



# Extensió d'una interfície de cerca d'imatges a les consultes amb regions

Per M<sup>a</sup> Pia Muñoz Trallero

Tutors:

Xavier Giró

Xavier Vives



# Resum

Aquest és un projecte universitat-empresa que presenta una interfície de cerca multimodal a partir de la integració de diferents eines ja existents per crear una eina global que combina les cerques textuais amb les consultes mitjançant exemple, ja sigui una consulta basada en una imatge o en una regió d'aquesta imatge. L'eina global s'integra en l'entorn del Digiton, el gestor de continguts audiovisuals de la Corporació Catalana de Mitjans Audiovisuals. La principal aportació del present projecte consisteix en la creació d'una interfície de cerca d'imatges basada en regions i la seva integració en l'eina global.

## **Agraïments:**

Aquesta feina ha estat fruit d'un esforç conjunt. És per aquest motiu que hi ha molta gent que mereix ser mencionada en aquestes línies.

En primer lloc, vull donar les gràcies al Xavier Giró i al Xavier Vives per haver estat un recolzament molt pròxim durant aquests mesos, la qual cosa m'ha permès avançar sense dificultats i sense dubtes sent conscient que, en tot moment, hi havia algú que em guiava en la meva feina.

En particular, vull felicitar al Ramon Salla i la Irene Zeller pel seu treball i també agrair la seva col·laboració i ajuda constants que m'han permès aprendre molt ràpidament els llenguatges de programació emprats i fer molta feina i molt ben feta.

Per últim, voldria agrair el treball de la Neus Camps i d'en Manel Martos, que han estat els responsables de la implementació del servei web ofert per la UPC que capacita la interfície per donar uns resultats en les cerques. En concret, la Neus ha estat una col·laboració molt activa en tota la part de processat de la imatge. Per això també he d'agrair la seva disponibilitat i amabilitat en tot moment.

# Índex

1	Introducció.....	4
2	Estat de l'art.....	7
2.1	Descriptors visuals.....	9
2.2	Cercadors que incorporen consultes basades en el contingut visual .....	10
2.2.1	IMARS.....	11
2.2.2	CALIPH & EMIR.....	12
2.3	Segmentació interactiva.....	14
2.3.1	Marcadors.....	15
2.3.2	Propagació dels marcadors.....	15
2.3.3	Comparativa de interfícies gràfiques .....	26
2.3.3.1	Entorn d'escriptori.....	26
2.3.3.2	Entorn web.....	31
3	Requeriments.....	33
3.1	Requeriments tècnics.....	33
3.1.1	Descripció dels elements existents.....	33
3.1.2	Integració i millores.....	37
3.2	Requeriments d'entorn.....	39
3.3	Requeriments funcionals.....	39
4	Disseny.....	40
4.1	Eina global.....	40
4.2	Digitrace.....	41
4.2.1	Segmentació interactiva mitjançant BPT.....	42
4.2.1.1	Marcadors.....	42
4.2.1.2	Criteris de propagació d'etiquetes.....	44
4.2.1.3	Estudi dels marcadors i els criteris de propagació.....	45
4.2.2	Interfície gràfica.....	52
4.3	Comunicació.....	54
4.3.1	Arquitectura distribuïda.....	54
4.3.2	Servei web.....	56
5	Desenvolupament.....	57
5.1	Entorn de desenvolupament.....	57
5.1.1	Marc de desenvolupament: Flex.....	57
5.1.2	Entorn: Adobe Flash Builder.....	59
5.1.3	Model multi-capa.....	59
5.2	Arquitectura.....	60
5.2.1	Protocol HTTP.....	61
5.2.1.1	Sintaxi d'una URL per HTTP.....	62
5.2.1.2	Mètodes HTTP.....	62
5.2.2	Arquitectura REST.....	63

5.2.3 Model client – servidor dins la CCMA.....	64
5.2.3.1 Crides HTTP client – servidor.....	64
5.2.4 Model client – servidor entre la CCMA i la UPC.....	69
5.2.5 Implementació amb Flex: HTTP Service.....	69
6 Resultats.....	71
6.1 Estudi comparatiu dels algorismes de segmentació interactiva.....	71
6.1.1 Avaluació de les tècniques de SRG, SIOX i BPT.....	71
6.1.2 Avaluació dels modes rectangle i traça.....	74
6.2 Millores al Digimatge.....	81
6.2.1 Digitrace.....	85
6.3 Integració en el sistema.....	91
7 Conclusions.....	95
7.1 Grau d'assoliment dels requeriments.....	95
7.2 Punts forts de la interfície.....	96
7.3 Conclusions tècniques.....	96
7.4 Conclusions personals.....	98
7.5 Treball futur.....	99
8 Bibliografia.....	101
Annex.....	103
Annex I: Exemples de segmentació interactiva.....	104
Annex II Artícles del Bitsearch.....	110
Annex III: Fonts d'informació.....	124

## 1 Introducció

La digitalització dels continguts audiovisuals ha donat lloc a un augment exponencial de la quantitat d'informació que s'emmagatzema diàriament als repositoris de les empreses del sector audiovisual. Aquest fet ha derivat en un interès creixent per la investigació en l'àmbit de la indexació de vídeos.

Actualment la gran majoria de recuperacions d'imatges i vídeos dins d'aquests grans repositoris es fa mitjançant cerques textuais contra una base de dades. Tot i així, cada cop s'evidencia més la necessitat de nous mètodes de cerca per oferir a l'usuari una major agilitat, flexibilitat i eficàcia que complementin els descriptors textuais. Avui en dia, aquesta tasca que rep la major part dels esforços en la investigació per a la cerca i recuperació d'informació audiovisual. Una de les noves vies de cerca proposa tècniques de processat de la imatge per poder fer cerques sobre la pròpia imatge mitjançant criteris de similitud visual. El valor principal d'aquest tipus de cerca resideix en que no depèn de cap anotació ni interacció i, a més, permet fer cerques sobre conceptes difícils d'expressar en paraules.

No podem oblidar que, per dur a terme aquesta millora, es requereix de noves interfícies d'usuari que permetin presentar les eines de processat de la imatge en un entorn visual senzill. Amb aquesta finalitat va néixer el projecte de final de carrera d'en Ramon Salla [1], precedent d'aquest projecte : Aplicació Rica d'Internet per a la consulta amb text i imatge al repositori de vídeos de la Corporació Catalana de Mitjans Audiovisuals<sup>1</sup>. La seva principal motivació va ser "*la recuperació d'imatges a partir d'una idea*", un concepte que es manté i s'estén en el present treball.

La Corporació Catalana de Mitjans Audiovisuals (CMMA) és una empresa catalana puntera en el camp dels mitjans de comunicació destinada al servei públic amb l'objecte de proporcionar informació i entreteniment amb la màxima qualitat possible. La CCMA

---

1 <http://www.ccma.cat>

està formada per un conjunt d'empreses que ocupen diferents àmbits a dins del sector audiovisual: Televisió de Catalunya, Catalunya Ràdio, CCRTV Interactiva, CCRTV ASI i Activa Multimèdia Digital.

La interfície desenvolupada pel Ramon Salla a la CCRTV ASI, el Digimatge, permet afegir a les cerques textuais la cerca per imatges similars i combinar-les. Després d'aquest primer pas, sorgeix la necessitat d'una cerca més acurada: la cerca per regions, la qual s'avança com a treball futur a les conclusions del projecte esmentat; així com la utilització d'un ventall més ampli de característiques visuals, que en la versió inicial del Digimatge quedava reduïda al color.

Aquest projecte de final de carrera, doncs, té com a finalitat l'assoliment d'aquestes mancances i la incorporació dels resultats a la pròpia interfície del Digition, l'eina amb la que compta la CCMA i, en concret, la Televisió de Catalunya, per fer les cerques textuais del material audiovisual digitalitzat i emmagatzemat al seu repositori de dades.

Per tant, l'objectiu primer del projecte consisteix en el desenvolupament d'una interfície gràfica d'usuari (*GUI*) que permeti fer una selecció de la regió d'interès dintre d'una imatge exemple a partir d'uns marcadors, de manera que l'usuari sigui capaç d'escollir aquesta regió, interactuant amb la màquina de manera senzilla, intuïtiva i eficaç.

Aquesta primera fita ha evolucionat cap a una eina més global aprofitant diferents aplicacions que la CCMA ja té en funcionament. En conseqüència, l'objectiu final esdevé la incorporació d'aquestes eines, juntament amb la interfície de cerca basada en regions desenvolupada, en el context del Digition per tal de proveir a l'usuari de majors possibilitats de cerca.

Tot això esdevé possible gràcies al suport i la col·laboració del grup de recerca de processat de la imatge de la Universitat Politècnica de Catalunya, el qual ens proporciona les eines de processat sobre les quals treballem.

Aquest és un projecte de col·laboració universitat – empresa situat en el marc d'un



projecte de recerca estatal: l'i3media. L'objectiu principal de l'i3media se centra en la creació i gestió automatitzada de continguts audiovisuals intel·ligents. En aquest context, la interfície global presentada permet la gestió automatitzada de continguts audiovisuals més enllà dels clàssics descriptors textuais aconseguint una cerca intel·ligent d'imatges a partir, únicament, del contingut visual d'una imatge o una regió. En definitiva, aquesta interfície global ofereix un sistema de cerca multimodal que s'inicia amb una consulta textual permetent utilitzar els resultats d'aquesta per noves consultes basades en el contingut d'una imatge o regió.

## 2 Estat de l'art

El projecte d'en Ramon Salla presenta una interfície d'usuari que introdueix conceptes molt valuosos com la consulta mitjançant exemple (*Query by Example*) i els descriptors visuals.

La consulta mitjançant exemple és una tècnica de consulta en que es proveeix d'una imatge exemple a un sistema CBIR (*Content-based Image Retrieval*) donant com a resultat imatges amb característiques similars a la imatge exemple segons uns criteris de cerca preestablerts. Aquest sistema CBIR es basa en el contingut visual de la imatge exemple. Com a conseqüència, aquesta tècnica no es limita exclusivament a imatges completes sinó que es pot limitar a parts d'una imatge. Aquesta fou la principal aportació d'aquest projecte precedent.

L'avaluació de la similitud entre les imatges o parts de la imatge es realitza a partir d'unes metadades extretes automàticament anomenades *descriptors* visuals. Els descriptors visuals són els encarregats de descriure les característiques elementals que definiran la imatge (el color, la textura, la forma, el moviment, la localització, la dispersió,...) en tot aquest sistema.

A la Figura 1 es mostra un esquema complet d'un sistema CBIR per a una consulta per regió. El procés es divideix en tres fases:

- La **ingesta** de totes les imatges que conformaran la base de dades amb la consegüent segmentació de cadascuna d'elles. Aquesta segmentació consisteix en dividir automàticament la imatge en regions més petites. A continuació s'extreuen una sèrie de descriptors visuals que defineixen cada regió mitjançant característiques de baix nivell com, per exemple, la forma o el color.
- La **segmentació interactiva** consisteix en seleccionar la regió que es vol buscar a dins d'una imatge. En aquest moment entra en joc l'usuari, ja que és l'encarregat

de interactuar amb una interfície que permeti extreure, d'alguna manera, un objecte de dins d'una imatge.

- La cerca consisteix en la comparació de tots els descriptors emmagatzemats a la base de dades amb els descriptors de la regió seleccionada. Per tant, hi ha un pas previ de càlcul dels descriptors de la regió, com s'observa a la Figura 1. Els resultats d'aquesta comparació s'expressen amb una distància, que permet generar un llistat ordenat de les imatges de la base de dades que contenen regions similars a la de consulta.

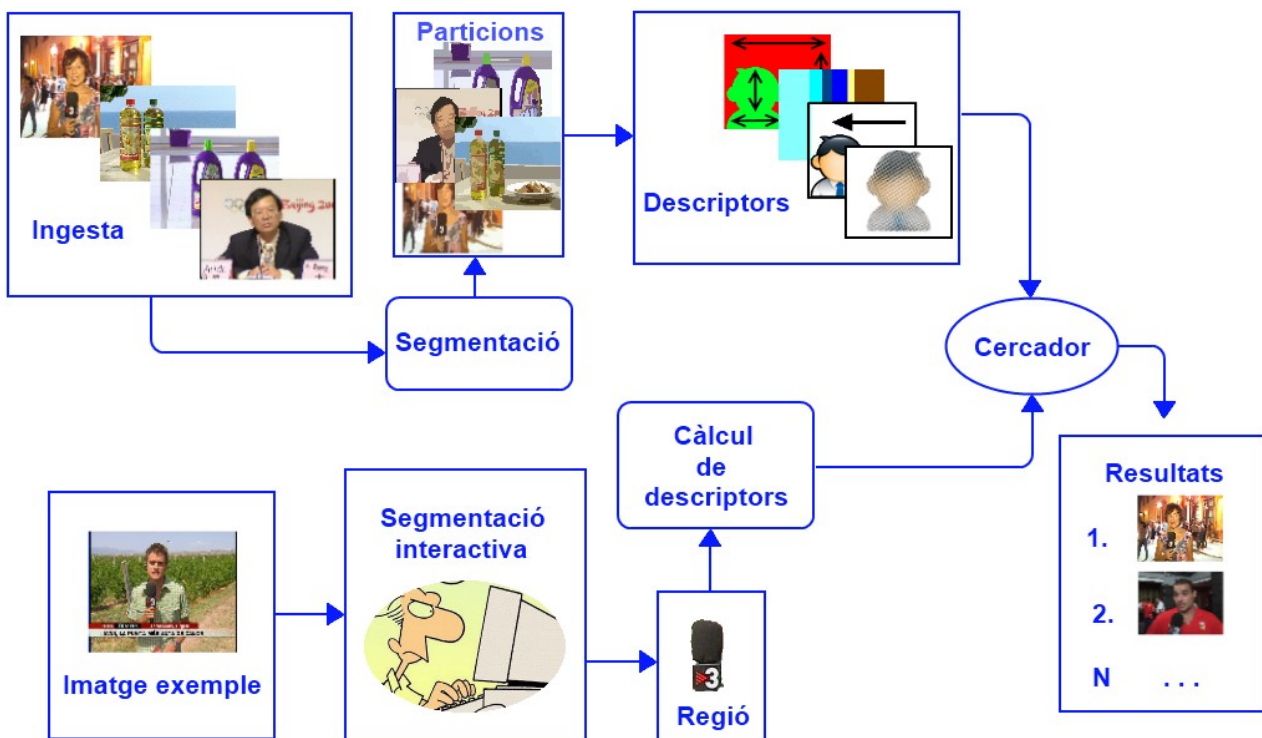


Figura 1: Sistema CBIR

En aquest apartat d'estat de l'art es fa un estudi d'aquestes tres fases. Per començar, s'introdueixen els descriptors visuals. A continuació, s'exploren algunes de les tècniques que permeten dur a terme l'extracció d'objectes en una imatge. I per últim, s'avaluen dues eines de l'estat de l'art que permeten fer consultes d'imatges a dins d'una base de dades basant-se en el contingut visual d'una imatge exemple.

## 2.1 Descriptors visuals

La indexació audiovisual consisteix en la generació de metadades sobre els continguts audiovisuals per poder ser recuperats a partir d'aquesta informació, que s'emmagatzema juntament amb els continguts audiovisuals. Existeixen moltes classes de descriptors per aquests continguts, des dels descriptors textuais fins als descriptors visuals passant pels descriptors d'àudio. En concret, els descriptors visuals són els encarregats d'extreure les característiques de baix nivell d'una imatge, com poden ser el color, la textura, la forma, la dispersió, etc.

A més, dins dels descriptors visuals es pot fer una distinció entre els descriptors basats en una imatge sencera i els que són específics per a regions.

Hi ha una gran varietat de descriptors visuals. Per això, per tal de facilitar la interoperabilitat entre sistemes, l'estàndard MPEG-7 [11] va crear un conjunt de descriptors que són, concretament, els que utilitza el nostre sistema [10][12]. A continuació es descriuen els més rellevants en el context del *Content-based image retrieval*.

– **Descriptors visuals específics per a imatges:**

- **Color Layout Descriptor** : aquests descriptor se centra en la disposició espacial dels colors a dins de la imatge, per tant, serà de gran utilitat quan es vulgui trobar imatges amb una disposició del color semblant.
- **Color Structure Descriptor**: aquest descriptor avalua com de compactes o dispersos estan els colors presents en la imatge.

– **Descriptors visuals per imatge i regió:**

- **Texture Edge Histogram Descriptor** : aquest descriptor no es basa en cap característica relacionada amb el color de la imatge sinó que estudia la distribució dels contorns (horitzontal, vertical, diagonal o no direccional) a cada porció de la imatge.

- **Dominant Color Descriptor** [13]: la finalitat d'aquest descriptor és trobar imatges que comparteixin els mateixos colors predominants amb la imatge exemple i amb una representativitat semblant.
  - **Scalable Color Descriptor** : aquest descriptor estudia l'histograma de color de la imatge en l'espai HSV.
- **Descriptors visuals específics per regions:**
- **Shape CSS**: Representa els punts d'inflexió del contorn de l'objecte seleccionat segons la seva posició a dins del contorn i el seu grau de curvatura sense tenir en compte la mida o el posicionament de l'objecte.
  - **Posició**: centre de masses de la regió respecte la imatge.
  - **Mida**: proporció de la regió respecte la imatge.<sup>2</sup>

## **2.2 Cercadors que incorporen consultes basades en el contingut visual**

Actualment, els cercadors que incorporen consultes basades en el contingut d'una imatge o un vídeo, compten amb dues eines diferenciades:

- L'**indexador** que analitza tota la col·lecció fent un estudi previ del contingut visual de cada element a partir de diferents descriptors visuals.
- El **cercador** que proveeix a l'usuari d'una interfície que li permet interactuar amb el sistema i navegar per la col·lecció, extreure informació i fer consultes.

A continuació es presenten dues propostes que ofereixen un sistema complet que permet des de la indexació d'una col·lecció fins al seu accés mitjançant consultes basades en el contingut visual, entre d'altres.

---

<sup>2</sup> Aquest descriptor no pertany a l'estàndard MPEG-7 però també s'utilitza en la nostra proposta.

### 2.2.1 IMARS

L'IBM Multimedia Analysis and Retrieval System<sup>3</sup> [2] és un sistema d'indexació, classificació i cerca pensat per a grans col·leccions d'imatges i vídeos digitals. IMARS analitza les característiques visuals de les imatges i els vídeos per organitzar-los automàticament i poder fer cerques basades en el seu contingut visual.

L'eina de classificació de l'IMARS extreu dues classes d'informació visual: per una banda, extreu un conjunt de descriptors visuals de cada imatge i, per l'altra banda, utilitza aquests mateixos descriptors per aplicar tècniques de classificació que siguin capaces de decidir de forma automàtica si un nou vídeo o imatge pertany a una categoria semàntica predeterminada.

Mitjançant els descriptors, aquesta eina és capaç d'identificar imatges i vídeos duplicats o molt semblants i agrupar les imatges per similitud visual creant subgrups en la col·lecció inicial i presentant-los visualment a l'usuari. Aquesta funcionalitat es coneix com a agrupament d'imatges (*clustering*) .

El *clustering*, com a concepte general, és un procediment d'agrupament d'elements que comparteixen una sèrie de trets en diferents grups particionant tota la informació en subconjunts o *clusters*. L'agrupament d'imatges, per tant, consisteix en la creació de subgrups d'imatges que comparteixen alguna característica comuna. L'IMARS agrupa les imatges segons la similitud únicament visual entre elles.

L'IMARS també permet crear etiquetes per les imatges, a partir de les quals l'usuari crearà categories a dins de la col·lecció. Mitjançant les tècniques d'entrenament comentades anteriorment, l'eina classifica automàticament els vídeos i imatges com a pertanyents o no a una categoria semàntica predeterminada i les mostra de forma ordenada.

La semàntica estudia el significat o la interpretació d'un determinat element. En aquest cas, ens referim a categoria semàntica com a un conjunt de característiques

---

<sup>3</sup> <http://www.alphaworks.ibm.com/tech/imars>

semàntiques compartides per una col·lecció d'imatges que permeten etiquetar aquesta col·lecció amb un significat global, com pot ser 'primavera', 'persona' o 'futbol'.

Per últim, l'IMARS permet dur a terme consultes mitjançant text i consultes per imatges semblants mitjançant una o més imatges de partida i els seus corresponents descriptors.

Com s'observa a la Figura 2, a la mateixa pàgina que mostra els resultats d'una cerca es poden veure uns enllaços a sota de cada imatge. Aquests enllaços són tres: el primer permet fer una cerca d'imatges similars, el segon fa una cerca d'imatges relacionades segons la informació recaptada de la imatge a nivell textual i el tercer afegeix la imatge a la consulta anterior.

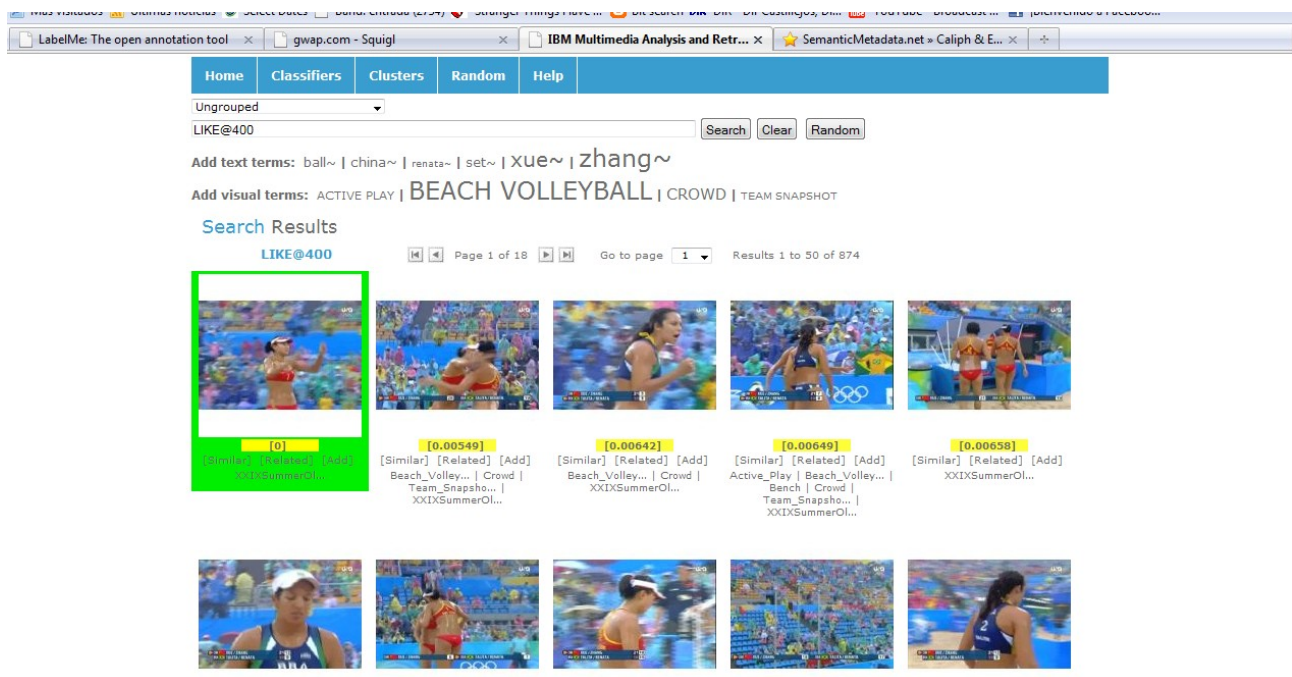


Figura 2: Interfície de l'IMARS

## 2.2.2 CALIPH & EMIR

Caliph & Emir <sup>4</sup>[3] és un prototipus per anotació i recerca d'imatges digitals basat en

<sup>4</sup> <http://www.semanticmetadata.net/features/>

l'estàndard MPEG-7. És un projecte de codi lliure i va guanya l'ACM Multimedia Open Source Competition de 2009<sup>5</sup>.

