algorithm.

- Once the optimal parameter setting is determined, we will compare query-dependent to query-(in)dependent approaches.

Friday, August 20, 2010

Scheme of query-dependent case

I explained the shceme of reranking video search in one of my last entries. I explained it in general and thinking, mostly, in the query-(in)dependent case, but I did not tell you how the query-dependent case can modify the scheme.

Step 1.

A non-ranked list, as no individual score is associated to each keyframe, is obtained by a query, for example, "Montilla" who is the current President from Catalan government.

Step 2. Similarity Graph

One similarity graph *(no intra-asset links)* is built from these keyframes results. Unlike the last scheme, there is no need to build the visual similarity graph for each descriptor and merge them later. Since the weights of the descriptors are set, we can compute a single graph similarity.

Step 3. Removal of visual links

As I said, in one of my last entries, there are two filters that can be applied, intra-asset and inter-asset.

The intra-asset filtering isn't made after building the visual similarity graph, as the query-(in)dependent. But it is performed **before** building the similarity graph because the calculation of similarity links which will be removed later is a waste of time.

Step 4. Random Walk

It assign scores to keyframes of similarity graph based on amount of links and their visual distances. There aren't changes in this block.

Step 5. Reranked list

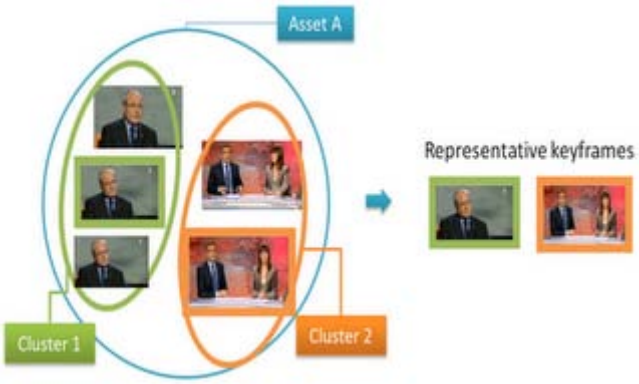
Finally, we get a new reranked list of the initial non-ranked results.

Friday, September 17, 2010

New approach for Reranking algorithm

These last days, we have been focusing in a new approach for the Reranking algorithm. In the last entries, two objectives had been defined. This is an overview of the approaches that have been defining.

1. Diversity of assets in the best results.
2. Relevant keyframes for the text-query.
3. **NEW!** Show only a representative keyframe for each cluster of similar keyframes from the same asset.



Now, I will explain what we want to achieve, what is the problem and what is the solution that is being taken for the third approach.

WHAT DO WE WANT?

That is, for a text query, for instance Montilla who is the actual president of the government of Catalonia, these could be the results for each case:

Without applying the Reranking algorithm

With applying the Reranking algorithm

As you see the two first keyframes are very similar and belong to the same asset. This is undesirable because we would like to obtain diversity of assets.

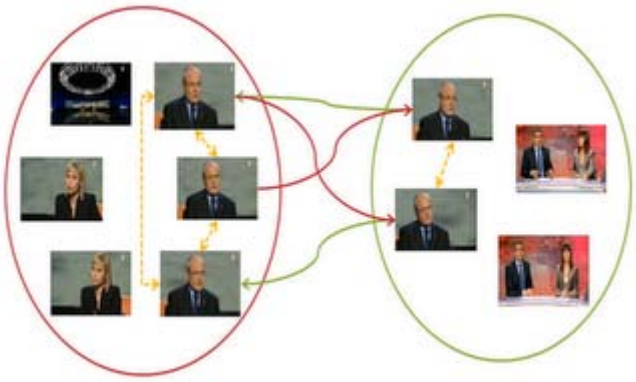
With applying Reranking and Cluster algorithm

That is what we want, only one representative keyframe for each cluster of similar keyframes from an asset.

NOTE: The color of the edge of keyframes represents to the same asset.