L'eina d'indexació, el Caliph, extreu diferents descriptors visuals de la imatge extrets de l'estàndard MPEG-7. Aquests són: Dominant Color, Color Layout, Scalable Color i Texture Edge Histogram.

El Caliph també permet l'anotació manual de metadades textuais i l'anotació i classificació d'imatges en categories semàntiques de la mateixa manera que l'IMARS.

El cercador d'aquesta eina, l'Emir, permet la recuperació d'anotacions d'objectes concrets i la consulta per imatge partint d'aquestes anotacions, les quals incorporen els descriptors visuals.

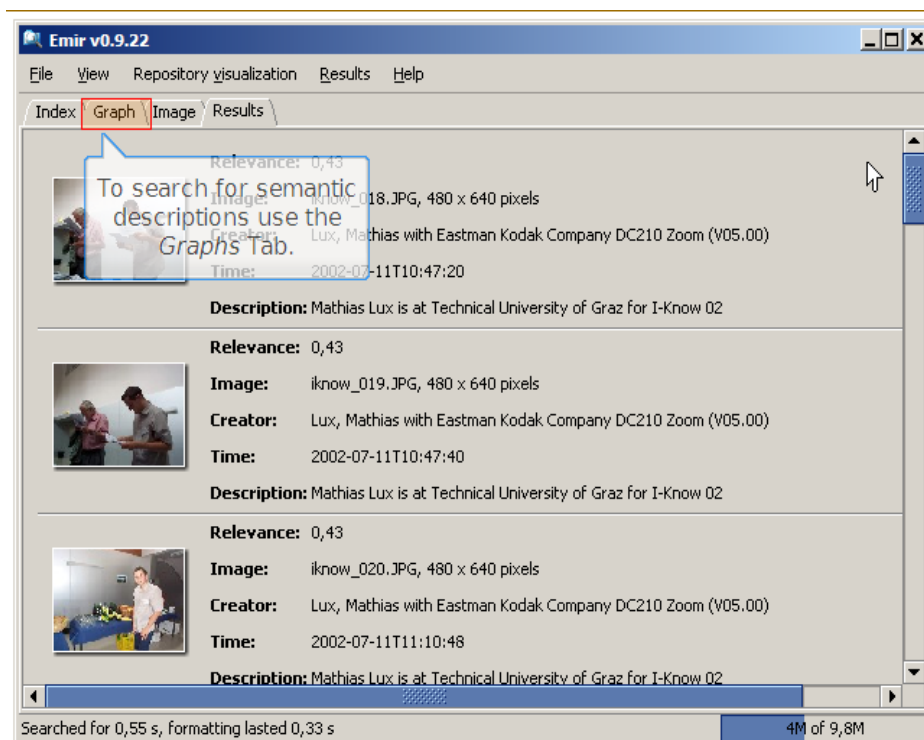


Figura 3: Resultats d'una consulta per imatge a l'Emir

<sup>5</sup> <http://www.acmmm09.org/default.aspx>



A més a més, l'Emir es basa en una indexador eficient de codi obert anomenada Lucene per a la recuperació intel·ligent del text. D'aquesta manera, recupera etiquetes amb diferents significats semàntics i les ordena basant-se en relacions entre elles mitjançant un sistema jeràrquic basat en nodes.

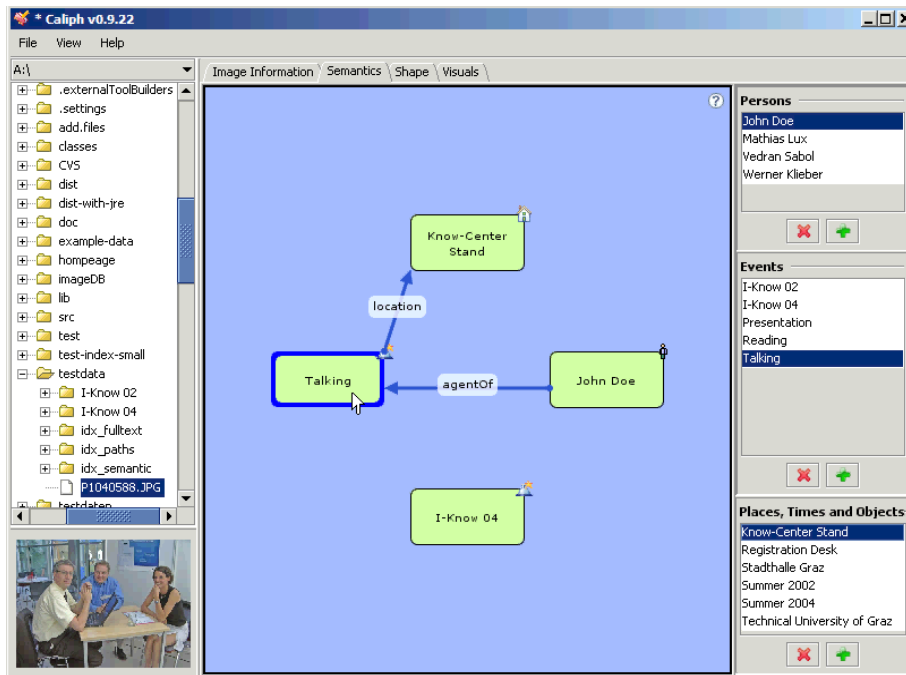


Figura 4: Anotació mitjançant relacions semàntiques al Caliph

## 2.3 Segmentació interactiva

La selecció de regions en una imatge avarca molts àmbits diferents, com podria ser l'edició de fotografies, aplicacions mèdiques o, simplement, l'etiquetat d'una persona o un objecte sense buscar una regió definida. Existeixen diverses tècniques en la literatura per assistir a l'usuari en la selecció de l'objecte de cerca. Per això cal comentar que aquesta avaluació de l'estat de l'art se centra en l'extracció d'objectes d'una imatge. Aquesta es coneix com a **segmentació interactiva**, ja que combina la segmentació de la imatge en regions amb la interacció entre l'usuari i la màquina.

### 2.3.1 Marcadors

Per dur a terme la segmentació interactiva, l'usuari treballa amb la interfície mitjançant marcadors dibuixats sobre la imatge. Per tant, el primer pas és trobar uns marcadors que permetin la selecció de la regió d'interès dins de la imatge per acabar definint l'àrea de suport exacte associada a l'objecte. Aquests poden ser, per exemple: rectes, rectangles, el·lipses, punts, traces o, fins i tot, esbossos. Segons el tipus d'aplicació podem distingir dos tractaments dels marcadors [4] diferents:

1. Seleccions **basades en regions** a on l'usuari fa la selecció de segments a partir d'una partició de la imatge generada automàticament.
2. Seleccions **basades en contorns** a on l'usuari genera una corba que després s'ajusta automàticament als píxels del contorn de l'objecte.

Les dues opcions poden donar resultats similars però la interacció és més simple en el primer cas, ja que en el segon es força a l'usuari a dibuixar tot el contorn de l'objecte i la precisió dels resultats dependrà de la precisió de l'usuari. De l'altre manera, l'usuari només ha de dibuixar uns marcadors que, juntament amb la partició prèvia, donaran lloc a una segmentació de l'objecte.

A més a més, podem distingir entre **marcadors d'objecte i marcadors de fons**. Els marcadors d'objecte s'utilitzen per la selecció de l'objecte que es vol extreure mentre que els marcadors de fons serveixen per descartar les regions que no interessin i, per tant, pertanyen al fons. D'aquesta manera l'algoritme compta amb més informació i pot apuntar de forma més precisa a l'objecte d'interès.

### 2.3.2 Propagació dels marcadors

Un cop arribats a aquest punt, és necessari adoptar un criteri de propagació dels marcadors a la resta de la imatge ja que s'ha de decidir a quina regió pertany cadascun dels píxels que no han sigut marcats ni com a objecte ni com a fons. Específicament, en el cas d'una selecció basada en regions, caldrà decidir si les regions pertanyen a l'objecte

o al fons. En canvi, en el cas d'una selecció per contorns, s'haurà de buscar un altre mètode, com l'estudi de cada píxel. En el present estat de l'art s'han considerat les següents propostes:

Nom de la tècnica	Grau d'interacció	Descriptor
SRG	Alt	Color
SIOX	Mitjà	Color
BPT	Baix	Color, textura, dispersió, ...

1. **Creixement de regions a partir de llavors - *Seeded region growing (SRG)*** [5] [6]: En aquest cas partim únicament dels píxels marcats, els quals s'agruparan en subconjunts de píxels formant regions. El marcador es comporta com una petita llavor que s'expandeix fusionant-se amb els píxels adjacents per formar una regió més gran. La tècnica es basa en el color dels píxels del voltant en comparació amb el propi color del píxel marcat. D'aquesta manera, es marca un subconjunt de píxels per a cada regió diferenciable de la imatge, incloent el fons, i s'afegeixen o es descarten els píxels adjacents als seleccionats de forma iterativa segons la similitud del color respecte al píxel seleccionat. Aquesta tècnica no té cap fundació matemàtica ni probabilística i això fa que sigui limitada però, tot i així, la seva fàcil implementació i baix cost computacional l'han fet popular. Va ser proposada per Adams i Bischof.

A la Figura 5 (a) podem veure una senzilla implementació ideal a on tots els objectes tenen colors homogenis. Per tant, quan es marquen els dos píxels (b), comença l'estudi dels píxels del voltant seguint un procés iteratiu. En aquest cas, només hi hauran dues iteracions (c i d).

A la Figura 6 (a) observem un dibuix homogeni que representa un presentador amb un fons color pastís amb dibuixos marrons. Suposant que es vulgui seleccionar el presentador sencer descartant el fons, Figura 6 (b), amb aquest mètode s'hauran de crear fins a 6 subconjunts de píxels independents que s'expandiran fins a trobar

les regions corresponents a la mà esquerra, la mà dreta, la cara, el cos, el braç esquerre i el braç dret. D'aquesta manera, cada cop que se selecciona un píxel del les diferents regions, tota la regió queda seleccionada, com es veu en la Figura 7 (a).

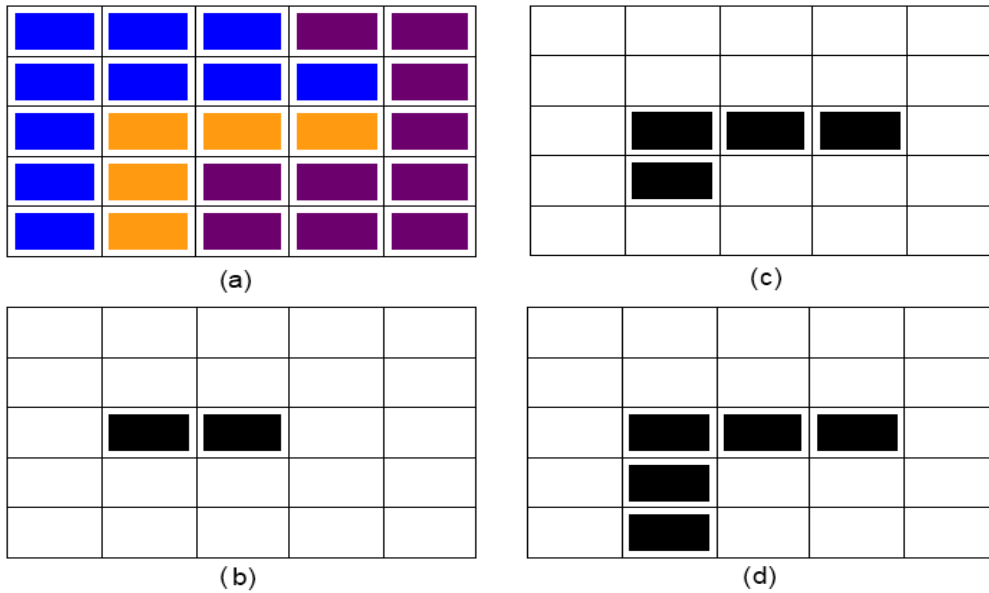


Figura 5: Interacions a partir de dos píxels marcats

Cal apuntar que aquesta eina incorpora un nivell de tolerància que permet crear un valor mig del color de l'objecte a extreure i un rang, de manera que tots els píxels del voltant del píxel seleccionat que estiguin a dins d'aquest rang siguin afegits a la regió. Aquesta tolerància permet aplicar el mètode a imatges no homogènies.

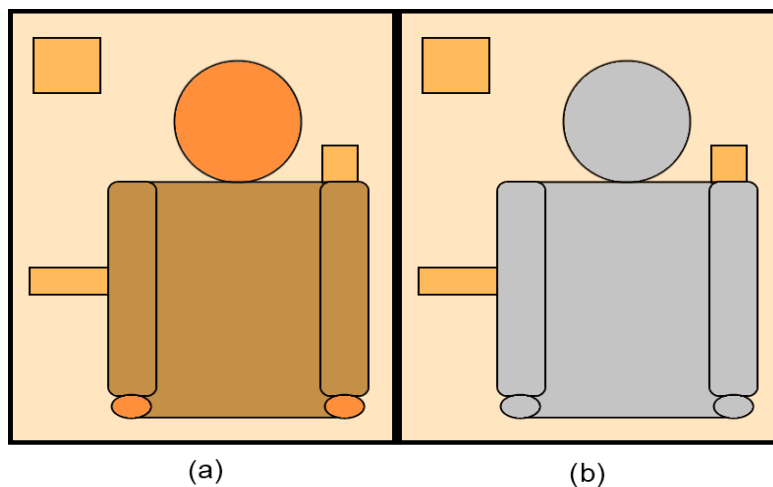


Figura 6: Exemple simple d'extracció d'una regió

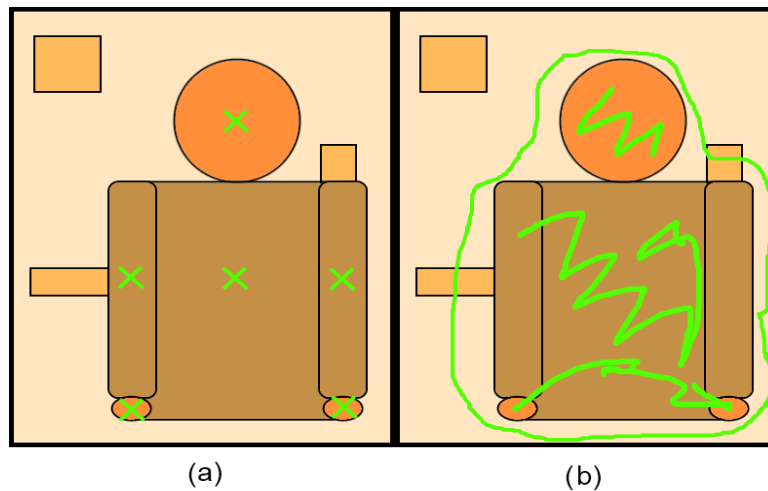


Figura 7: Exemple homogeni amb SRG i SIOX

2. **Extracció interactiva d'objectes mitjançant color - Simple Interactive Object Extraction (SIOX)** [5]:

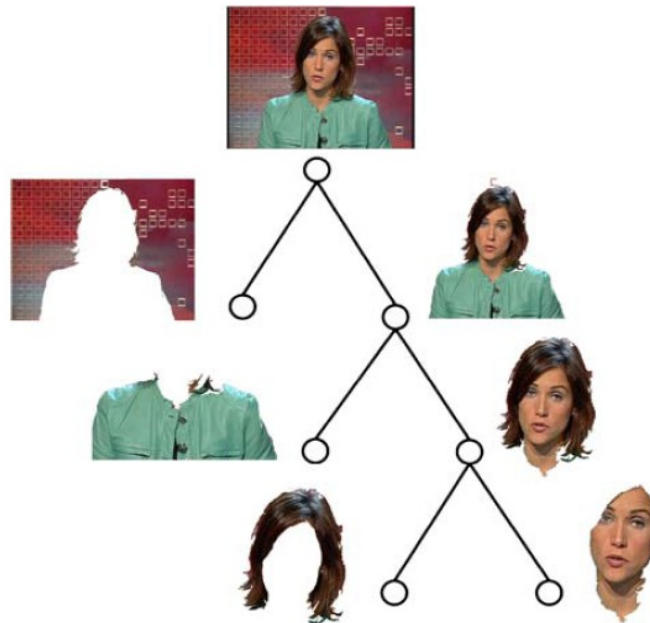
En aquest cas també partim de subconjunts de píxels marcats com a regió. L'eina crea un model de color per cadascuna d'aquestes regions a partir del color de cada píxel pertanyent al mateix subconjunt i assigna a cadascun dels píxels de la imatge una regió determinada. Per assignar la regió, compara el color del píxel amb els diferents models i li atribueix el model amb menor distància al seu color.

Si apliquem el mateix exemple homogeni que en la tècnica de SRG, en aquest cas els marcadors són traces però, com s'observa a la Figura 7 (b), és necessari marcar els mateixos punts: braços, mans, cos i cap. Els resultats també són molt similars ja que les dues eines es basen en el color dels píxels per fer la segmentació.

3. **Representacions jeràrquiques, els arbres de particions:** Aquest mètode parteix d'una partició prèvia de la imatge seguint criteris més complexos que les dues exposades a dalt, ja que la partició depèn de diferents característiques de la imatge com poden ser el color, la textura, la forma, la posició, etc. Els segments inicials s'anomenen fulles o nodes. A partir d'aquests, se segueix una algoritme iteratiu per a crear nodes a partir de la unió de diferents nodes inferiors, fills d'un mateix pare, seguint uns criteris de semblança preestablerts dins l'algoritme de partició

inicial. Aquest procés iteratiu se segueix fins a obtenir la imatge sencera.

Els arbres de particions binaris o *Binary Partition Trees* (BPT) [7] són un cas particular d'arbres de particions on cada pare compta amb dos únics fills. En cada pas de la interacció, la unió es realitza entre els dos nodes veïns més similars segons la distància definida. Aquesta distància es calcula a partir dels descriptors visuals de cadascuna de les regions.



*Figura 8: Descomposició d'una imatge en un arbre binari de particions*

S'han trobat tres propostes diferents que utilitzen arbres binaris de particions per l'extracció d'un objecte en una imatge. L'UPSeek [8] és la proposta de Giró, Camps i Marqués i utilitza punts com a marcadors. Les altres dues propostes, d'en Lluís Garrido [7] i de Tomasz Adamek [5] [9] utilitzen traces per marcar els objectes i el fons.

A les figures Figura 9 i Figura 11 trobem la partició de la imatge de la Figura 6 (a) en regions i l'arbre binari d'aquesta partició, respectivament. La tècnica de propagació de marcadors sobre l'arbre de particions es pot dividir en dos passos:

### Conversió de marcadors a fulles:

Un cop s'han dibuixat els marcadors sobre la imatge, les seleccions es tradueixen en segments de la partició inicial. En el cas d'utilitzar punts, un clic del ratolí es tradueix en una única fulla marcada de la partició. En canvi, si es parteix de traces, es pot marcar més d'una fulla alhora i, a més, es pot diferenciar entre marcadors d'objecte i marcadors de fons. En l'exemple de la Figura 10, les fulles 2, 3 i 4 pertanyen a la regió objecte i, pel contrari, les fulles 5, 6 i 8 pertanyen a la regió fons. Les fulles 1 i 7 queden indefinides.

### Propagació d'etiquetes:

Cal definir les regions que no han sigut assignades com a objecte ni com a fons amb els marcadors. Anomenarem etiqueta a cadascuna de les regions. En el cas de l'extracció d'un objecte dins d'una imatge, es vol cercar una única regió i, per tant, només distingim dues etiquetes: l'etiqueta positiva es refereix a la regió objecte i l'etiqueta negativa al fons. A continuació s'exposaran tres propostes diferents per a la propagació d'etiquetes:

- a) Proposta manual de Giró – Camps – Marqués
- b) Proposta automàtica de Lluís Garrido
- c) Proposta automàtica de Tomasz Adamek

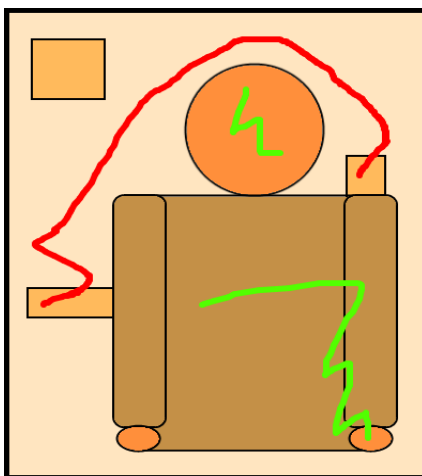


Figura 10: Etiquetes inicials

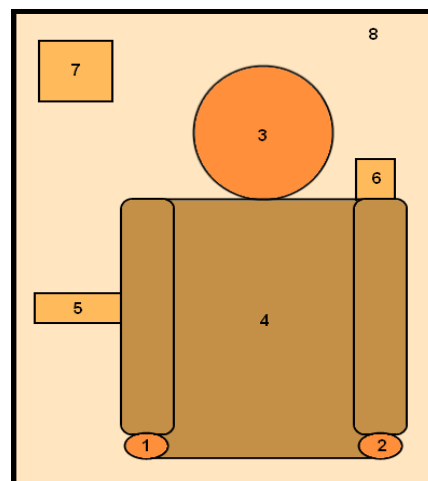
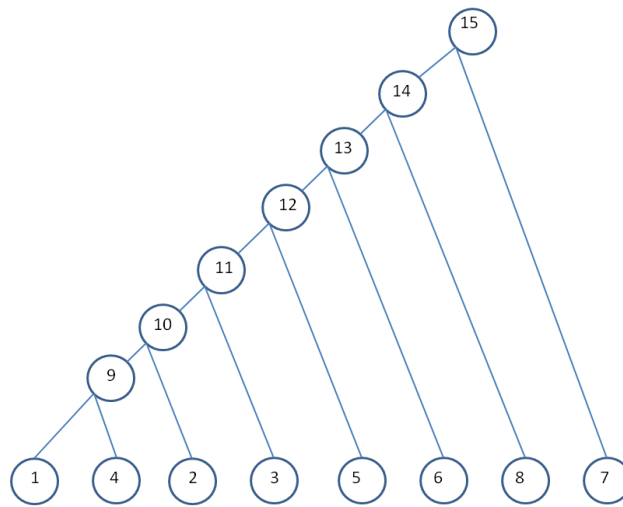


Figura 9: Exemple de segmentació



*Figura 11: Arbre binari de particions*

- a)** La proposta de Giró, Camps i Marqués, l'UPSeek, defineix una única fulla de partida com a objecte. L'etiqueta es propaga de forma manual mitjançant la rodeta del ratolí. Quan l'usuari fa girar la rodeta cap avall, l'etiqueta es propaga cap al node pare de la fulla seleccionada. A partir d'aquí se segueix un procés iteratiu en el qual, cada vegada que l'usuari gira la rodeta cap avall, l'etiqueta es propaga cap al node pare de l'últim node seleccionat. Si es vol tornar cap enrere, l'usuari haurà de girar la rodeta en sentit contrari.

Seguint l'exemple de les figures Figura 9 i Figura 11, si l'usuari fes un primer clic sobre la fulla 4, podria girar la rodeta seleccionant el seu pare, el node 9. En conseqüència quedaria seleccionada la fulla 1. A mesura que l'usuari anés girant la rodeta, aniria afegint les fulles 2 i 3. Així obtindria l'objecte.

- b)** La proposta de Garrido funciona de la següent manera: un node seleccionat pot expandir-se cap al seu pare sempre i quan el germà no estigui en conflicte amb el marcador, és a dir, que el germà no s'hagi assignat a una regió diferent. Sempre estem parlant de l'extracció d'un únic objecte a dins d'una imatge, per tant, la tècnica queda limitada a dues etiquetes. Les etiquetes positives s'aniran propagant fins a trobar una etiqueta negativa, que es convertirà en un node limitant.



A partir de l'arbre binari de la Figura 11 i els marcadors de la Figura 10, la Figura 12 mostra els resultats de la propagació a on s'observa l'expansió de l'etiqueta positiva fins a trobar un node negatiu que entra en conflicte i frena la propagació, Figura 12 (c). La fulla 7 queda sense assignar.

**Nodes sense etiquetar:**

Quan tenim un node sense assignar i aquest té més d'una regió veïna, una primera solució inicial és atribuir-li arbitràriament una d'aquestes regions o decantar-se per la regió més propera dins de l'arbre de particions que sigui veïna del node en quant a posicionament. Tot i així, hi ha diferents paràmetres que es poden tenir en compte a l'hora de fer aquesta assignació com la similitud o alguns criteris geomètrics. Garrido proposa assignar la regió que comparteix el major perímetre amb el node sense assignar. En aquest cas, la fulla 7 comparteix tot el seu perímetre amb la fulla 8, que ha sigut assignada al fons. Per tant, afegim aquesta fulla sense assignar a la regió de fons, com s'observa a la Figura 12 (d).

- c) La proposta d'Adamek, Figura 13, segueix els següents passos: per començar, es propaga cap a dalt l'etiqueta positiva o negativa des de les fulles inferiors (b) per tota la ruta de l'arbre (*bottom-up propagation*).

Seguidament, com es veu a la Figura 13 (c) en blau, es marquen els nodes conflictius. Els nodes conflictius són aquells que tenen fills amb diferents etiquetes. Per últim, els nodes no conflictius es propaguen a tots els nodes de la sub-branca dels seus fills (*top-down propagation*) (d).

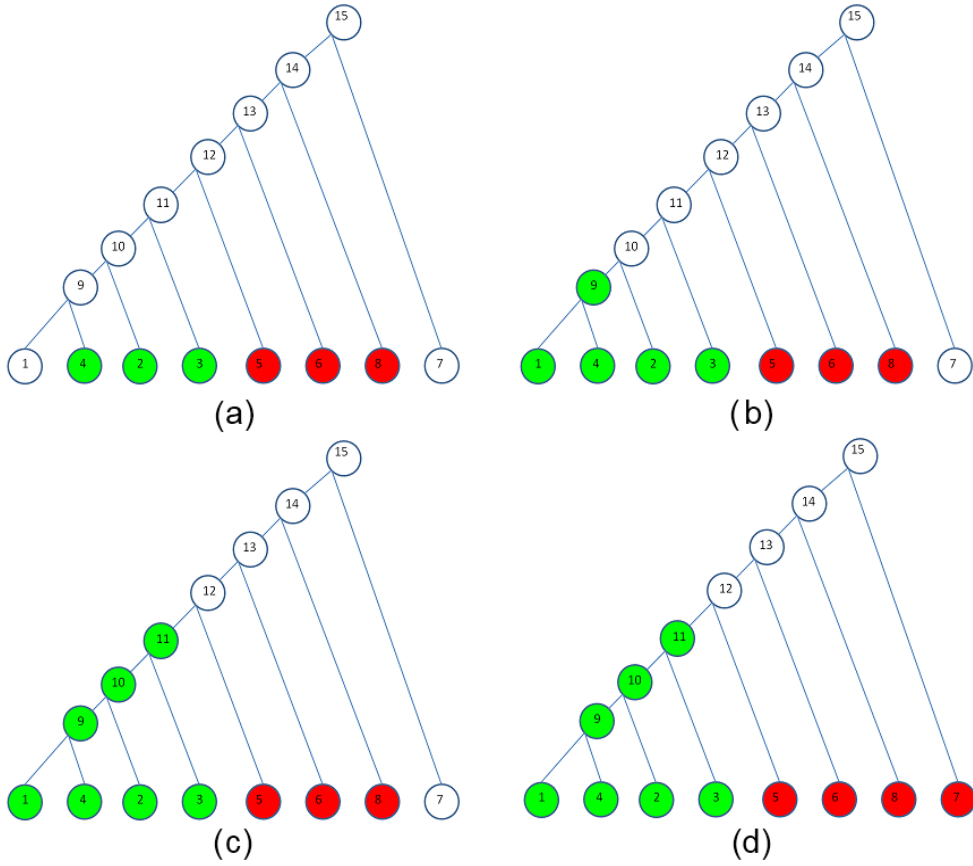


Figura 12: Propagació dels marcadors d'un BPT proposada per Garrido

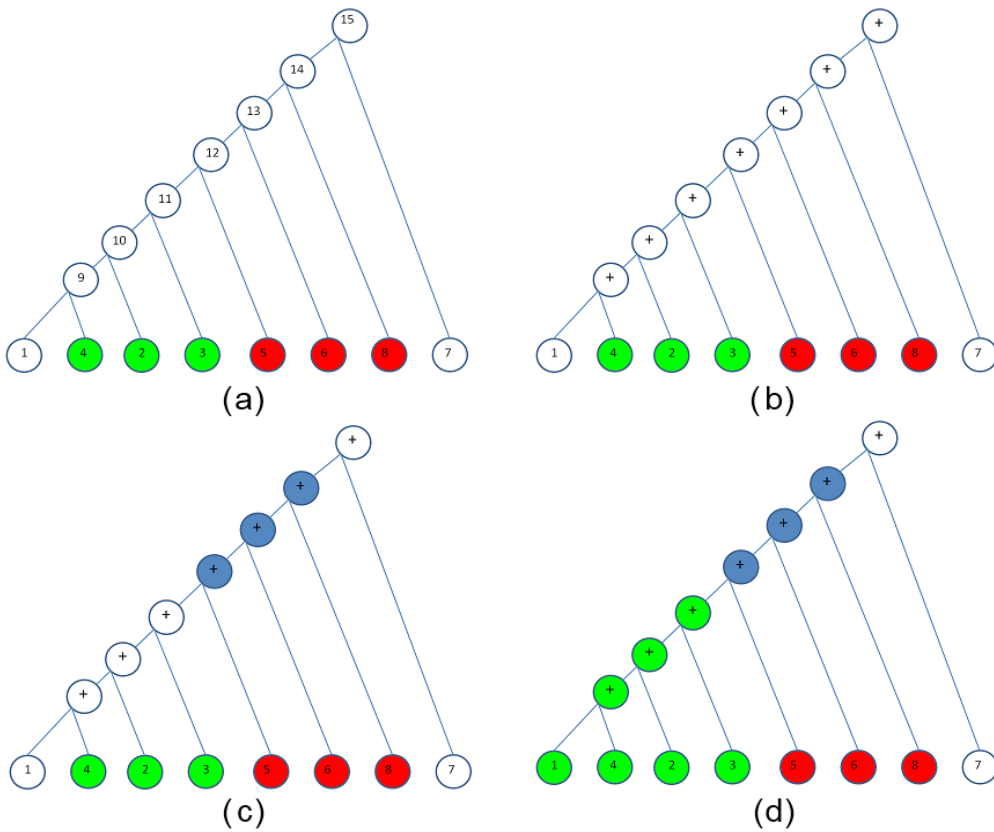


Figura 13: Propagació dels marcadors d'un BPT proposada per Adamek

**Nodes sense etiquetar:**

Adamek considera que és un error utilitzar els arbres binaris per solucionar els problemes causats per les limitacions d'aquests mateixos. Per això, proposa una solució que considera més encertada que la proposada per Garrido per a assignar una regió a les fulles sense etiqueta.

El procediment és semblant al SRG però treballant a nivell de node en comptes de treballar a nivell de píxels, la qual cosa disminueix el cost computacional.

El mètode descrit per l'autor consisteix en estudiar la distància de cada node sense assignar respecte al node adjacent més proper de cadascuna de les regions. En l'extracció d'un objecte distingim dues úniques regions: objecte i fons. Per tant, en el cas que el node sigui adjacent a una única regió, serà assignat a aquesta regió. Pel contrari, si el node comparteix perímetre amb totes dues regions, s'estudiarà la distància entre aquest node i el node més proper i adjacent pertanyent a cadascuna de les regions.

Per tant, la distància entre el node sense assignar  $R_i$  i un objecte  $O$ , serà la distància més curta entre  $R_i$  i una de les regions ja assignades a l'objecte  $O$  i adjacent a  $R_i$ ,  $R_j$ :

$$D(R_i, O) = \min_{R_j \in O} d_r(R_i, R_j)$$

A on  $d_r(R_i; R_j)$  es calcula com a una distància euclidiana entre el color mig representat en LUV entre les dues regions adjacents  $R_i$  i  $R_j$ .

L'autor planteja un estudi de l'esperança de que una fulla sense etiquetar s'assigni a un objecte:

$$C^F(R_i) = \begin{cases} \frac{D(R_i, O^B)}{D(R_i, O^F) + D(R_i, O^B)} & \text{if } \left( (D(R_i, O^F) \neq 0 \vee D(R_i, O^B) \neq 0) \wedge \right. \\ 0.5 & \left. (D(R_i, O^F) \neq \infty \vee D(R_i, O^B) \neq \infty) \right) \\ & \text{otherwise} \end{cases}$$

Aquesta esperança  $C^F$  tindrà valors entre [0,1]. Si el valor es troba per sobre del 0.5, indica que el node hauria de ser assignat a l'objecte i si es troba per sota, al fons. Si el valor és 0.5, s'interpreta que les distàncies són 0 o bé infinites. Una distància infinita indica que les dues regions no es toquen en cap moment. En conseqüència, la esperança de que una fulla sense etiquetar s'assigni al fons pot ser definida com:

$$C^B(R_i) = 1 - C^F(R_i).$$

Es pot definir una tercera variable C:

$$C(R_i) = \max \{C^F(R_i), C^B(R_i)\}$$

Aquesta compara ambdues esperances i decideix quedar-se amb el valor màxim. D'aquesta manera, comparant els valors de la variable C de cadascuna de les regions sense etiquetar (pertanyents al objecte  $O^U$  – Unlabeled), s'escull el node que retorni una esperança superior,  $R_{max}$ :

$$R_{max} = \arg \max_{R_i \in O^U} \{C(R_i)\}$$

Definitivament, es decideix si el node pertany a l'objecte o al fons recalculant l'esperança del node més idoni a dins de l'objecte,  $R_{max}$ :

$$C^F(R_{max}) > 0.5$$

En l'exemple, el node 7 comparteix perímetre únicament amb el node 8 i, per tant, se li assignarà la seva etiqueta sense necessitat de calcular distàncies.

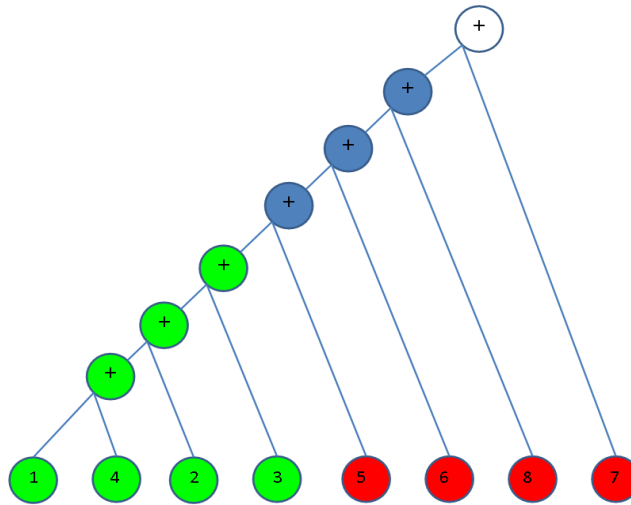


Figura 14: Solució d'Adamek per nodes sense assignar

### 2.3.3 Comparativa de interfícies gràfiques

La interfície gràfica que es presenta a l'usuari és tan important com l'eficiència i precisió de l'algoritme implementat ja que ha de ser suficientment atractiva i intuïtiva com perquè l'usuari vulgui fer-ne ús. A més a més, l'usuari ha de ser capaç d'interactuar amb l'ordinador sense que això suposi un gran esforç i arribar a un resultat acceptable en un màxim de temps raonable. En aquest apartat es farà un petit estudi de l'estètica i el procediment de diferents eines que permeten la selecció de regions actualment. Distingim entre interfícies web i interfícies d'escriptori:

#### 2.3.3.1 Entorn d'escriptori

Les eines que s'han trobat que utilitzen les tècniques explicades a dalt són interfícies d'escriptori, és a dir, no són accessibles via web. Això suposa un impediment ja que, en no disposar d'una interfície web, es requereix d'una instal·lació prèvia i, per tant, són **menys accessibles**. Aquestes són:

- **Adobe Photoshop<sup>6</sup>**

La tècnica de creixement de regions a partir de llavors (SRG) ha sigut implementada en la Vareta màgica de l'editor d'imatges comercial Adobe Photoshop.

1. Presentació a l'usuari: s'escull a través d'una icona dins la barra d'eines de dibuix.
2. Marcadors: punts.
3. Interacció de l'usuari per arribar al resultat: cada regió es marcar amb un únic clic sobre un píxel de la imatge. Per tant, si es vol seleccionar una regió amb colors diferenciats, el més probable és que s'hagi de repetir el procés més d'un cop.
4. Aspecte de la interfície: Figura 15.

- **Observació:** es poden sumar diferents regions fent ús de la tecla Ctrl.

- **GIMP<sup>7</sup>**

La tècnica d'extracció interactiva d'objectes mitjançant color és adoptada a l'editor d'imatges comercial GIMP, a l'eina de selecció de primer pla.

1. Presentació a l'usuari: s'escull a través d'una icona dins la caixa d'eines.
2. Marcadors: traces.
3. Interacció de l'usuari per arribar al resultat: per començar, es dibuixa un cercle aproximat al voltant de l'objecte que es vulgui extreure. Seguidament es marca l'objecte que es vol extreure dibuixant a sobre traces molt gruixudes de forma que la regió quedi pintada. Si es volgués retocar el resultat, es pot recórrer a una eina igual a l'anterior que permet pintar amb traces gruixudes les parts que pertanyen al fons per eliminar alguns trossos que no formin part de l'àrea d'interès
4. Aspecte de la interfície: Figura 16.

---

<sup>6</sup> <http://www.adobe.com/es/products/photoshop/photoshop/>

<sup>7</sup> <http://www.gimp.org/>

- **Observació:** Resulten de gran ajuda les instruccions que apareixen abaix per anar seguint les diferents passes de l'eina.

- **Clients UPSeek (GAT i GOS<sup>8</sup>)**

La tècnica dels arbres de particions binaris s'aplica en l'eina de cerca visual desenvolupada pel departament de processat d'imatge de la UPC, el GOS [7] (*Graphical Object Searcher*). En aquest apartat analitzarem el GOS només com a interfície de selecció de regions.

1. Presentació a l'usuari: s'escull a través d'una icona a dalt de la imatge.
2. Marcadors: punts.
3. Interacció de l'usuari per arribar al resultat: Després de fer un primer clic a sobre de la regió d'interès, l'usuari pot fer girar la rodeta del ratolí aconseguint ampliar la regió quan gira la rodeta cap abaix i estrènyer la regió quan gira la rodeta cap amunt. En el cas que es vulguin seleccionar diferents parts que l'algoritme no entén com a una mateixa regió, només s'ha de repetir el mateix procés tants cops com sigui necessari.
4. Aspecte de la interfície: Figura 17.

- **Observació:** la principal limitació d'aquesta tècnica és la necessitat d'interactuar amb un ratolí.

- **IST**

L'eina que proposen Kevin McGuinness i Noel E. O'Connor [5], anomenada *Interactive Segmentation Tool* (IST) pot utilitzar qualsevol de les tres tècniques anteriors, que serà triada per l'usuari.

1. Presentació a l'usuari: l'algoritme s'escull a través del menú *Tool*.
2. Marcadors: traces.

---

<sup>8</sup> <http://gps-tsc.upc.es/imatge/i3media/gos/>

3. Tècnica de propagació: BPT, SIOX, SRG i *Interactive Graph Cuts*.
4. Interacció de l'usuari per arribar al resultat: S'han de dibuixar traces positives i negatives. L'eina obliga a l'usuari a marcar ambdues abans de mostrar els primers resultats. Un cop s'han dibuixat, la transparència s'actualitza cada cop que es fa una nova traça.
5. Aspecte de la interfície: Figura 18.

### ***Punts forts del IST:***

- Compta amb diferents modes de pantalla: es pot mostrar la imatge amb una transparència superposada mostrant la regió inclosa pels marcadors, també pot mostrar la imatge eliminant els objectes marcats com a fons o bé la imatge amb la vora de l'objecte marcada. També pot mostrar la mascara en blanc i negre.
- Utilitza el botó esquerre del ratolí per a dibuixar els marcadors positius i el botó dret del ratolí per a dibuixar els marcadors negatius.



*Figura 15: Interfície de Adobe Photoshop*





Figura 16: Interficie del GIMP

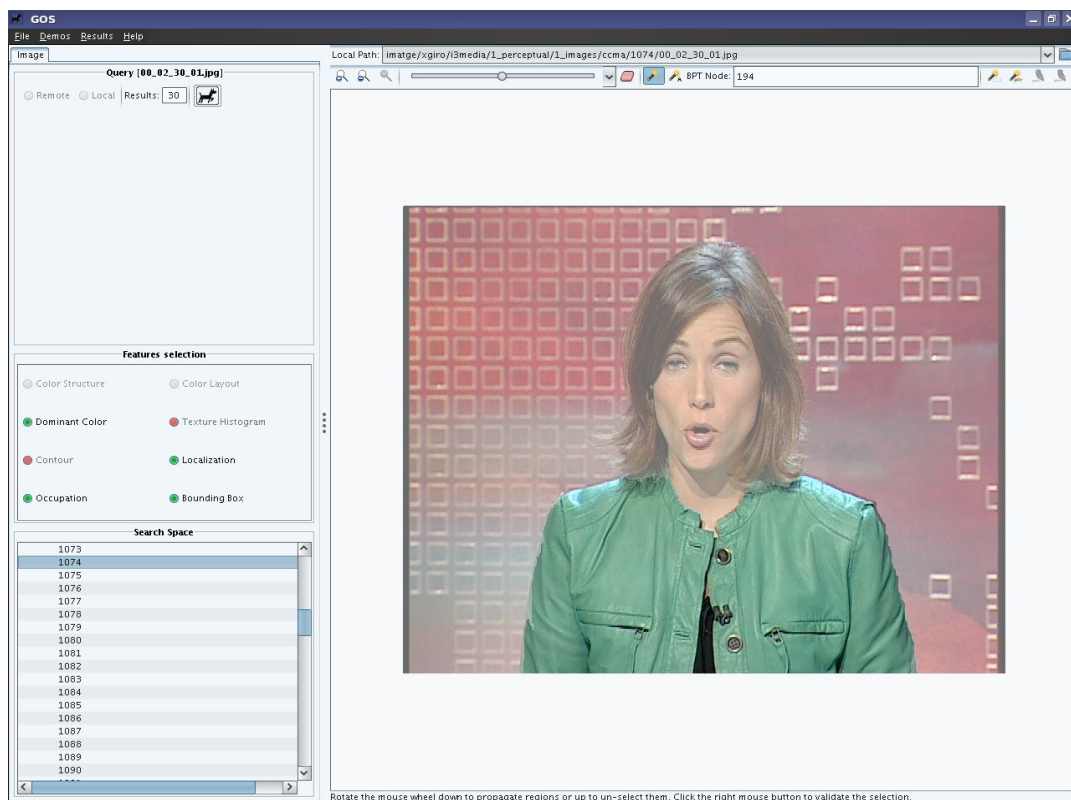


Figura 17: Interficie del GOS



*Figura 18: Interfície del IST*

### 2.3.3.2 Entorn web

Trobem també alguns precedents de segmentació interactiva accessibles via web. La manca d'aquestes eines resideix en el mètode per extreure l'objecte de la imatge ja que no utilitzen cap tècnica per a la propagació d'etiquetes sinó que es tracta d'una segmentació completament **manual**. Aquestes interfícies demanen un esforç gran a l'usuari, ja que aquest haurà de dibuixar tot el **polígon** que contingui la regió que es vol seleccionar.

En concret, el LabelMe<sup>9</sup> és una interfície de segmentació manual que permet dibuixar un polígon a partir de línies rectes enllaçades. Una altra eina, Squigl<sup>10</sup>, permet dibuixar de forma lliure a sobre de la imatge. Aquestes interfícies serveixen per anotar imatges.