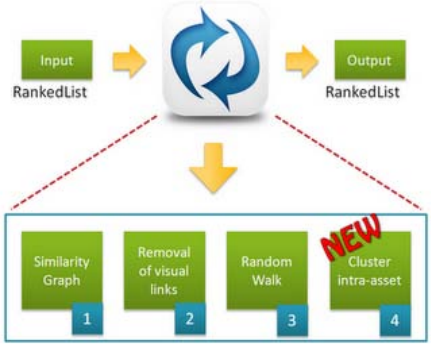
WHAT IS THE PROBLEM?

The new problem is that although filtering intra asset is performed to remove the links of the keyframes of the same asset. These can be connected indirectly through inter-asset edges, as you can see in the figure below. And, therefore we wouldn't get only one representative keyframe of the same cluster of visual keyframes from an asset.



WHAT IS THE SOLUTION THAT IS BEING TAKEN?

The solution that we adopted and Jaume is studying and implementing is creating clusters of similar keyframes, as you have seen in the first figure.
So this is the final scheme of Reranking algorithm.



If you would like to know more about the clusters algorithm I encourage you to follow Jaume's post.

Sunday, January 2, 2011

Relevance and Diversity Evaluation of the Ranking Algorithm

These days I'm finishing my Bachelor Thesis. In this post I will expose the results we obtained from de reranker algorithm.

Evaluation

The reranker algorithm for filtering the SGs at the intra and inter-asset level were tested on a representative dataset from CCMA, the public national broadcaster in Catalonia. The obtained ranked lists were evaluated in terms of keyframe relevance and asset diversity for different query topics and the generated resutls compared.

Experiments

We selected a set of text queries form a list of controlled terms used by the documentalists that manually annotated the video assets. For every query, all retrieved keyframes were individually annotated as the relevant or not-relevant for the query. The criterion to establish the relevance was to consider if the keyframe may be tagged with textual query by an annotator who would only access keyframe. The text queries were:

Query	# assets	# KFs
Table tennis	3	1,116
Formula 1	6	3,441
Parliament	12	2,816
Accident	8	66
Football	16	416

Metrics

These experiments pursues the generation of results that accomplish two basic properties: relevant keyframes and diversity of assets. Two different metrics were used to evaluate these two qualities: the average precision and the average asset recall.

$$Averaged\ Precision(AP) \equiv \frac{1}{m}\sum_{k=1}^{m} Precision(k)$$

Here, I present a new metric that is specifically designed to evaluate the reranker algorithm in the case of the corpus of CCMA.

Average Asset-Diversity

The goal was to desgn a metric that behaved similarly to the AP, that is, a normalized value whose best output were the unit and that would introduce a larger penalization to the non-

diverse results when they occur amount the earliest positions than when they occur in the latest ones. In a first approach, the Diversity at k would measure the variety of the results as:

$$Diversity\ at\ k\ \equiv\ D(k) = \frac{d(k) - 1}{k - 1}$$

where d(k) corresponds to the amount of different video assets contained in the positions 1...k of the ranked list. Notice that this metric is only defined for k greater or equals 2 as the diversity can only be evaluated on asset of multiple items.

Combining the concept of AP with the diversity measure introduced, we proposed the Average A-Diversity (AD) as the second metric to evaluate any system where the diversity is among its specification. The next expression combines the Diversity at the k first positions, starting on 2 and going on until m, where m represents the total amount of different assets that are relevant to the query.

$$AveragedDiversity(AD)\ \equiv\ \frac{1}{m-1}\sum_{k=2}^{m} D(k)$$

Both AP and AD were calculated for every text query and their values averaged among all topics to obtain the Mean Average Precision (MAP) and Mean Average Asset-Diversity (MAD).

Results

The previous solutions for the filtering of the SGs have been evaluated by considering the four presented options for ranking: non filter (only random walk), intra-asset filtering, inter-asset fitlering and both types of filtering. An additional baseline case was considered by using the resutls list obtained after the next search, with no further processing.