<sup>9</sup> <http://labelme.csail.mit.edu/>

<sup>10</sup> <http://www.gwap.com/gwap/gamesPreview/squigl/>

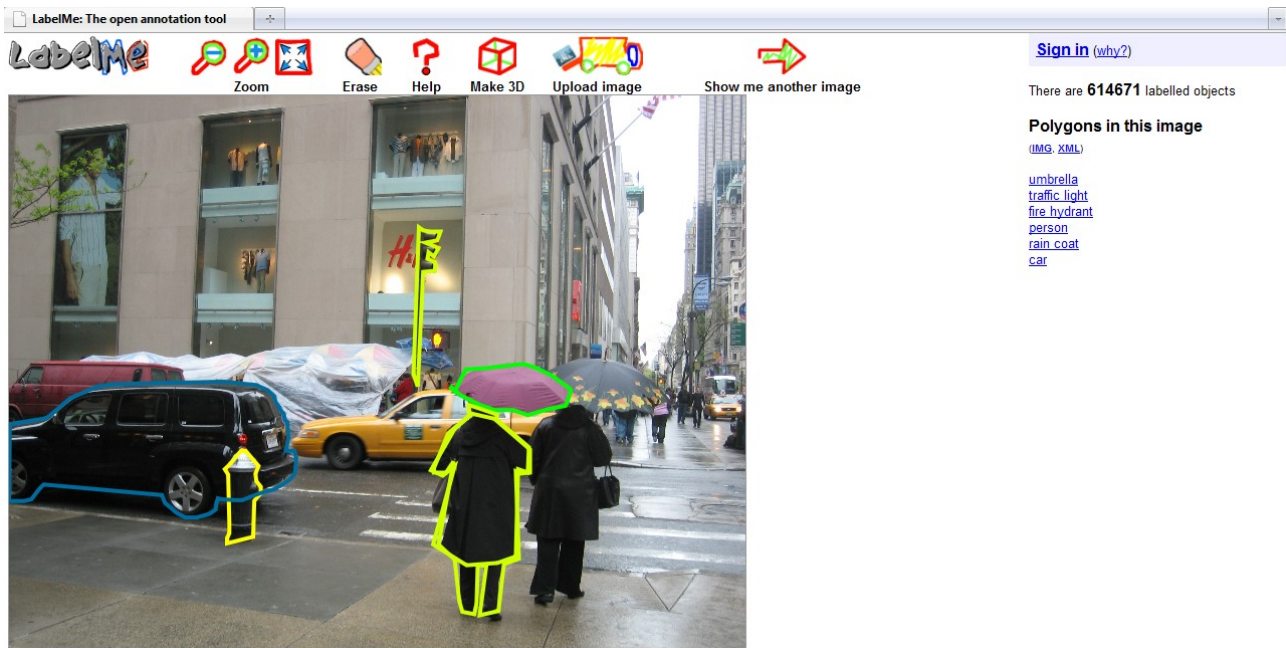


Figura 19: Interficie del LabelMe

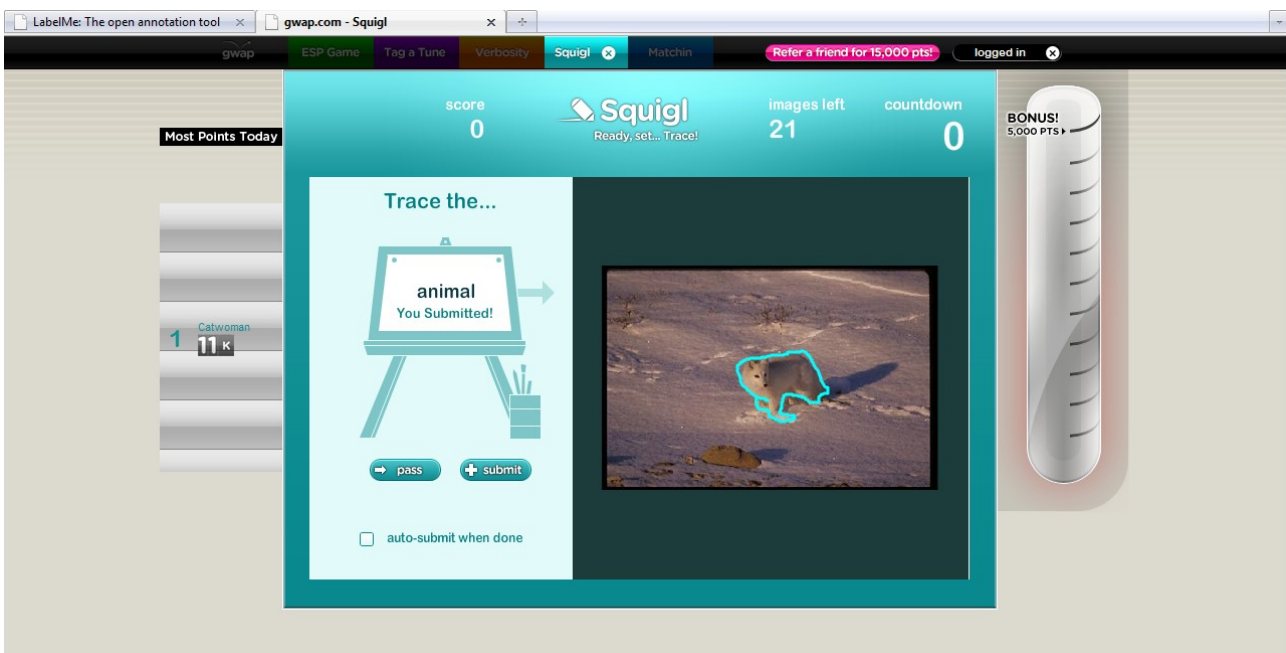


Figura 20: Interficie del Squigl

## 3 Requeriments

### 3.1 Requeriments tècnics

La CCMA compta amb una sèrie d'eines per a l'emmagatzematge, anàlisi i cerca de vídeos i imatges. L'objectiu d'aquest projecte és la integració de totes aquestes eines en una nova interfície global capaç d'oferir les funcionalitats de cadascuna de les seves precedents. Aquestes eines són el Digion, el Digimatge i el Detector de text.

#### 3.1.1 Descripció dels elements existents

##### 1. Digion:

El Digion és el repositori de la CCMA on es fa la ingesta de tot el material audiovisual per la consegüent cerca digitalitzada d'aquest.

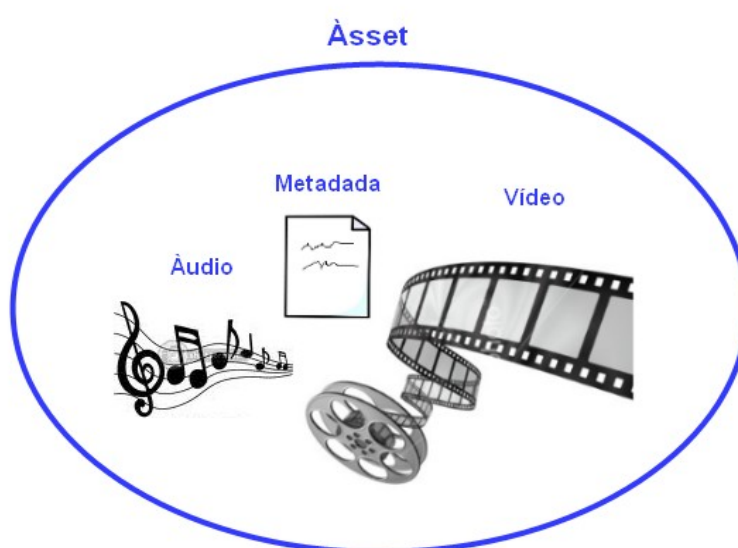


Figura 21: Composició d'un àsset

Una ingesta consisteix en la incorporació del material audiovisual des de la càmera de vídeo al repositori així com la conversió d'aquest al format de treball. La Corporació treballa amb un conjunt de fitxer de vídeo, àudio i metadades que anomena *assets*. Les metadades dels assets contenen les descripcions del contingut audiovisual que permetran recuperar l'asset mitjançant cerques textuais.

El Digition és, també, l'eina que la Corporació ofereix als documentalistes i periodistes de TVC (Televisió de Catalunya) per a la cerca digitalitzada d'assets. Cada asset està format per un fitxer de vídeo, un d'àudio i un tercer amb la metadada relacionada amb el vídeo i amb cadascun dels *keyframes* que el conformen. Els *keyframes* són les imatges clau que s'extreuen del vídeo per agilitzar la cerca i poder fer un primer cop d'ull més ràpid al vídeo. Els *keyframes* són les imatges base sobre les quals fem les cerques.



Figura 22: Mòduls del Digition

El Digition consta de **quatre mòduls**:

- El mòdul de **resultats** ofereix un llistat de resultats ordenats per *ranking*, 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. Aquest mòdul és la base del nostre treball final.
- El mòdul de **metadada** mostra tota la fitxa de l'àsset. La metadada està formada per 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. Digimatge

El Digimatge és la proposta d'en Ramon Salla [9] com a Projecte de Final de Carrera. Aquesta eina pretén ampliar les possibilitats de cerca que permet el Digiton afegint una cerca per imatge basada en els *keyframes* que conformen els àssets. Aquesta cerca retorna nous *keyframes* procedents de diferents àssets que han sigut processats i comparats amb el *keyframe* que ha iniciat la cerca. Òbviament, els *keyframes* resultants són aquells que han donat un nivell de semblança superior a l'original seguint uns criteris basats en descriptors visuals.

El Digimatge comparteix una estètica i estructuració comuna amb només dos mòduls per mostrar el resultat de la cerca textual o de la cerca basada en imatge. Amb això es pretén simplificar l'eina donant en cada moment únicament la informació necessària.

A la Figura 23 es mostra un exemple de cerca textual a on trobem els *keyframes* de l'àsset en el mòdul inferior i les metadades d'aquest àsset en el mòdul superior.

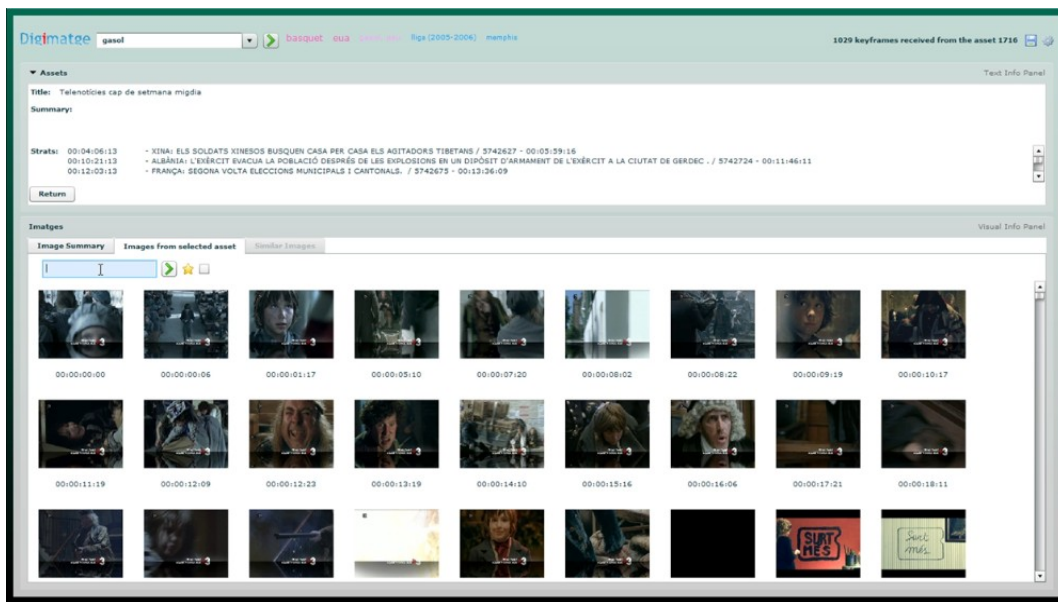


Figura 23: Mostra d'un únic àssset

A la Figura 24 es mostra un exemple de cerca per imatge a on trobem els *keyframes* resultat provinents de diferents àsssets en el mòdul inferior i un llistat amb informació de cada àssset resultant al mòdul superior.

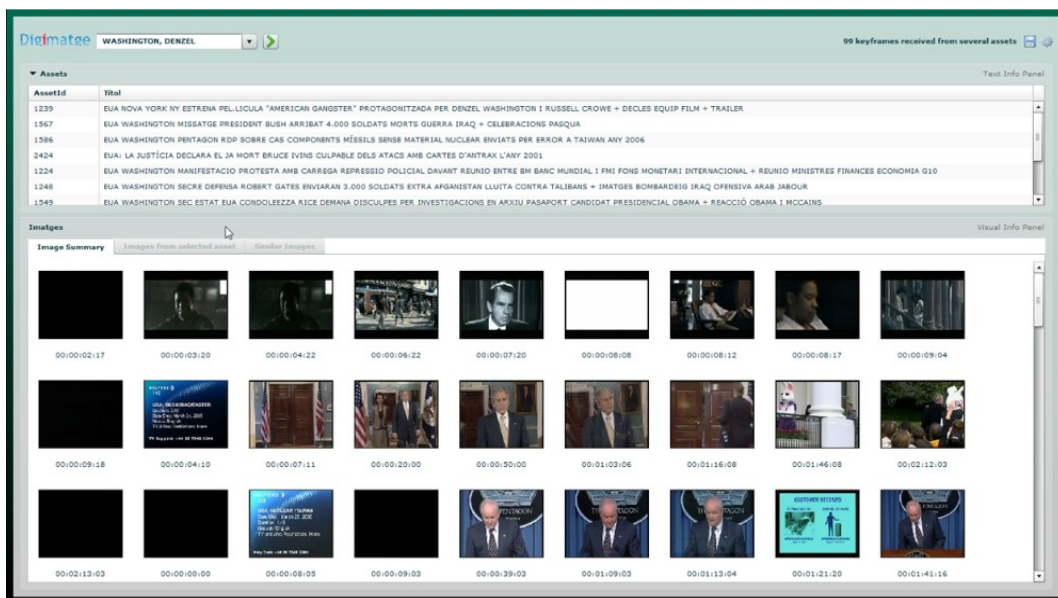
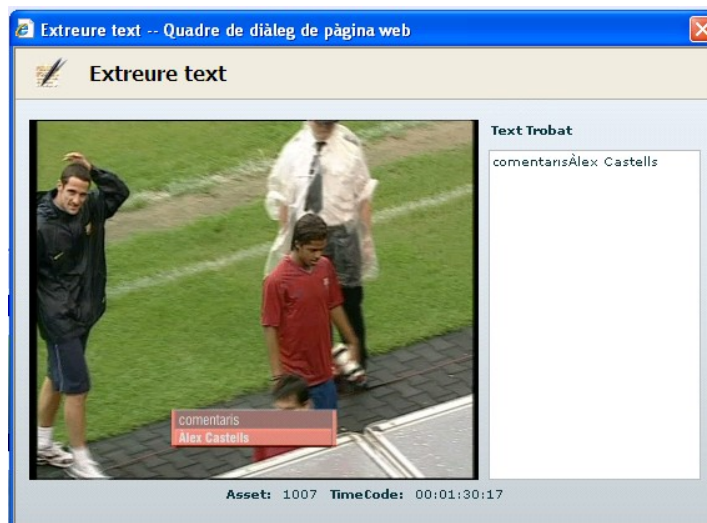


Figura 24: Mostra dels keyframes resultants d'una cerca

### 3. Detector de text

El detector de Text és un petit mòdul implementat en el Digition per la cerca de text a dins d'una imatge. L'usuari escull un *keyframe* i demana el text que es llegeix d'aquest. Normalment, el text es troba als telenotícies i vídeos d'esports, com s'observa en l'exemple de la Figura 25. Els resultats es mostren visualitzant la imatge amb un requadre a sobre del text trobat i, a la seva dreta, el text escrit en un editor de text.



*Figura 25: Detector de text*

#### 3.1.2 Integració i millores

La interfície final ha de permetre una cerca multimodal combinant les funcionalitats del Digition, Digimatge i Detector de text permetent tant la cerca textual com la consulta per imatge i la detecció de text.

Una de les millores que es vol incorporar ve motivada per les necessitats expressades pels usuaris de TVC. El Digimatge va ser sotmès a unes proves d'usuari a on bona part d'aquests usuaris van coincidir en un punt negatiu de l'eina: demanaven uns resultats més clars a on poguessin obtenir més informació visual sobre la procedència de cada imatge resultant, ja que, en tractar-se d'una cerca basada en el processat de la



imatge, cada imatge podia provenir de qualsevol vídeo. Per això, en aquest projecte s'ha buscat la manera de cobrir aquesta mancança.

A més, es demana la seva millora mitjançant la incorporació d'una nova eina per a la cerca d'imatge mitjançant regions. Per tant, es requereix una eina capaç de fer una segmentació interactiva per seleccionar una regió dins d'una imatge amb diferents marcadors. Aquesta eina ha de permetre, a més, fer cerques basades en el contingut visual de la regió seleccionada i proposar els resultats a l'usuari de forma ordenada.

Amb aquesta finalitat, l'eina es comunicarà amb l'UPSeek, el mòdul de la UPC que s'encarrega del processat de la imatge i d'oferir els resultats de la cerca, tant a nivell d'imatge global com de regió. A més a més, a nivell de regió, la UPC oferirà una màscara amb els resultats de cada marcador dibuixat per l'usuari en temps real fins que aquest obtingui la regió desitjada i decideixi fer la consulta per regions; tot seguint una arquitectura distribuïda.

Dins de la pròpia interfície s'han d'integrar els descriptors globals i els descriptors basats en regions. Aquests capacitaran a l'usuari d'una major precisió de cerca permetent que decideixi la importància de diferents elements característics de la imatge representats per aquests descriptors visuals i uns pesos que configurarà l'usuari. Aquests descriptors s'enviaran, juntament amb la informació necessària per a localitzar la imatge que s'està manipulant cap al mòdul de la UPC. En el cas de fer una consulta per regió, també s'enviaran els marcadors per seleccionar la regió d'interès.

Per poder dur a terme l'extracció d'un objecte dins d'una imatge és necessari un mètode de segmentació de la imatge. Per això, un altre requeriment a l'hora de dur a terme el projecte és el disseny d'un algoritme de propagació d'etiquetes per arbres binaris de particions (BPT): l'algoritme que decidirà a quina regió (objecte o fons) pertany cada píxel de la imatge es basarà en una partició prèviament calculada de la imatge i, a partir d'aquesta, s'assignarà la regió a cada node passejant-se per la ruta i les branques de l'arbre seguint uns criteris establerts.

### **3.2 *Requeriments d'entorn***

El primer requeriment que fa l'empresa i que condicionarà tant l'aprenentatge com el desenvolupament, és l'entorn de programació amb que s'implementarà l'eina: Adobe Flex. El Flex és un conjunt d'APIs, connectors i llenguatges de programació desenvolupat per Adobe que dona suport al desenvolupament d'Aplicacions Riques d'Internet (RIA) basat en la seva plataforma Flash. Això fa possible la realització de canvis sobre la pàgina sense la necessitat de recarregar-la contínuament ja que aquesta classe d'aplicacions s'executa al client. Així, s'augmenta la velocitat i usabilitat de les aplicacions permetent una major interactivitat.

### **3.3 *Requeriments funcionals***

Una característica que no pot fallar en una RIA és la usabilitat. El Flex permet oferir a l'usuari un disseny elegant i clar que concedeix una interfície atractiva, de fàcil comprensió i ús. Per això, és necessari fer un esforç en el disseny de l'eina cuidant tots els detalls de presentació per aconseguir oferir una interfície amigable per l'usuari final.

Per últim, l'eina serà accessible via web, per tant, es tindrà accés des de qualsevol ordinador dins de la Corporació, indiferentment del sistema operatiu o del navegador que s'utilitzi.

## 4 Disseny

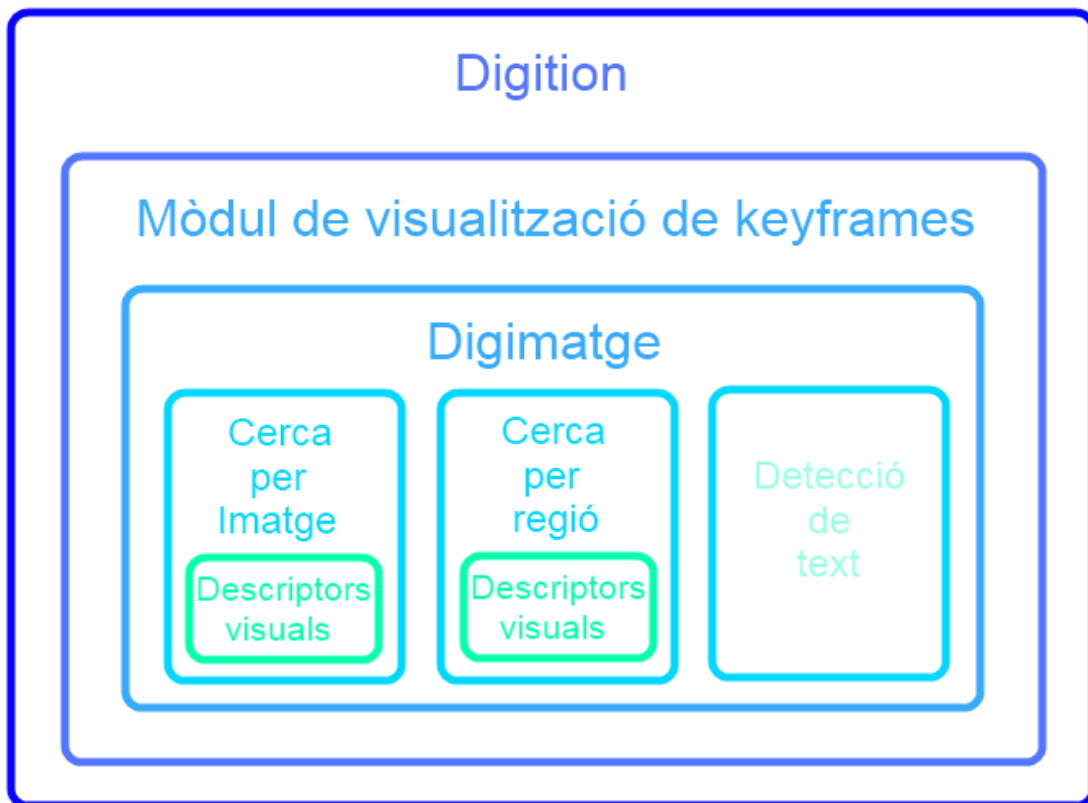
En aquest apartat es presenta l'eina global de cerca d'imatges i vídeos que permet una consulta multimodal dels assets que conformen el repositori de la Corporació agilitzant la cerca a l'usuari final. Aquesta eina s'ha desenvolupat a partir de la unió i adaptació del Digiton, el Digimatge i el Detector de text. A més, s'ha incorporat la nova interfície de cerca basada en regions, el **Digitrace**.

A continuació s'exposa la proposta de segmentació interactiva mitjançant arbres binaris de particions. Es proposen dos modes independents per marcar la regió: el mode rectangle i el mode traça. Seguidament, es presenta detalladament el Digitrace amb els seus mòduls, funcionalitats, configuracions i opcions. Per últim, es donarà una visió general del funcionament de la comunicació entre la CCMA i l'UPSeek, mòdul de la UPC, que serà especificada més endavant, en l'apartat de desenvolupament.

### 4.1 Eina global

El disseny de l'eina final consisteix en una suma jeràrquica del Digiton, Digimatge i Cercador de text així com de la inclusió del Digitrace i dels menús de pesos pels descriptors visuals globals i locals.

Com s'observa a l'esquema de la Figura 26, tota l'eina multimodal ha estat integrada a l'entorn del **Digiton**. En concret, el seu mòdul de visualització *de keyframes* consta d'una llista d'imatges en miniatura amb un estructura i unes funcionalitats definides, que a partir d'ara anomenarem **Digimatge**, ja que aquesta conté, també, part de les classes i codi del Digimatge original. El nou Digimatge ofereix les tres funcionalitats que s'han esmentat: consulta basada en imatges, consulta basada en regions i detecció de text.



*Figura 26: Eina global*

En el cas de fer una **consulta basada en imatge**, l'usuari tindrà llibertat per escollir els pesos dels descriptors visuals abans de fer la cerca. Si es vol fer una **consulta basada en regió**, l'eina mostrarà la interfície del Digitrace i l'usuari haurà d'interactuar amb aquesta per seleccionar la regió i els pesos dels descriptors visuals. Per últim, si es vol fer una **detecció de text** en una imatge, es farà visible la interfície del Detector de text mostrant la imatge de partida i el text detectat.

## 4.2 Digitrace

En aquest apartat s'introdueix la nostra proposta de segmentació interactiva, que s'ha dut a terme mitjançant arbres binaris de particions. A continuació es presenta el disseny de la interfície del Digitrace amb els seus elements i les funcions i relacions de cadascun d'ells.

### 4.2.1 Segmentació interactiva mitjançant BPT

El disseny dels algoritmes de segmentació interactiva s'ha fet a partir d'una imatge prèviament segmentada mitjançant un **arbre binari de particions** i sota la suposició de que es parteix d'un bon arbre de particions. Per tant, totes les decisions es basaran en aquesta partició prèvia a la interacció amb l'usuari i que no es modificarà.

A més a més, cal destacar que el punt fort de la proposta que s'exposarà a continuació consisteix en el desenvolupament d'un algoritme prou restrictiu i precís com per poder donar un primer resultat coherent amb **marcadors únicament d'objecte** sense necessitat d'utilitzar els marcadors de fons, la qual cosa suposa un avantatge des del punt de vista de l'usuari tant en quant a eficàcia i eficiència com en quant a usabilitat i intuitivitat.

Per últim, remarcuem que, com ja s'ha explicat en l'apartat de l'estat de l'art, l'objectiu de la interfície de selecció de regions és separar un únic objecte del fons. Per tant, en tot moment es treballa amb dues úniques etiquetes: etiqueta positiva per l'objecte i etiqueta negativa pel fons.

#### 4.2.1.1 Marcadors

En aquest projecte es presenten **dos modes** de segmentació interactiva segons els marcadors que s'utilitzen:

##### a) Mode rectangle

Aquest mètode parteix d'una selecció inicial a partir d'un rectangle. Un cop dibuixat aquest primer marcador i vista la regió resultant, la selecció es pot modificar amb punts que seleccionaran o deseleccionaran segments de la partició inicial de la imatge. Per utilitzar aquest mode, s'ha de seguir un procediment específic de forma ordenada:

## 1. Rectangle com a únic marcador d'objecte:

Tots els nodes continguts plenament a dins del rectangle són considerats nodes positius, és a dir, objecte. La resta de nodes es converteixen automàticament en fons, per tant, tots els nodes queden etiquetats.

## 2. Punts a dins del rectangle:

A continuació, l'usuari pot modificar la selecció automàtica mitjançant punts. Aquest pas consisteix en realitzar un nombre indefinit de clics a dins de l'àrea predeterminada pel rectangle que serviran per afinar el primer resultat. En definitiva, aquests clics defineixen un píxel a dins del rectangle i, a partir d'aquesta informació es troba la fulla sobre la que l'usuari vol interactuar.

Seguidament, coneixent l'etiqueta (objecte o fons) d'aquesta fulla, recorrem a l'invers del que trobem. És a dir, si la fulla on s'ha fet el clic havia estat etiquetada com a objecte, passarà a formar part del fons i viceversa.

### b) Mode traça

El mode traça es basa en dibuixar traces sobre la imatge. En aquest cas es pot diferenciar entre traces que marquen l'objecte i traces que marquen el fons. El primer pas en una segmentació basada en un arbre de particions és la **conversió dels marcadors en fulles**. Distingim 4 tipus de fulles:

- Fulles amb marcadors d'objecte
- Fulles amb marcadors de fons
- Fulles sense marcar
- Fulles marcades com a objecte i fons alhora

L'objectiu d'aquest pas del procés és arribar a diferenciar totes les fulles en tres etiquetes:

- Objecte
- Fons
- Neutra

Per tant, les fulles amb marcadors d'objecte seran etiquetades com a nodes objecte i les fulles amb marcadors de fons seran assignades com a fons.

En aquesta primera assignació, s'assumiran com a neutres les fulles sense marcadors dibuixats a sobre.

Per últim, les fulles amb marcadors d'objecte i fons alhora es decidiran segons l'ordre dels marcadors dibuixats sobre la fulla. L'últim marcador dibuixat definirà l'etiqueta positiva o negativa.

#### **4.2.1.2 Criteris de propagació d'etiquetes**

Definim dos criteris de propagació d'etiquetes a dins de l'arbre, un cop els marcadors s'han convertit en fulles, vàlids pels dos modes proposats:

1. **Criteri d'expansió:** Per a cada node positiu, seleccionarem el seu pare sempre i quan el subarbre del seu germà tingui, com a mínim, un node seleccionat i no en tingui cap de negatiu. Aquesta és l'aportació principal de la nostra proposta. Tot i així, no és aplicable en tots els casos i, per aquest motiu, també és necessari un criteri de no expansió.
2. **Criteri de no expansió:** No expandeix l'etiqueta més enllà de la conversió de marcadors a fulles.

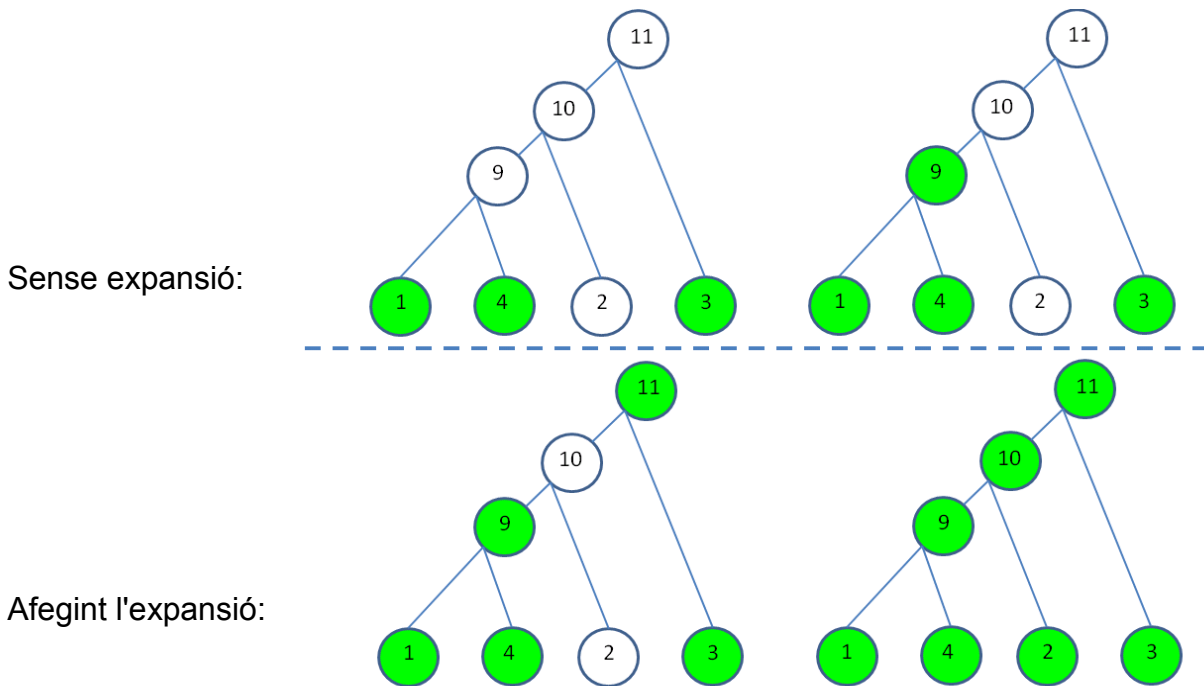


Figura 27: Criteris de propagació d'etiquetes

#### 4.2.1.3 Estudi dels marcadors i els criteris de propagació

En aquest apartat es presenta un estudi basat en un exemple teòric per tal de definir el criteri de propagació més adequat per assignar a cada mode: mode rectangle i mode traça. A més, a l'apartat de **resultats** i a **l'annex II** es poden trobar un conjunt d'experiments amb imatges reals que justifiquen les decisions preses en aquest punt de la memòria.

Aplicant el **mode rectangle sense cap expansió** a l'arbre d'exemple de les figures Figura 9 i Figura 11, el comportament esperat és el següent: com s'observa a l'arbre de la Figura 29, la fulla 6 ha estat seleccionada com a objecte ja que està continguda plenament pel rectangle dibuixat a la Figura 28. Si l'usuari no vol que formi part de la regió d'interès haurà de marcar-la amb un clic. D'aquesta manera es commutarà la fulla seleccionada. Així, la fulla 6 passa a formar part del fons.



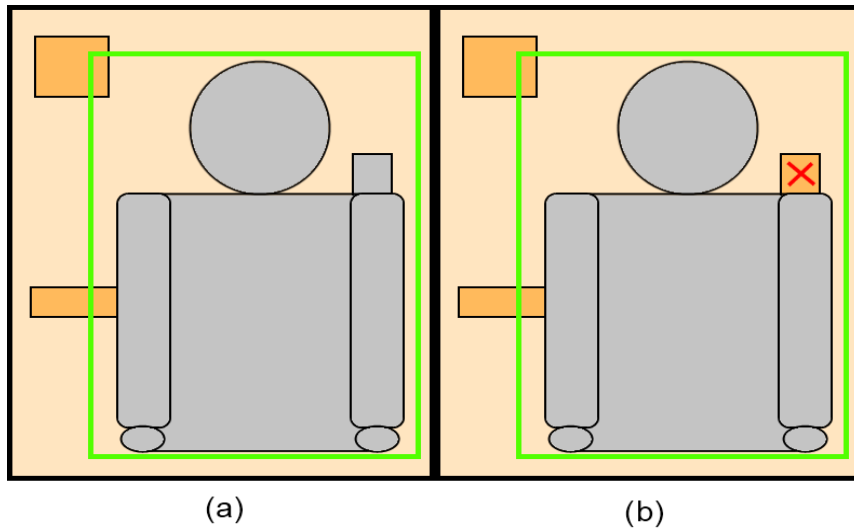


Figura 28: Exemple simple del mode rectangle sense expansió

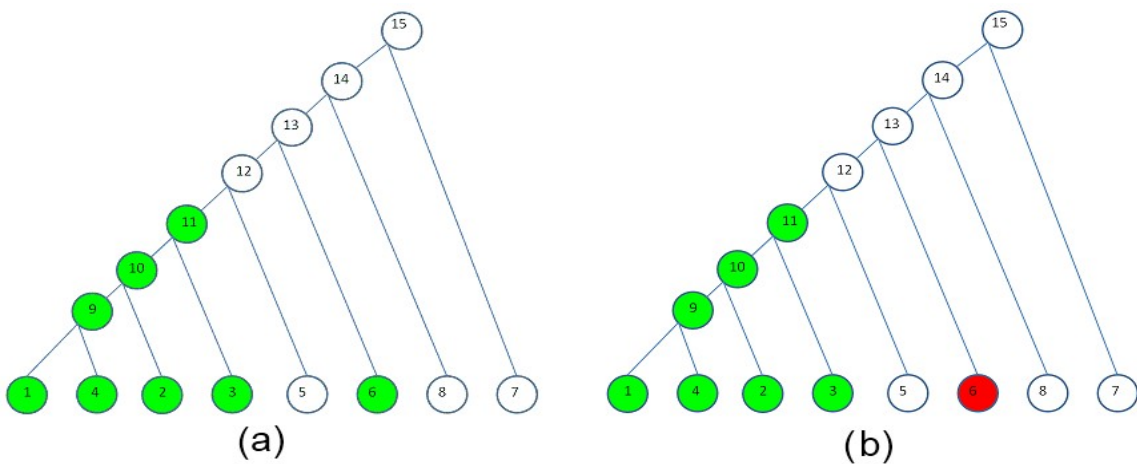


Figura 29: BPT exemple simple del mode rectangle sense expansió

Si s'afegeix la propagació per l'arbre, seguint el criteri d'expansió amb el mode **rectangle**, s'obtenen dos possibles solucions segons quina sigui la primera fulla que seleccioni l'usuari un cop dibuixat el rectangle:

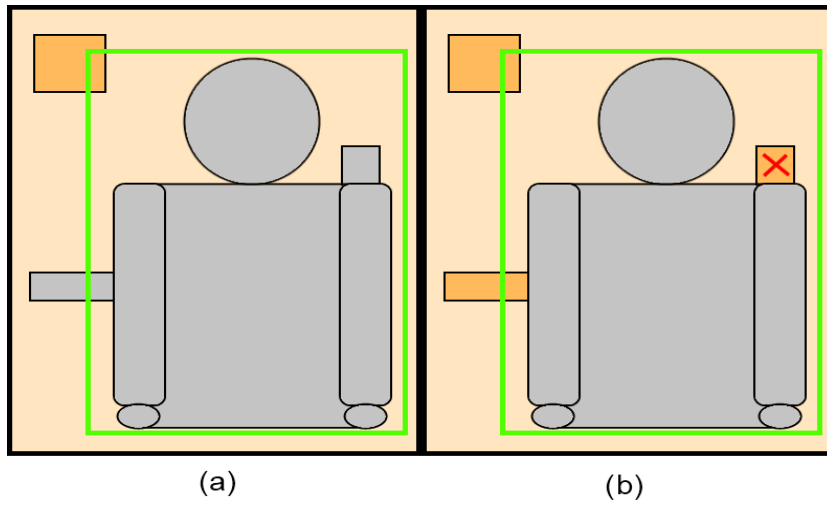


Figura 30: Primer exemple simple del mode rectangle amb expansió

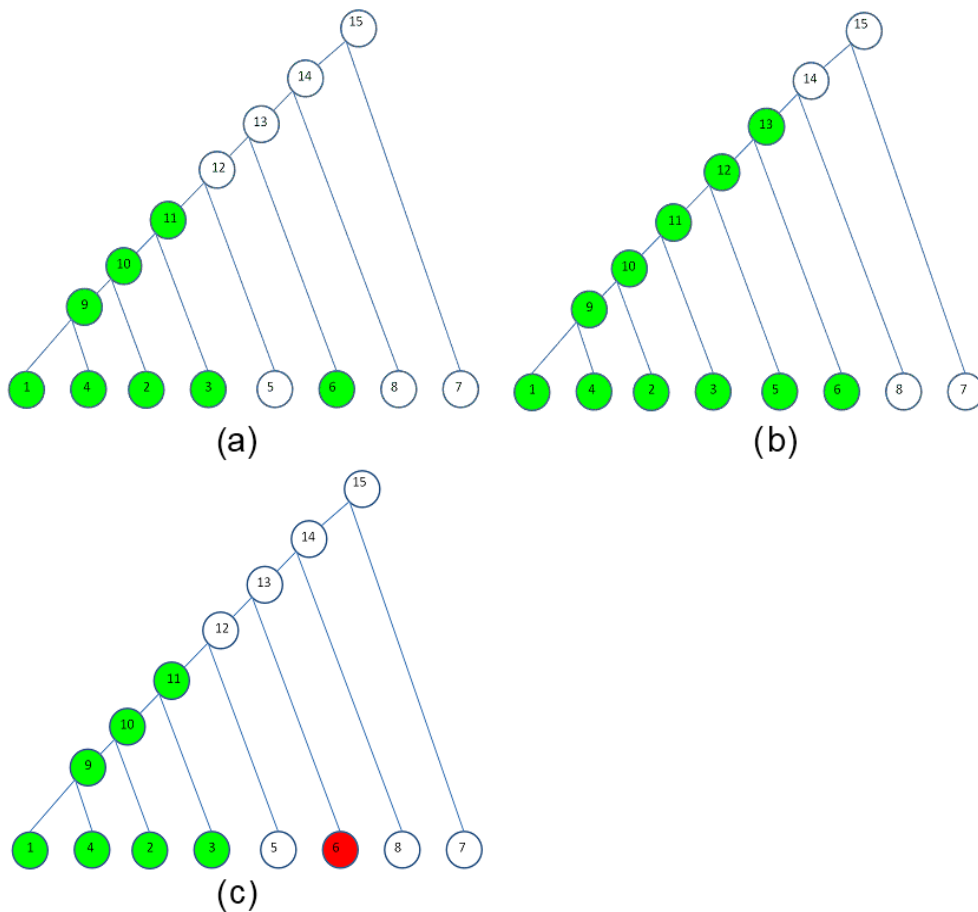
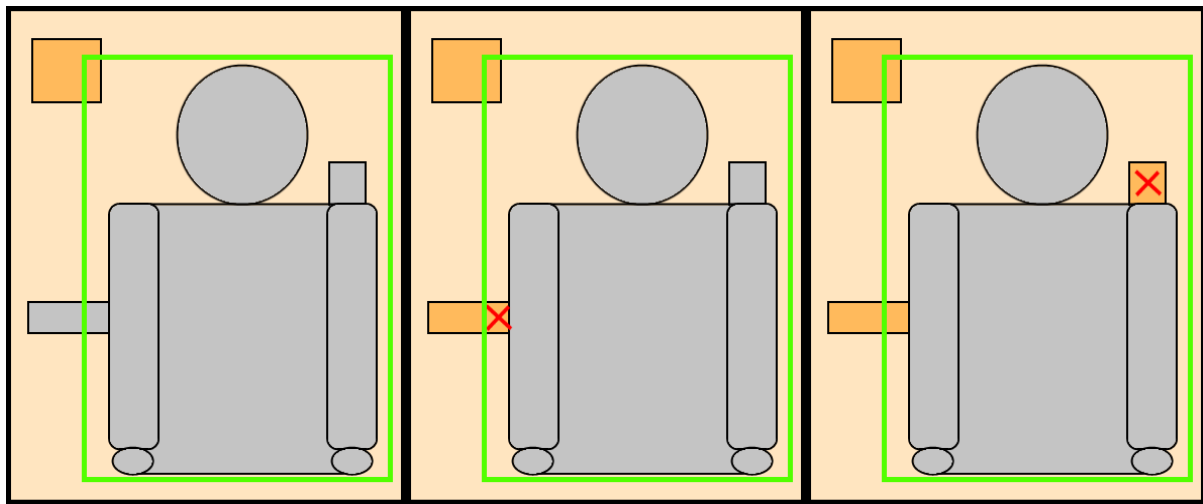


Figura 31: BPT primer exemple simple del mode rectangle amb expansió

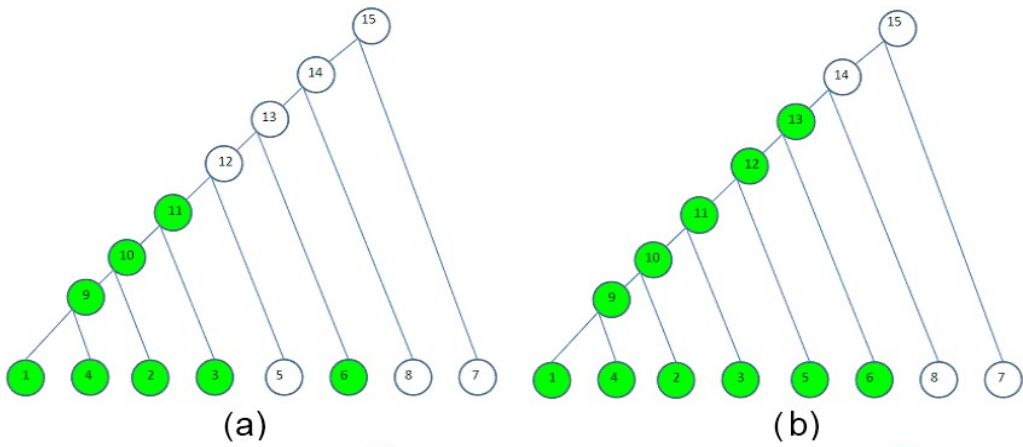


(a)

(b)

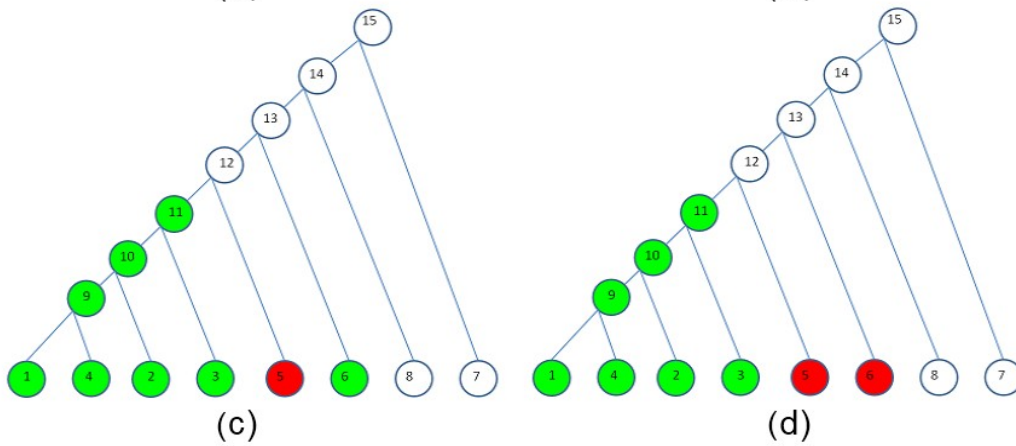
(c)

Figura 32: Segon exemple simple del mode rectangle amb expansió



(a)

(b)



(c)

(d)

Figura 33: Segon BPT exemple simple del mode rectangle amb expansió

Com s'observa en les figures Figura 30 (a) i Figura 32 (a), en tots dos casos el rectangle selecciona, de la mateixa manera que pel criteri sense expansió, la regió 6 de la imatge, ja que està continguda plenament a dins del rectangle. Però a més, en la Figura 32 (a) selecciona també la regió 5 de la partició per expansió. Totes dues fulles pertanyen al fons de la imatge.

Si l'usuari marca la fulla 6, com en el primer exemple de la Figura 30 (b), la seva acció tindrà un doble efecte, com es pot seguir a l'arbre de la Figura 31: la fulla 6 passarà a ser fons i, a més, frenarà l'expansió evitant que es propagui l'etiqueta positiva cap a la fulla 5. Finalment el nombre d'interaccions entre l'usuari i la interfície seran les mateixes que si s'apliqués el criteri sense expansió. Tot i així, amb el criteri sense expansió, el primer resultat s'aproxima millor a la regió final.

Si pel contrari, un cop dibuixat el rectangle i observat el resultat, l'usuari decideix fer un primer clic a la fulla 5, com en el segon exemple de la Figura 32 (b), només aconseguirà marcar com a fons aquesta fulla, com s'observa en l'arbre de la Figura 33 (c). Per tant, haurà de fer un segon clic per canviar l'etiqueta de la fulla 6 (c), tal com es mostra a la Figura 32 (c) i l'arbre de la Figura 33 (d).

S'ha decidit que el criteri més adequat pel mode rectangle és el **criteri de no expansió** ja que, quan l'usuari dibuixa un rectangle, ja està decidint que totes les regions contingudes a dins d'aquest formin part de la regió. Per tant, l'usuari no espera que la regió s'expandeixi més enllà de l'àrea d'aquest rectangle i, en el cas que ho volgués, podria resoldre-ho dibuixant un rectangle més gran. A més, en els exemples a on una regió no està plenament continguda en el rectangle de l'annex, es pot veure com, en molts dels casos, això es deu a una mala partició. Per aquest motiu, aplicant el criteri d'expansió, la regió s'expandeix molt cap enfora dels límits marcats pel rectangle.

En el cas del **mode traça**, el comportament observat en el mateix exemple seguint el **criteri d'expansió** es mostra en la Figura 34. L'usuari només necessita dibuixar una traça vertical per sobre de les fulles 3 i 4, que conformen el tronc i la cara del presentador. L'expansió per l'arbre mostrada a la Figura 35 propaga l'etiqueta a totes les fulles que

conformen l'objecte obtenint el resultat esperat.