The MAP and MAD represented in the figures 1 and 2 clearly show the relevance increase introduced by the random walk. the filtering of the SG has little impact in the MAP, with a slight decreased when the inter filtering is introduced. The decrease is reasonable as any filtering operation is an action against the principles of relevance estimation in the SG: the more relevant are the more connected and, by removing connection, there is a loss in the data used to estimated relevance. In compensation, figure 2 proves that the filtering strategies increase the diversity of assets in the results. The removal of only inter-asset connections significantly decreases the MAP as it isolates groups of relevant keyframes whose score decreases in favor of other keyframes from their same asset. Nevertheless, the best results are obtained when the inter-asset filtering is combined with the intra-asset solution.

Figure 1

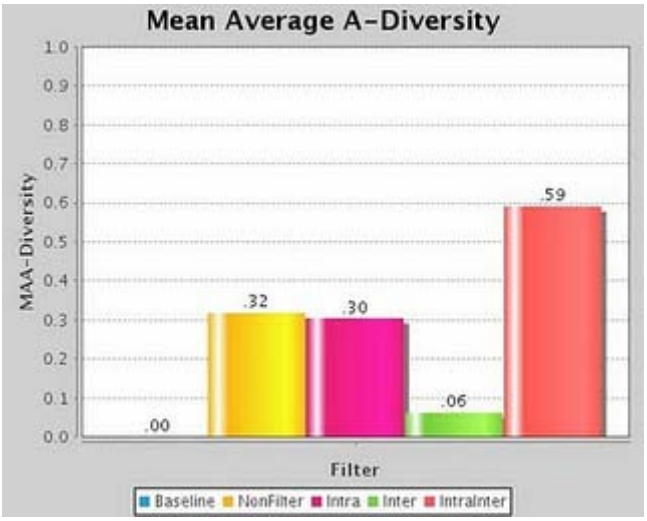


Figure 2

The resutls per query concept are presented in figures 3 and 4. The first conclusion from these figures is that the domain of application of the filtering techniques has and impact on the obtained results. While the general conclusion drawn from the MAP and MAD analysis apply, not all query concepts present the exact same behaviour. For example, the intra+inter filtering does not present the best AD in the "Formula1" and "Accident" domains, althought in general its behaviour is the most regular in terms of diversity.

Figure 3

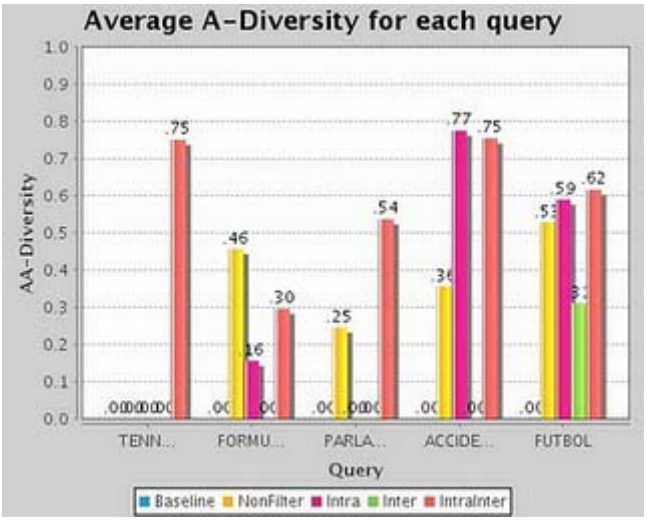


Figure 4

The filtering stages requiere a computation effort to delete the edges in the SG but, on the other hand, they also simplify the computation of the iterative process of the random walk. The experimental measures shown in the table 1 point out that there is no generic conclusion about what the final impact of the filtering is. It has also been observed that, in case of combining both intra and inter filtering it is advisable to perform the intra case first because this step normally deletes more edges than the inter case. When applied first, the intra case reduces the amount of edges that must be checked for inter and reduces the omputation effort. Also for this reason, the obtained times for the inter and intra case do not correspond to the sum of times when applied separately.

Query	NonFilter	Intra	Inter	InterIntra
T. Tennis	80,943	95,649	97,154	109,971
Formula 1	495,578	819,981	610,358	855,534
Parliament	575,740	846,400	724,752	977,864
Accident	1,277	806	734	737
Football	12,382	6,148	5,284	6,431

Table 1. Computation time (ms)