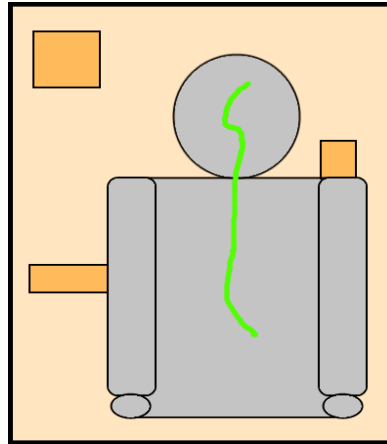


Figura 34: Exemple del mode traça amb expansió

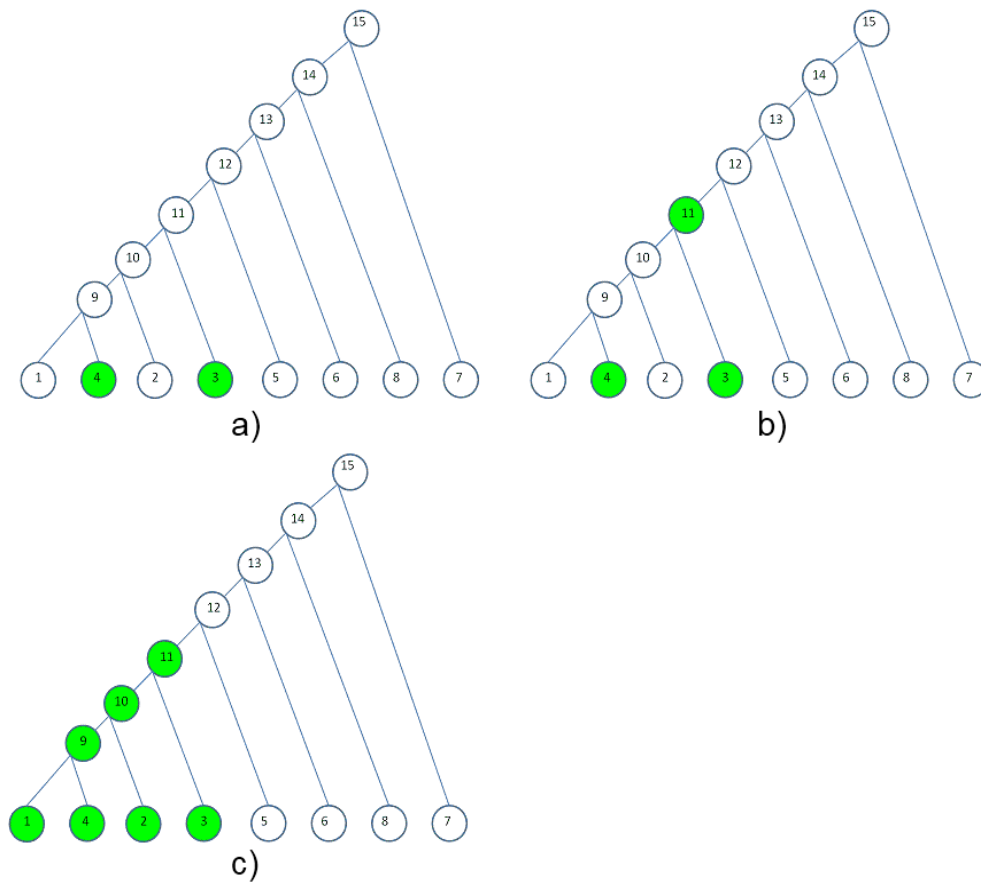


Figura 35: BPT exemple del mode traça amb expansió

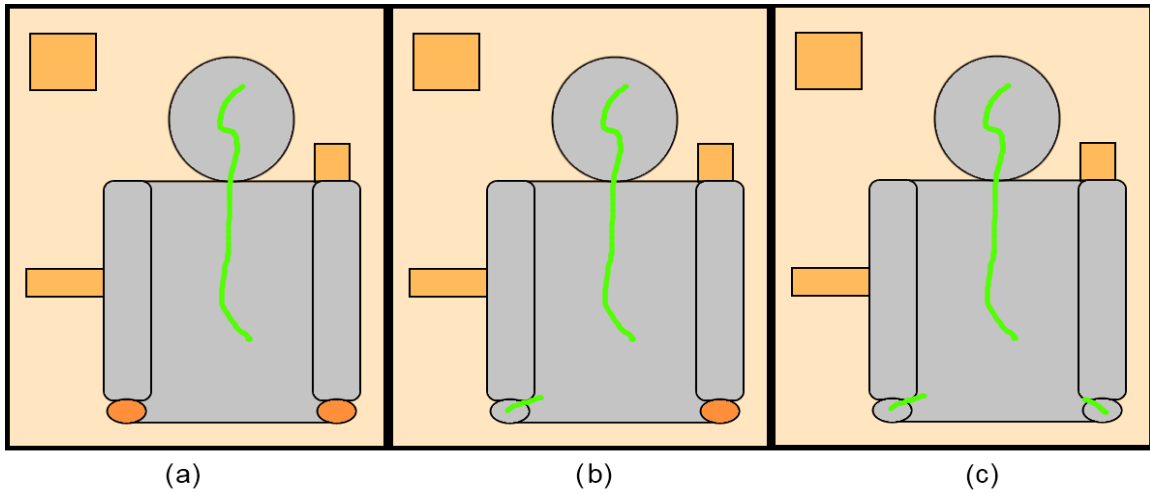


Figura 36: Exemple simple del mode traça sense expansió

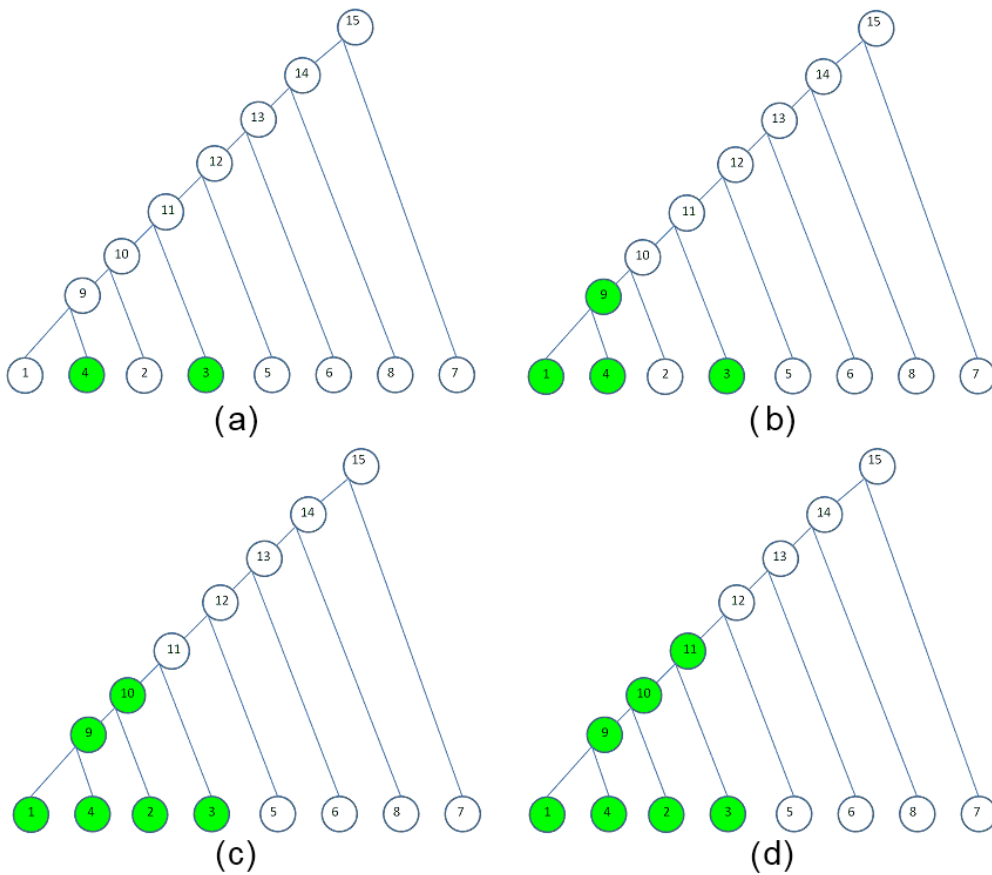


Figura 37: BPT exemple simple del mode traça sense expansió

Pel contrari, si es fa servir el mètode **sense expansió**, hi haurà petites regions que quedaran sense seleccionar i s'hauran de marcar manualment d'una en una, tal com s'observa a la Figura 36 (a). En concret, l'usuari necessita dues interaccions més (b) (c)

per seleccionar les fulles 1 i 2 de l'arbre, tal com es mostra en la Figura 37.

En el cas de la traça, contràriament al mode rectangle, l'usuari només dibuixa a sobre d'alguns dels píxels que componen cada fulla i, per aquest motiu, és molt més precís i, a causa del desconeixement de la partició inicial, l'usuari pot deixar algunes fulles sense marcar. Per aquest motiu, el criteri que millor s'adapta a aquest tipus de marcadors és el **criteri d'expansió**.

#### 4.2.2 Interfície gràfica

En aquest apartat es presenta el disseny de la interfície del Digitrace, l'eina que interactuarà amb l'usuari per dur a terme la segmentació interactiva. El Digitrace compta amb dos mòduls, com es mostra a l'esquema de la Figura 38:

- El **mòdul principal** està compost per tres elements:
  - La **imatge** que es vol segmentar amb la **màscara** incrustada a sobre d'aquesta, que permetrà distingir la regió seleccionada del fons.
  - El **botó de cerca** tanca automàticament el Digitrace i envia la informació de la imatge, els marcadors i els pesos dels descriptors cap al mòdul de la UPC perquè faci la cerca d'imatges semblants i retorni els resultats de la cerca per mostrar-los al mòdul de *keyframes* del Digitation.
  - Els botons d'**opcions** que es divideixen en quatre menús:
    - **Menú de mode** permet escollir d'entre els dos modes de segmentació interactiva: mode rectangle i mode traça.
    - **Menú d'objecte o fons**, a on l'usuari pot escollir entre marcadors d'objecte i marcadors de fons.
    - **Menú d'edició d'imatges** amb el qual l'usuari pot esborrar un marcador, tornar enrere o tornar a començar de zero.

- **Menú de configuració:** aquests és el menú més complet ja que interactua amb moltes fases diferents de l'eina. És aquest menú el que obre el mòdul secundari del Digitrace.

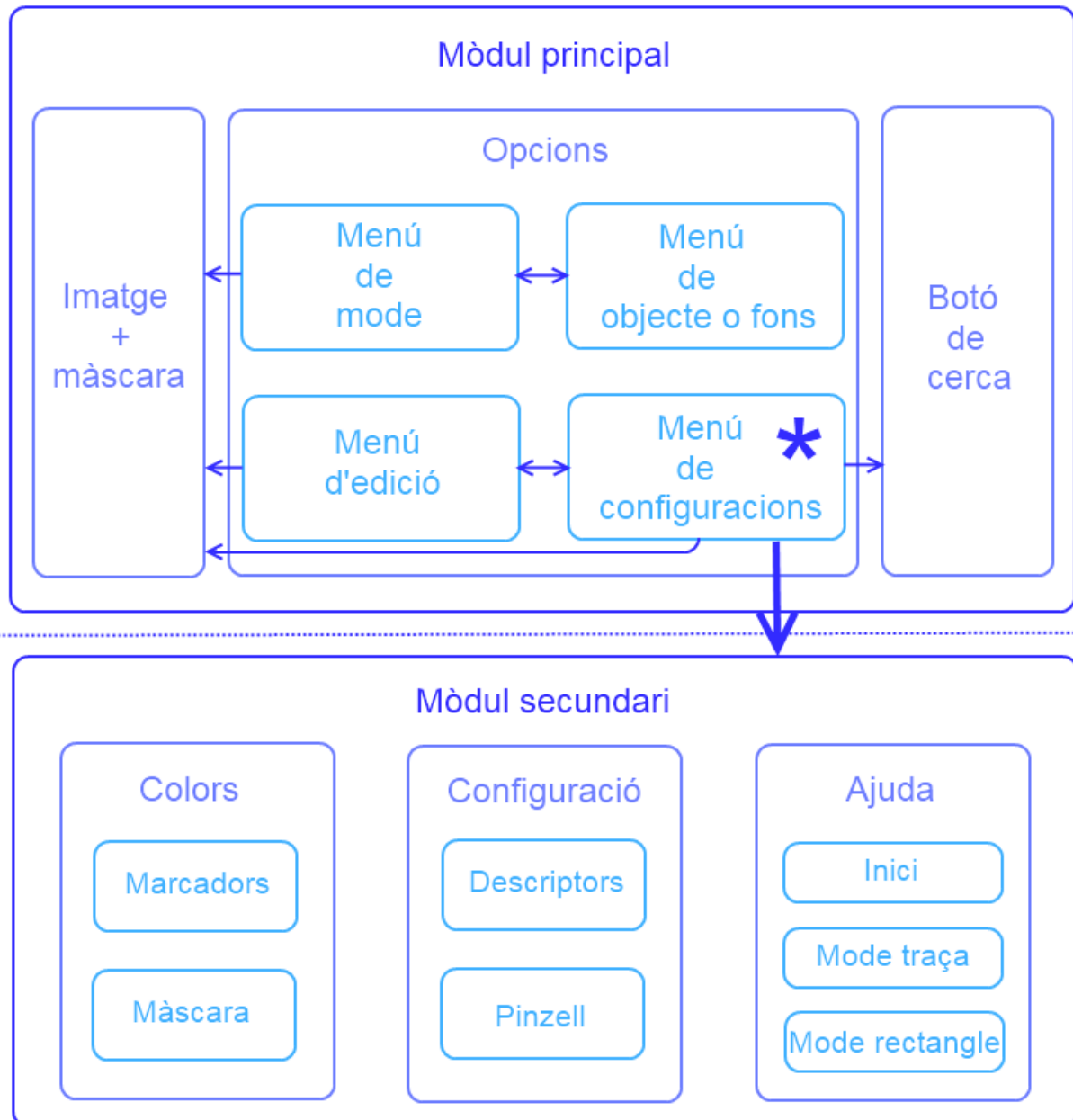


Figura 38: Esquema de les funcionalitats del Digitrace

- El **mòdul secundari** també està format per tres elements principals:
  - El mòdul de **colors** és l'encarregat de dotar a l'usuari de llibertat per canviar els



colors dels **marcadors** i de la **màscara**.

- El mòdul de **configuració** permet fer configuracions que afecten a dues fases diferents de l'eina. La configuració dels **descriptors** condiciona la cerca dels resultats, per tant, actua en la última fase del Digitrace. Per l'altre banda, la configuració del **pinzell** actua sobre el menú d'edició permetent variar els gruixos dels marcadors.

- L'**ajuda** és una pàgina que guia a l'usuari en la selecció de la regió explicant pas per pas cadascuna de les tècniques de segmentació interactiva.

### **4.3 Comunicació**

Darrera d'aquesta capa de presentació es troben els servidors de la CCMA i de la UPC, els quals segueixen un model d'**arquitectura distribuïda** motivats per les tendències organitzatives actuals. A més, per dur a terme l'intercanvi de les dades necessàries per a la comunicació entre ells i amb el client web fan ús d'un conjunt de protocols i estàndards que es defineixen com a **servei web**.

#### **4.3.1 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. Aquest tipus de sistema ha de complir una sèrie de característiques:

- **Tolerància als errors:** si un component a dins del sistema es descomposa, un altre ha de ser capaç de substituir-lo.
- **Integrabilitat** de dades i aplicacions.
- **Modificabilitat:** en ser un sistema format per màquines independents, les aplicacions i funcions també ho seran facilitant, així, la modificació d'alguna d'elles sense afectar a la resta.
- **Escalabilitat:** cada mòdul pot créixer independentment per adaptar-se a la situació.

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

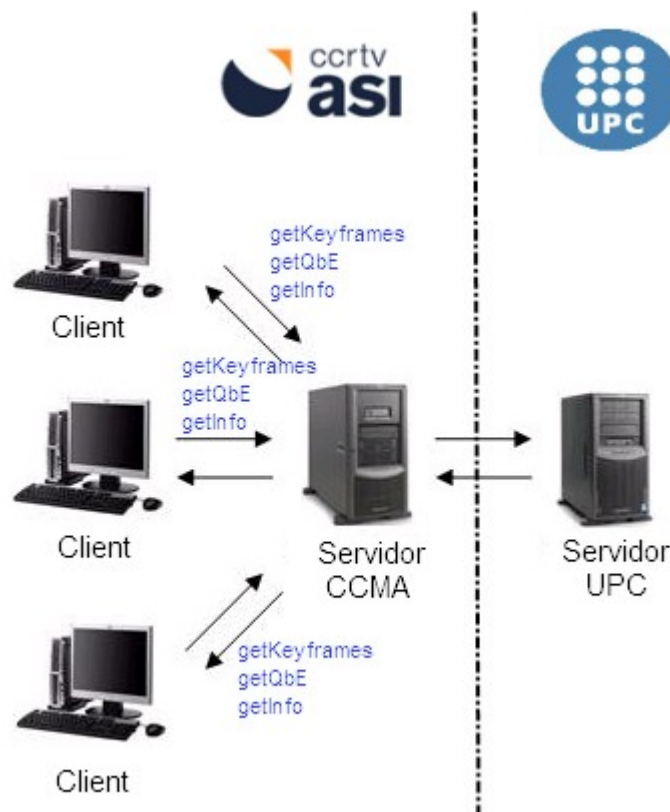


Figura 39: Estructura Client - Servidor

- 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**, per la seva banda, es comunica únicament amb el servidor de la CCMA, el qual fa les peticions de cerca. La UPC processa aquestes peticions i retorna els resultats al servidor de la CCMA.

### 4.3.2 Servei web

Cadascun dels servidors esmentats proporciona un **servei web** al seu client directe. Un servei web consisteix en un conjunt de protocols i estàndards que serveixen per intercanviar dades entre diferents aplicacions que poden estar desenvolupades en llenguatges de programació diversos i executades des de qualsevol plataforma mitjançant xarxes d'ordinadors, en aquest cas **Internet**.

Els serveis web, per tant, aporten **interoperabilitat** entre aplicacions provinents de diferents plataformes i, en ser accessibles via Internet, no limiten la ubicació geogràfica del servidor respecte el client: en la nostra aplicació el servidor de la UPC es troba a Terrassa i el client a Esplugues. Una altra característica fonamental és el foment dels estàndards i protocols basats en text, els quals fan més fàcil l'accés al contingut i l'enteniment entre client i servidor.

## 5 Desenvolupament

### 5.1 Entorn de desenvolupament

En aquest apartat es presentaran, a grans trets, els llenguatges i l'entorn de desenvolupament emprats per dur a terme l'aplicació del client web. Cal comentar que, en aquest projecte, el desenvolupament de codi s'ha limitat a la part client (Digimatge i Digitrace), mentre que els serveis de segmentació interactiva i cerca els ha desenvolupat la UPC.

#### 5.1.1 Marc de desenvolupament: Flex

**Adobe Flex** és un marc de desenvolupament que agrupa diferents tecnologies publicades al Març de 2004 per Macromedia, basant-se en la seva plataforma propietària Flash. L'objectiu de Flex és permetre als desenvolupadors d'aplicacions web construir de forma fàcil i eficaç Aplicacions Riques d'Internet (**RIAs**). Aquest tipus d'aplicacions tracten de combinar els avantatges de les aplicacions web i les aplicacions tradicionals.

Normalment, en les aplicacions web, hi ha una descàrrega contínua de la pàgina cada vegada que l'usuari interactua amb la interfície i això dona lloc a un trànsit molt elevat entre el client i el servidor i, en conseqüència, a una limitació de la interacció. Els entorns RIA carreguen tota l'aplicació al començament i només es comuniquen amb el servidor per demanar un recurs o una execució. A més a més, les RIA pretenen oferir a l'usuari capacitats multimèdia molt més sofisticades donant lloc a una millora en l'experiència visual i fent més agradable la utilització de les aplicacions prestant una major interacció. Això converteix Adobe Flex en el marc idoni per desenvolupar la nostra aplicació ja que s'adapta al model client-servidor del que partim.

Aquest projecte s'ha desenvolupat prioritàriament amb la versió 3.0 de Flex tot i que, en certs moments, ha calgut fer una adaptació de part del codi cap a la nova versió Flex 4.0 alliberada<sup>11</sup> l'abril de 2010.

El **codi font** de Flex està format per arxius del tipus MXML i ActionScript:

**ActionScript** és la opció tradicional pels desenvolupadors, un llenguatge de programació orientat a objectes que s'encarrega del control de la lògica a la banda del client.

Per la seva banda, **MXML** és un llenguatge descriptiu basat en XML (*Extensible Markup Language*) que permet construir interfícies visuals de manera ordenada i intuïtiva, crear i estendre components visual i definir l'apariència i el comportament de l'aplicació Flex.

MXML permet crear *composites*, que són patrons per a construir objectes complexos a partir d'altres més simples seguint una estructura jeràrquica en forma d'arbre. Per tant, les aplicacions Flex parteixen d'un node principal, que s'anomena **Application**, i de diferents nodes que pengen d'aquest o dels seus fills, que seran els diferents components visuals: botons, llistes desplegable, menús, camps de text, taules, etc.

A banda d'això, MXML també facilita la creació de *widgets*, és a dir, de petites aplicacions o programes independents que es poden reutilitzar. Aquests resulten molt útils en funcions o components visuals utilitzats amb freqüència.

El codi font de Flex compila conjuntament els arxius amb extensió .MXML i .as segons el codi de bits Flash, donant lloc a un únic fitxer .SWF, que s'executa en la màquina virtual d'ActionScript continguda en el connector per Flash Player o per AdobeAIR que trobem en el client. Els connectors Flash Player i Adobe AIR són multi plataforma i, conseqüentment, permeten treballar amb gran varietat de hardwares,

---

11 <http://opensource.adobe.com/wiki/display/flexsdk/Download+Flex+4>

sistemes operatius i navegadors, la qual cosa proporciona als desenvolupadors de Flex una gran flexibilitat. En en nostre cas, hem creat una aplicació Flex mitjançant l'eina propietària Adobe Flash Builder.

### 5.1.2 Entorn: Adobe Flash Builder

**Adobe Flash Builder** és un entorn de desenvolupament integrat construït sobre la plataforma d'Eclipse per crear RIAs i aplicacions multi plataforma d'escriptori particularment per a la plataforma Flash.

El **desenvolupament d'una aplicació Flex** està condicionat per les funcions assignades a cadascun dels llenguatges (MXML i ActionScript) utilitzats paral·lelament. Podem dividir-lo en diverses **etapes** de desenvolupament independents, les quals s'ordenen cronològicament per obtenir un rendiment òptim:

1. Definició de la interfície de l'aplicació amb el conjunt de components visuals pre-definits que ofereix Flex (botons, taules, gràfiques, efectes d'animació, etc.).
2. Disseny de la interfície: ordenar els components que formen l'aplicació.
3. Utilitzar els estils i temes per definir el disseny visual.
4. Afegir comportament dinàmic (interacció entre els diferents components).
5. Definir i connectar els serveis web per intercanvi de dades (*HTTPService*).
6. Compilar el codi font en un arxiu .SWF que funcioni en qualsevol reproductor Flash.

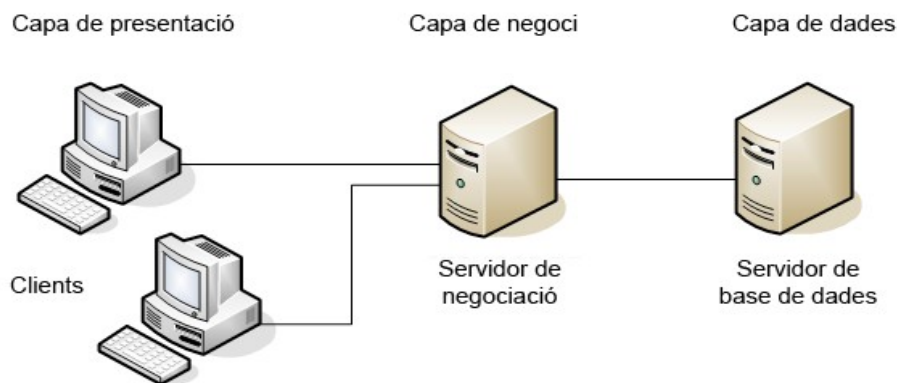
Aquest projecte s'ha desenvolupat amb la versió d'Adobe Flash Builder 4, la qual suporta el codi escrit tant en Flex 3 com en Flex 4, facilitant, així, la utilització de tots dos a la vegada segons les necessitats de l'usuari.

### 5.1.3 Model multi-capa

El model multi-capa és la implementació més comuna d'una arquitectura de tipus **distribuïda**. Flex està dissenyat per treballar en models multi-capa a on les aplicacions

Flex es troben al nivell de **presentació**. L'objectiu principal d'un model multi-capa consisteix en la separació de la lògica de negocis i la lògica de disseny. Per tant, distingim 3 capes:

1. **Capa de presentació**, també anomenada capa d'usuari, és la que es troba aquest quan fa us de l'aplicació. Aquesta capa comparteix les característiques de flexibilitat i usabilitat de les RIA. Es comunica amb la immediatament inferior: la capa de negoci.
2. **Capa de negoci**: on resideixen els programes que s'executen i on es reben les peticions d'usuari. En el nostre cas, aquesta capa es troba dividida entre el servidor de la CCMA i el de la UPC.
3. **Capa de dades**: l'encarregada d'emmagatzemar totes les dades. En la nostra aplicació, aquesta capa la constitueix el repositori de dades de la Corporació, a la qual accedim des del servidor (capa de negoci), des del qual es fan les peticions de recuperació o emmagatzematge d'informació.



*Figura 40: Model multi - capa*

## 5.2 Arquitectura

Com s'ha introduït a l'apartat de disseny, partim de dos canals de comunicació independents que s'encadenen per arribar des del servidor de la UPC fins al client web passant pel servidor de la Corporació. 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 a més, tots dos servidors segueixen una arquitectura del tipus **REST** servint-se del protocol HTTP entre d'altres.

### 5.2.1 Protocol HTTP

HTTP és el protocol emprat per a fer qualsevol transacció a la World Wide Web. Aquest protocol està orientat a les **transaccions client-servidor** i segueix un esquema de petició-resposta entre tots dos. També és important tenir en compte que HTTP és un protocol sense estat, és a dir, sense memòria per guardar cap informació de 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**. Aquesta informació 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 s'ha retornat.

#### 5.2.1.1 Sintaxi d'una URL per HTTP

En un servei web, la petició d'un recurs s'efectua seguint el **model URL** (*Uniform Resource Locator*). 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 de 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 a posicions a dins d'una mateixa pàgina.

#### 5.2.1.2 Mètodes HTTP

A més 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 Digimatge ha requerit de dos d'aquests mètodes: **GET** i **POST**. El mètode GET s'ha utilitzat per a 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, pel contrari, només s'ha utilitzat per un mode concret del Digitrace, degut a la necessitat d'enviar una màscara de la traça dibuixada pel client.

### 5.2.2 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 unequivoca** 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.

### 5.2.3 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, i 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.

Les crides client – servidor es fan de forma **asíncrona**. En contraposició a les transferències síncrones, a on el canal de transmissió queda assignat en permanència de manera que aquest canal estarà dedicat a una crida en concret, la millora en les crides asíncrones resideix en que aquestes eviten el bloqueig del client mentre espera el resultat d'una petició i, així, s'optimitza la usabilitat de l'aplicació i el client guanya llibertat d'actuació a costa d'un augment de dificultat en la gestió de les crides.

#### 5.2.3.1 Crides HTTP client – servidor

Un **objecte de valor** (Value Object) consisteix en un patró de disseny que agrupa diversos valors a dins d'un objecte únic per tal de facilitar el seu enviament i rebuda proporcionant una major comoditat i seguretat.

Els objectes de valor principals que s'han utilitzat en aquest projecte són dos:

KeyframeVO, que representa a un *keyframe* i AssetVO, que representa a un àssset.

S'ha definit el concepte '**asset**' com un objecte digital amb el qual la Corporació defineix els seus vídeos. L'asset agrupa un fitxer de vídeo, un fitxer d'àudio i diverses metadades i identificadors que el descriuen. L'objecte de valor **AssetVO** compta amb diferents variables i mètodes:

Variable	Tipus	Descripció
Asset	String	Identificador únic per a cada àssset del repositori de la CCMA.
keyframes	ArrayCollection	Array amb tots els <i>keyframes</i> de l'àssset del tipus KeyframeVO.
key	Boolean	True quan el ratolí es troba a sobre d'aquest <i>keyframe</i> o d'un <i>keyframe</i> del mateix àssset.

Mètode	Paràmetres d'entrada	Descripció
Constructor	AssetId	Còpia el valor de l'identificador a la variable <i>asset</i> i crea un ArrayCollection de <i>keyframes</i> buit.

Un **keyframe**, per la seva banda, és una imatge clau en un instant determinat d'un àssset en concret. Per tant, està contingut a dins d'un àssset. El **KeyframeVO** ve definit per una sèrie de variables i mètodes:

Variable	Tipus	Descripció
assetId	String	Identificador únic per a cada àssset del repositori de la CCMA.
timeCode	String	Codi de temps del <i>keyframe</i> a dins de l'àssset: hh:mm:ss:ff.
imageUrl	String	Ruta de la imatge en concret.
videoUrl	String	Ruta del vídeo.
image	String	ByteArray de la imatge codificada.
score	String	Ranking en el cas que el <i>keyframe</i> sigui el resultat d'una consulta per imatge.

numKey	Int	Color del requadre que envolta el <i>keyframe</i> definit segons l'àsset al qual pertany.
key	Boolean	<i>True</i> quan el ratolí es troba a sobre d'aquest <i>keyframe</i> o d'un <i>keyframe</i> del mateix àsset.
maxValue	Number	Màxim score del resultat al qual pertany el <i>keyframe</i> per normalitzar el ranking que es mostra a l'usuari.

Mètode	Paràmetres d'entrada	Descripció
Constructor	assetId timeCode imageUrl videoUrl numKey score maxVal	Còpia els valors dels paràmetres d'entrada als paràmetres del propi objecte.  Abans de copiar-lo, el valor de <i>score</i> passa per un petit tractament per transformar-lo en un percentatge amb un rang [0 , 100] i es normalitza.

Les **crides** que es duen a terme entre el client web i el servidor de la CCMA fan ús d'aquests objectes de valor per facilitar les transaccions de dades. Distingim quatre crides principals:

Crida	Mètode HTTP	Paràmetres d'entrada	Tipus	Paràmetres de sortida	Tipus
<b>getKeyframes</b>	GET	assetId	String	Llistat amb els <i>keyframes</i> de l'àsset.	Array String del tipus KeyframeVO
<b>GetQbE: per imatge o per regió</b>	GET / POST	assetId timeCode descriptors	String String Array String		
<b>GetQbE: Cerca de Text</b>	GET	assetId timeCode	String String	Keyframe descriptionText	KeyframeVO Array amb cada entitat detectada i les coordenades a dins de la imatge.
<b>getInfo</b>	GET	Ids	Array String	Títols de cada àsset	Array String

En el cas de la detecció de text, la Corporació disposa, dins del propi sistema, del mòdul que duu a terme el processat. Per tant, la crida es limita a una comunicació entre el client web i el servidor de la CCMA.

La crida *getQbE* s'utilitza per a tots els mètodes de consulta per imatge, ja sigui per imatge global o per regió independentment del mode (rectangle o traça). No obstant, els paràmetres d'entrada i sortida varien en cada cas i cal definir-los.

A més, la crida per regions es divideix en dues diferents, que seran distingides pels paràmetres d'entrada.

1. **Obtenció de la màscara:** aquesta es farà tantes vegades com l'usuari ho disposi.
2. **Obtenció de resultats** un cop l'usuari està satisfet amb la regió marcada i es farà una última vegada donant per finalitzada la cerca i tancant el Digitrace per mostrar els resultats al mòdul de keyframes del Digiton.

En concret, s'ha establert que els 'descriptors' siguin el paràmetre d'entrada que marqui quina consulta està fent el client web. Si aquest paràmetre està buit, la crida s'executa per obtenir una màscara. Si, pel contrari, el paràmetre està ple, s'esperaran els resultats de la cerca per regió.

GetQbE: per imatge	GET	assetId timeCode descriptors mode	String String Array String String	Llistat amb els keyframes més semblants a la imatge o regió exemple	Array String del tipus KeyframeVO
GetQbE: mode rectangle 1	GET	assetId timeCode mode coordenades	String String String Array String	Màscara que separa la regió de l'objecte seleccionat	Imatge JPG en blanc i negre
GetQbE: mode rectangle 2	GET	assetId timeCode mode coordenades descriptors	String String String Array String Array String	Llistat amb els keyframes més semblants a la regió exemple	Array String del tipus KeyframeVO
GetQbE: mode scribble 1	POST	assetId timeCode mode label Imatge bpt_neg	String String String String Array String Array String	Màscara que separa la regió de l'objecte seleccionat	Imatge JPG en blanc i negre

		bpt_pos	Array String		
GetQbE: mode scribble 2	GET	assetId timeCode mode bpt_neg bpt_pos	String String String Array String Array String	Llistat amb els keyframes més semblants a la regió exemple	Array String del tipus KeyframeVO

La crida de la màscara en el mode traça és diferent a la resta perquè requereix l'enviament d'una imatge des del client cap al servidor. Això obliga al client a fer una crida del tipus Post.

Quan es va pensar com implementar el servei de consulta per regions amb el mode traça, es van estudiar diferents opcions per a la comunicació entre el client i el mòdul de processat. De bon començament, el problema més evident va ser l'eficiència de la crida. Una traça correspon a un número molt gran de píxels dins d'una imatge i, per tant, s'havia de pensar molt bé com fer les crides tenint en compte que es treballa amb un sistema sense memòria, a on cada crida ha de contenir tota la informació necessària per calcular la regió des de zero. Es va considerar poc efectiu enviar tots els punts que conformaven la traça per coordenades, com en el cas del rectangle. La primera solució va consistir en l'enviament d'una imatge de dos colors amb totes les traces positives dibuixades i una altra imatge de dos colors amb totes les traces negatives.

Més endavant, es va trobar un segon impediment: el mode traça funciona diferent segons l'ordre en que es dibuixen les traces. Això es deu a que una mateixa fulla pot tenir més d'una traça dibuixada a sobre. Si aquestes traces no pertanyen totes a la mateixa etiqueta, l'algoritme necessita conèixer quin ha estat l'últim marcador dibuixat. Per tant, el fet de considerar totes les traces alhora no era factible. Aquest problema s'ha resolt mitjançant els dos conjunts de *String* **bpt\_pos** i **bpt\_neg**. En aquests *arrays* es guarda cadascuna de les fulles que formen part de l'objecte (bpt\_pos) i les que formen part del fons (bpt\_neg) i s'actualitzen per a cada nou marcador dibuixat. Aquests *arrays* s'envien cap al client, que fa la funció de memòria. Així, per cada marcador dibuixat, el client retorna cap al servidor els dos *arrays* bpt\_pos i bpt\_neg juntament amb una imatge de dos colors únicament amb la última traça dibuixada (Image) i una etiqueta per definir si es tracta d'un marcador d'objecte o de fons (label).

Així, el mòdul de processat pot calcular ràpidament l'actualització de les fulles segons l'últim marcador dibuixat mantenint les etiquetes de les fulles que no han estat tocades per l'últim marcador, que són recuperades mitjançant els *arrays* `bpt_pos` i `bpt_neg`.

#### 5.2.4 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. La CCMA demana resultats sobre una imatge, ja sigui per una consulta basada en imatge o en regió o bé la detecció de Text seguint un model de crida molt semblant al presentat anteriorment. Seguidament, la UPC processa aquesta imatge i retorna uns resultats. En aquest moment, el client de la CCMA passa a actuar com a servidor del nostre client web, adequa la resposta i la envia cap al client.

#### 5.2.5 Implementació amb Flex: HTTP Service

HTTPService és el component que ofereix Flex per crear canals de comunicació entre el client i el servidor mitjançant el protocol HTTP. La funció *send()* fa una crida HTTP a una URL especificada al paràmentre *url* i rep una resposta HTTP que serà processada per la funció especificada a *result()*.

Flex ofereix set dels diferents **mètodes** HTTP: `get`, `post`, `head`, `options`, `put`, `trace`, `delete`.

També proposa diverses opcions pel mode de comunicació asíncron, que anomena **concurrència**. Aquestes són:

- **Multiple**: Aquest és el mode que utilitzem i el mode per defecte. Quan la concurrència és múltiple, el desenvolupador és el responsable de gestionar les



respostes a les diferents crides paral·leles.

- **Single:** Fins que no es rebin els resultats corresponents, la primera crida és la única vàlida. Les crides següents generaran un missatge d'error.
- **Last:** Cada petició cancel·la les peticions anteriors.

## 6 Resultats

### 6.1 Estudi comparatiu dels algorismes de segmentació interactiva

#### 6.1.1 Avaluació de les tècniques de SRG, SIOX i BPT

Per començar, a l'estat de l'art s'ha fet un estudi de tres tècniques de segmentació interactiva: SRG, SIOX i BPT. En aquest apartat de resultats es faran dos experiments. El primer consistirà en la selecció d'una regió amb les diferents tècniques esmentades, així podrem comparar el nivell d'interactivitat necessari en cada cas per a un mateix resultat. El segon experiment consistirà en observar els diferents resultats que dona cada tècnica per a una mateixa imatge i uns mateixos marcadors. D'aquesta manera quedarà palesa l'eficiència de cadascun d'ells.



*Figura 41: Exemple real*

Aplicant les tècniques de SRG i SIOX a l'exemple real de la Figura 41, i suposant que es vol seleccionar la cara de la presentadora, els resultats no són tan impecables com en l'exemple homogeni de les figures Figura 6 i Figura 7 que es mostra a l'estat de l'art. Això es deu a que aquestes tècniques es basen únicament en el color i això resulta

bastant limitant per a imatges més complexes.

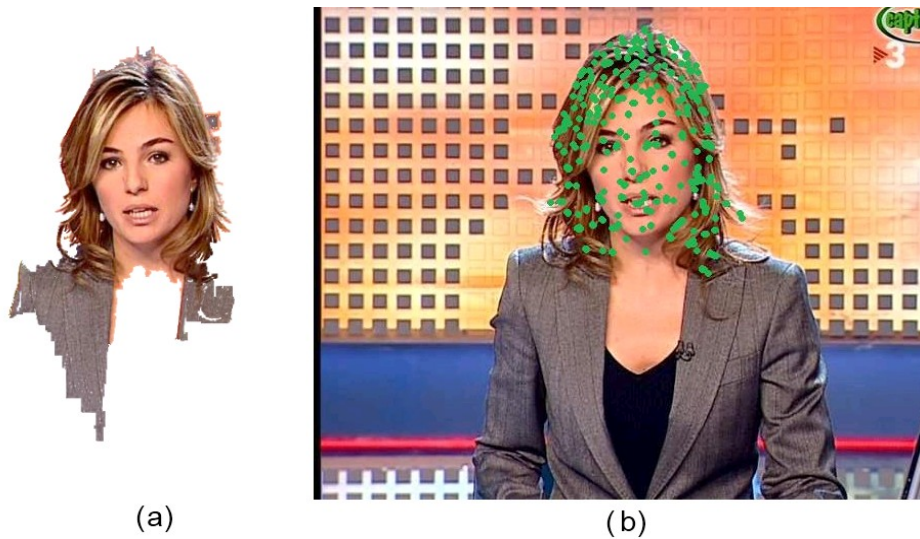


Figura 42: Exemple real amb SRG

Aplicant l'eina de creixement de regions a partir de llavors (SRG) un únic cop s'arriba a resultats nuls. Per arribar a obtenir el resultat mostrat en la Figura 42 (a) es requereix d'un esforç per part de l'usuari, el qual s'ha de dedicar a marcar (b) regions molt petites per conformar la regió definitiva.

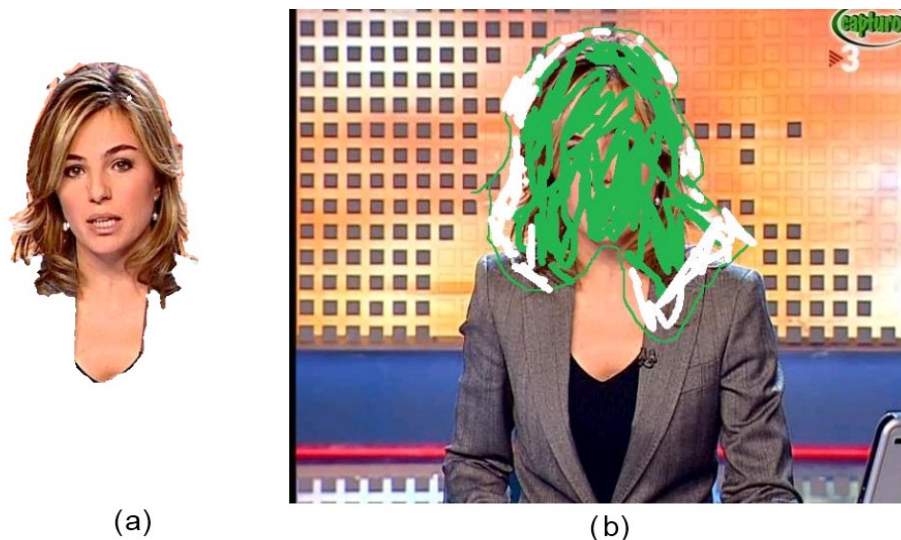


Figura 43: Exemple real amb SIOX

Amb l'eina d'extracció interactiva d'objectes mitjançant color (SIOX), Figura 43, resulta més senzill trobar un primer resultat raonable però és necessari dedicar-hi una estona per retallar millor el contorn, com s'observa amb les pinzellades blanques (b).

Ambdues tècniques requereix un nombre de marcadors molt elevat si la imatge és complexa. En canvi, amb la tècnica d'arbres binaris de particions, com s'observa en la Figura 44, els marcadors necessaris per obtenir la regió de la Figura 45 són bastant més reduïts, permetent una interacció menor de l'usuari.



*Figura 44: Traces amb IST*



*Figura 45: BPT amb IST*



*Figura 46: SRG amb IST*



*Figura 47: SIOX amb IST*

*La Interactive Segmentation Tool (IST)*, té la capacitat d'utilitzar totes aquestes tècniques. Això permet dur a terme el següent experiment: Utilitzant l'eina i marcant les regions per BPTs trobem, amb un petit número de traces dibuixades a la Figura 44, el resultat que s'observa a la Figura 45. Canviant l'algoritme però conservant les traces, veiem que els resultats són molt menys precisos amb les eines de SRG i SIOX, Figura 46 i Figura 47, respectivament. Aquesta imprecisió obliga a una major interacció de l'usuari, que haurà d'augmentar el nombre de traces.

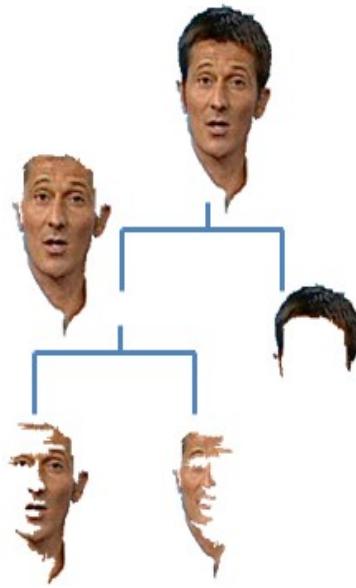
### 6.1.2 **Avaluació dels modes rectangle i traça**

Un cop ben definits cadascun dels dos modes de segmentació interactiva, sembla una bona idea experimentar una mica amb cadascun d'ells. S'han fet proves amb tres tipus de regions diferents: cares, objectes i logotips.



*Figura 48: Exemple de regió homogènia*

En general, es pot concloure que el mode traça funciona satisfactòriament quan les regions objecte són bastant homogènies i ben diferenciades del fons, com en l'exemple de la Figura 48 i l'arbre de la Figura 49.



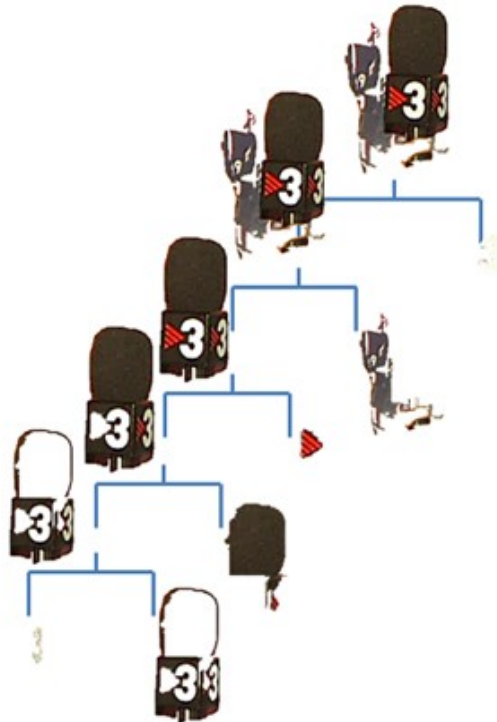
*Figura 49: Exemple d'arbre ben diferenciat del fons*

A més, els resultats milloren si l'arbre està ben fet i la regió que es vol separar del fons no es barreja en cap moment amb aquest fons. Això es deu a que el mètode traça utilitza un criteri d'expansió que fa que l'error es propagui més ràpidament.



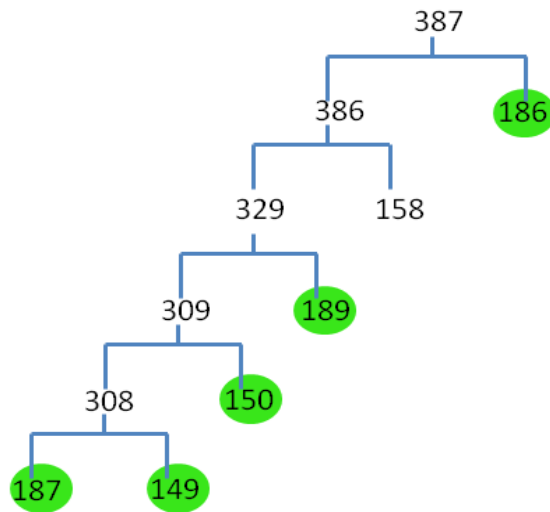
*Figura 50: Exemple de regió dispersa amb mode el rectangle*

En la Figura 50 s'observa una fotografia i un objecte d'interès: un micròfon. Seguint la premisa del mode rectangle, es comença per dibuixar un rectangle que contingui l'objecte. A la Figura 51 es poden veure les diferents segmentacions de l'objecte i com s'uneixen a dins de l'arbre de particions. S'observa com la fulla 158 embruta la partició del micròfon però, degut a que el rectangle no conté tota aquesta regió, la segmentació queda ben feta sense necessitat de dibuixar més marcadors. A la Figura 52 s'observen els nodes seleccionats en color verd.



*Figura 51: Exemple d'arbre amb una fulla que embruta la regió d'interès*

Pel contrari, a la Figura 53, a on s'utilitza el mode traça per seleccionar la mateixa regió, la propagació arriba fins a la fulla 158, que no forma part de la regió d'interès. Per tant, s'observa que el comportament del mode traça empitjora per a arbres que contenen fulles que no interessin barrejades entre les fulles que sí es volen seleccionar a dins l'arbre.



*Figura 52: Fulles seleccionades*



*Figura 53: Exemple de regió dispersa amb el mode traça*

Conseqüentment, amb arbres molt dispersos formats per moltes regions petites a on es barregen les fulles que componen la regió objecte amb les altres fulles com en la Figura 54, el mode rectangle dona resultats molt superiors.



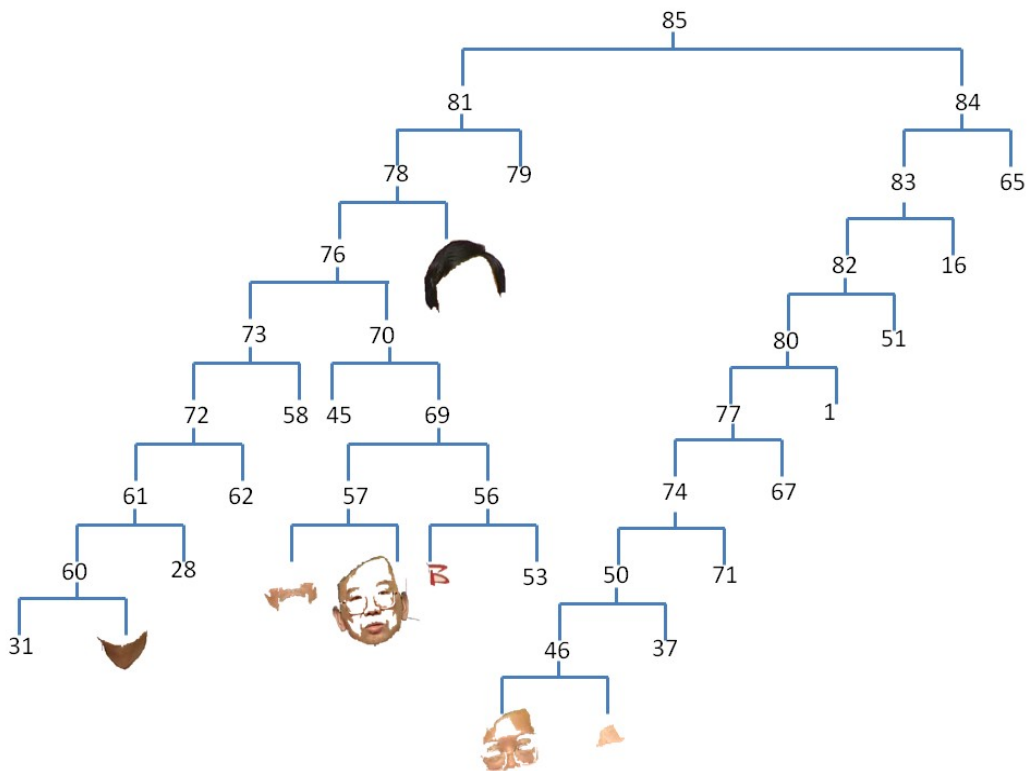
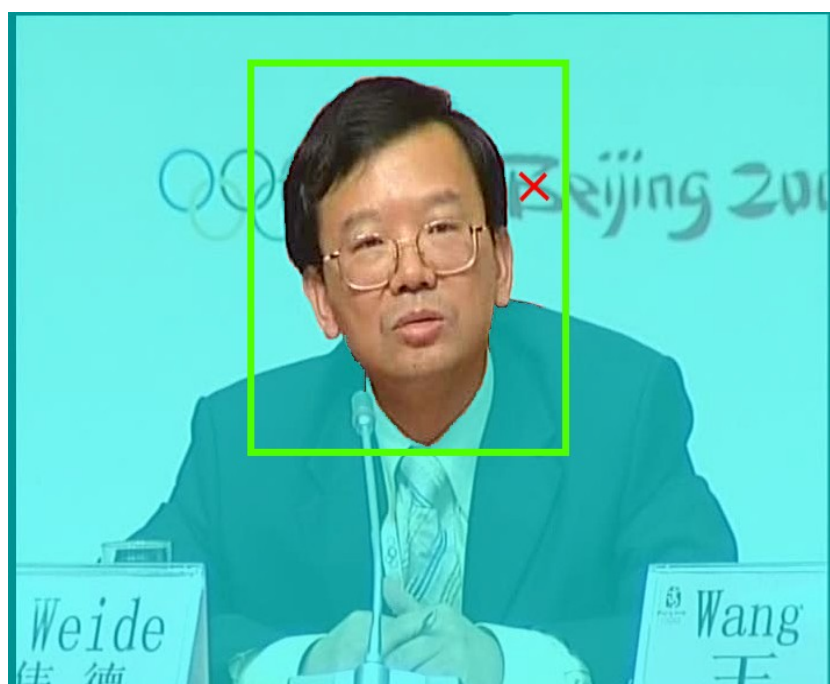


Figura 54: Exemple d'arbre molt dispers

Per l'altra banda, el criteri d'expansió del mode traça permet que l'usuari dibuixi traces sense gran esforç i obtingui resultats bastant coherents sempre que es segueixi la premissa mencionada: regions bastant homogènies i ben diferenciades del fons. El mode rectangle, pel contrari, dona bons resultats amb regions menys homogènies ja que les fulles que no formen part de la regió objecte no solen colar-se a dins de la regió objecte.

A la Figura 55 s'observa un exemple d'una regió molt dispersa en l'arbre de particions a la Figura 54, la qual dona resultats molt bons amb el mode rectangle. Com s'observa, només cal fer una segona interacció deseleccionant una fulla.



*Figura 55: Exemple de regió molt dispersa*

De la mateixa manera, quan es tracta d'objectes formats per moltes regions petites la millor opció és el mode rectangle, ja que amb la traça l'usuari hauria d'assegurar-se de pintar totes les regions necessàries i no disposa de la partició inicial així que hauria de pintar molt a sobre de l'objecte per garantir-ho. Pel contrari, amb el rectangle només cal que aquest polígon contingui totes les fulles que formen l'objecte.



*Figura 56: Exemple d'un objecte format per moltes fulles*

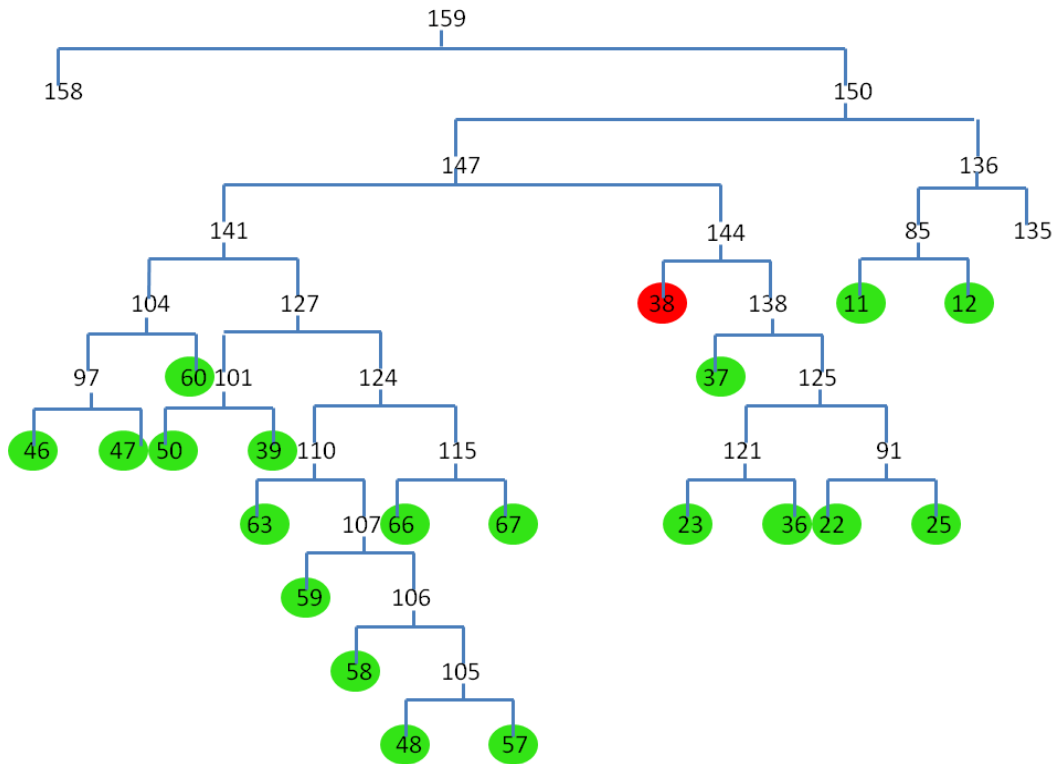


Figura 57: BPT exemple d'un objecte format per moltes fulles

A la Figura 56 trobem una ampolla amb molts contorns i colors marcats, això donarà lloc a un arbre molt particionat, com s'observa en la Figura 57. Si es volgués seleccionar totes aquestes fulles amb el mode traça, també s'hauria de pintar molt i, tot i així, probablement es deixaria alguna fulla sense seleccionar. En canvi, amb un únic rectangle es troben resultats gairebé impecables necessitant un únic clic per deseleccionar la fulla 38.

Per últim, quan es tracta de logotips, si aquests no són molt compactes, el mode traça és molt inadequat ja que sempre se seleccionaran regions que no formen part de l'objecte. Com s'observa en la Figura 58, en fer una traça l'etiqueta es propaga per moltes petites regions que no interessin. En canvi, a la Figura 59 amb un únic rectangle s'aconsegueix seleccionar de forma neta el logotip del detergent de la fotografia.



*Figura 58: Exemple d'un logotip amb el mode traça*



*Figura 59: Exemple d'un logotip amb el mode rectangle*

## **6.2 Millores al Digimatge**

En definitiva, aquest treball ha consistit en la incorporació del Digimatge, Digitrace i Detector de text a dins del Digion, per tal de complementar l'eina. Això s'ha aconseguit mitjançant la creació d'un **nou mòdul de visualització de keyframes** al Digion que compartís les mides i estètica de l'anterior però incorporant un seguit de novetats prenent que l'usuari final de l'eina es trobi amb una interfície amb la qual pugui

interactuar de forma còmode, amigable i intuïtiva ampliant la llibertat a l'hora de fer consultes, que abans es limitaven a consultes textuais.



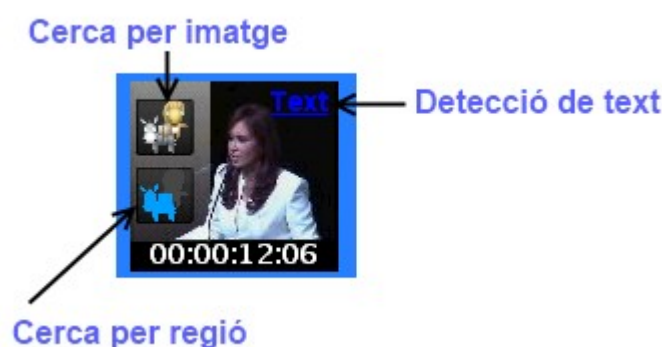
Figura 60: Nou mòdul de visualització de keyframes del Digition

Per aconseguir-ho, s'han implementat diferents aplicacions. Aquestes són:

- **Historial de cerques:** El programa té una memòria de fins a 3 cerques, que l'usuari podrà tornar a visualitzar simplement clicant a sobre de cada *step*. Un *step* consistirà en un enllaç cap a la pantalla de cerca corresponent. A més, situant el ratolí a sobre de cada *step* es pot llegir el títol de l'asset al qual pertany el *keyframe* utilitzat per fer la consulta i l'instant de temps en que s'ha capturat així com el tipus de cerca realitzada.
- **Menú de visualització:** Aquest menú neix mogut per les necessitats expressades pels usuaris de TVC per tal d'aclarir la procedència de cada *keyframe* resultant d'una cerca.

Per això, el menú permet escollir com es mostren els *keyframes* resultants així com donar més informació d'aquests. Es pretén que el mòdul de visualització dels resultats sigui capaç d'oferir la màxima informació possible sense saturar a l'usuari.

- En el cas de mostrar els resultats per **score**, l'usuari veurà els resultats ordenats segons la semblança a la imatge original, que es podrà conèixer mitjançant un percentatge amb un rang del 0% al 100% a sota de cada imatge. A més, quan el ratolí passa per sobre d'una miniatura, aquesta i totes les miniatures provinents del mateix asset es marcaran mitjançant un color de fons determinat, diferent per a cada asset mostrat.
  - Si els resultats es mostren per **asset**, l'usuari veurà els resultats ordenats per assets i, a dins de cada asset, per codi de temps. En aquest cas, es visualitzarà el codi de temps a sota de cada miniatura. Si s'escull aquest mode de visualització, l'usuari sempre veurà els colors de fons per a cada miniatura i, així, podrà distingir quan acaba un asset i comença l'altre.
- **Accés als mòduls d'anàlisi de la imatge:** Al llarg d'aquests mesos, la manera de facilitar a l'usuari la consulta per imatge o regió i la detecció de text ha estat un punt de discussió, ja que es volia integrar perquè sigués intuïtiu i, alhora, eficaç. S'ha decidit que la forma més còmode és poder fer-ho sobre la pròpia imatge en miniatura. Per això, l'usuari visualitza les opcions de cerca mitjançant un menú de botons a sobre de cada miniatura quan hi passa per sobre amb el ratolí.



*Figura 61: Detall d'una miniatura*

D'aquesta manera, l'usuari pot fer una cerca simplement situant-se a sobre de la imatge en miniatura i prement un botó i, alhora, s'evita la mala visibilitat de la imatge ja que els botons només apareixen si l'usuari passa per sobre de la imatge amb el ratolí. El tipus de resposta del programa depèn de l'eina d'anàlisi

seleccionada:

- En el cas de **cerca per imatge** es reben els resultats de la cerca, que es visualitzen al mateix mòdul de visualització de *keyframes* obrint un nou *step*. El mode de visualització per defecte es per *score* excepte a l'inici.

L'enllaç de *Inici* als passos de cerca del nou mòdul de *keyframes*, a la Figura 60, mostra el resultat de la primera cerca, que serà una cerca textual. Per tant, aquest enllaç mostra tots els *keyframes* d'un asset concret i, per aquest motiu, no té sentit habilitar la opció de visualització per *score*.

- Si l'usuari escull la **cerca per regió** s'obre una finestra emergent amb la interfície del Digitrace per tal de seleccionar la regió amb ajuda dels marcadors. Un cop s'envia la regió definitiva, es tanca el Digitrace i s'esperen els resultats, que seran mostrats d'igual manera que els anteriors. Per tant, s'afegeix un nou *step* als passos de cerca.
- En el cas de la **detecció de text** s'obre una finestra emergent amb la interfície adaptada del *showText*, mantenint la mateixa estètica i les mateixes funcionalitats de l'antic detector de text.

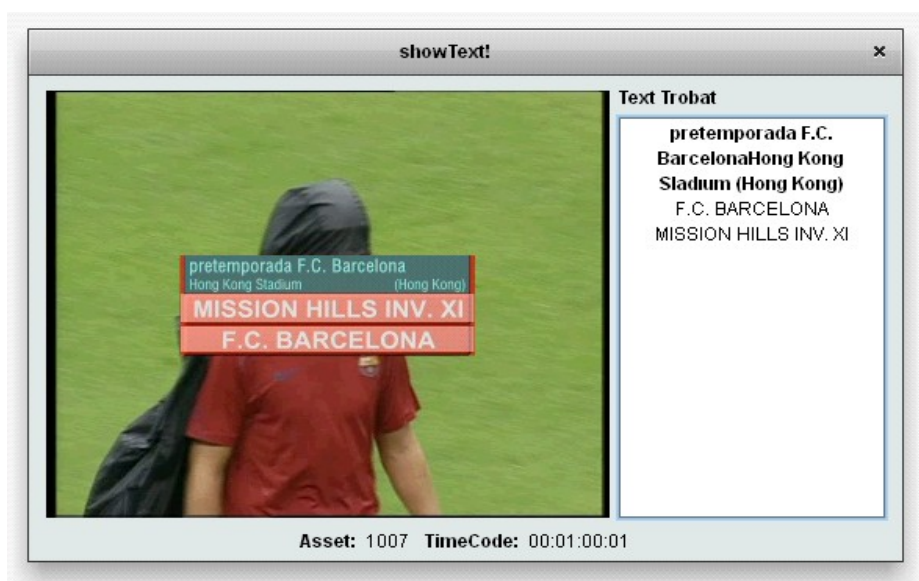


Figura 62: Pop-up del *showText* adaptat a la interfície

- **Configuració:** Per últim, a dins del nou Digimatge també hi trobem el menú de configuració, a on es poden canviar els pesos dels descriptors visuals per la consulta per imatge sencera. La idea principal és que aquest mòdul només es faci visible quan l'usuari ho decideixi per evitar saturar la pantalla amb informació innecessària.

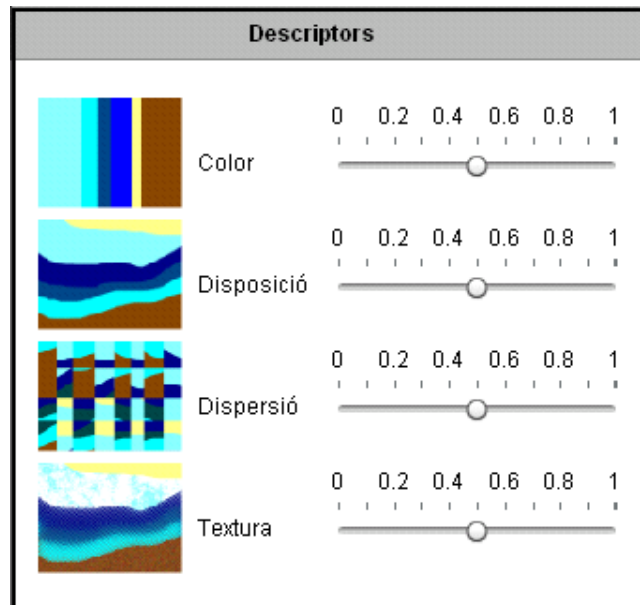


Figura 63: Menú de descriptors visuals d'imatge global

### 6.2.1 Digitrace

En aquest apartat es presenta la interfície de selecció de regions desenvolupada en aquest projecte, el **Digitrace**. El Digitrace es compon d'un mòdul principal integrat per una imatge central i els botons necessaris pels diferents modes i opcions distribuïts al voltant d'aquesta imatge. La condició de la que es parteix per dissenyar aquesta interfície és que ha de constituir un mòdul més a dins d'una gran interfície. Per aquest motiu, es decideix que sigui un mòdul petit i la imatge original, de 720X576 píxels, s'escala a la meitat de la mida real.





Figura 64: Digitrace

S'han definit dos modes de segmentació interactiva que conformen dues tècniques diferents que podrà fer servir l'usuari per seleccionar un objecte en la imatge. Per tant, el Digitrace compta amb dos modes configurables des d'el **menú de mode**.

- Amb el **mode traça** l'usuari només ha de dibuixar traces per marcar l' objecte o el fons, l'objecte serà l'opció per defecte. Cada vegada que es dibuixa una traça es retorna una màscara que es visualitza a sobre de la imatge per mostrar l'objecte resultant diferenciat del fons.  
L'usuari pot seleccionar quin tipus de traça vol dibuixar amb el **menú d'objecte o fons**, el qual disposa de dos botons per escollir marcadors d'objecte o marcadors de fons. Aquests marcadors es diferencien pel seu color.
- El **mode rectangle** segueix unes passes més definides: per començar, es dibuixa

un rectangle que es pot repetir tantes vegades com calgui ja que, cada vegada que es dibuixa un nou rectangle, s'esborra l'anterior. Per a cada rectangle dibuixat, es retorna una màscara sobre la imatge que mostra la regió resultant de la cerca. El marcador per defecte en el menú d'objecte o fons és el marcador d'objecte i, fins que no s'ha dibuixat un rectangle, no s'habilita el botó per escollir marcadors de fons.

Un cop s'ha dibuixat el rectangle, es passa a seleccionar o deseleccionar petites regions mitjançant clics a dins de l'àrea predefinida pel rectangle. Aquest pas es pot fer de dues maneres:

- Clicant amb el botó dret del ratolí a dins de l'àrea del rectangle.
- Seleccionant el botó per marcadors de fons, de manera que es podran fer els clics amb el botó esquerre. Si s'escull aquesta opció, per dibuixar un nou rectangle s'haurà de prémer el botó per a marcadors d'objecte.

El Digitrace també compta amb un **menú d'edició d'imatge**. Aquest menú té tres botons:

1. **Enrere:** opció per tornar enrere, borra o redibuixa un marcador segons la última acció guardada a la memòria.
2. **Esborra:**
  - Quan la goma s'apropa a un marcador, el gruix del marcador creix i el color canvia indicant que ens trobem a dins del radi d'incidència de d'aquesta. L'apropament haurà de ser major o menor segons la tolerància d'esborrat seleccionada al mòdul de configuracions.
  - Quan la goma s'allunya del marcador, aquest recupera el seu gruix i color originals.
  - En fer un clic, l'usuari esborrarà el marcador o marcadors que estan

remarcats.

En el mode rectangle, quan el marcador esborrat és el rectangle, s'esborren automàticament tots els marcadors deixant la memòria a zero i la pàgina sense màscara, ja que aquest mode exigeix un primer rectangle per trobar una regió.

En el mode traça, l'usuari pot fer ús del botó dret del ratolí per esborrar un marcador situant-se a sobre d'aquest en qualsevol moment.

- 3. Comença:** Deixa la pàgina del mode en ús completament en blanc i sense màscara però manté la pàgina del mode que no s'està utilitzant tal i com estava abans.

El **menú de configuracions** fa visible un segon **mòdul secundari**. L'usuari pot decidir en tot moment si vol que el mòdul secundari sigui visible o bé amagar-lo. Aquest menú també compta amb tres botons. Cadascun d'ells fa visible una pàgina diferent del mòdul secundari:

#### 1. Mòdul de colors:

Com indica el seu nom, aquest mòdul serveix per configurar colors. Aquests colors poden ser els dels marcadors o bé els de la màscara, per tant, en aquesta pàgina hi trobem dues pestanyes:

- **Marcadors:** Serveix per canviar el color dels marcadors per objecte i fons per separat segons l'usuari ho disposi. D'aquesta manera es garanteix una correcta visió del marcador independentment del color de la imatge. També es facilita tornar a l'opció per defecte: color verd pel marcador d'objecte i vermell pel fons.
- **Màscara:** Serveix per canviar el color i la transparència de la màscara que se superposa a sobre de la imatge per mostrar la regió seleccionada en contraposició amb el fons. Aquest color s'aplicarà a sobre del fons i l'objecte quedarà visible. Per

definir la transparència de la màscara es disposa d'una barra lliscant que permetrà un contrast major o menor de l'objecte respecte el fons.

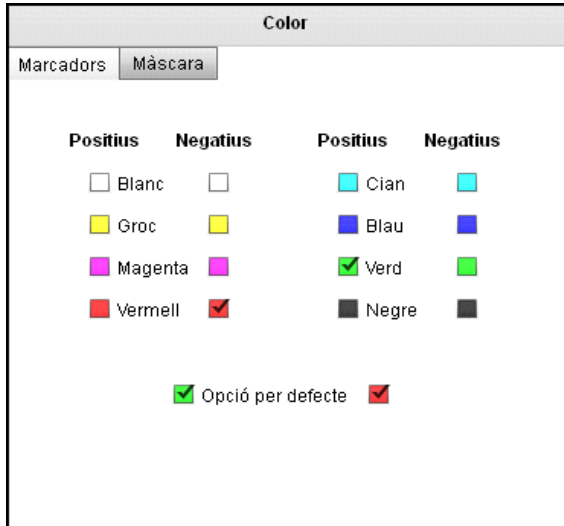


Figura 65: Configuració del color dels marcadors

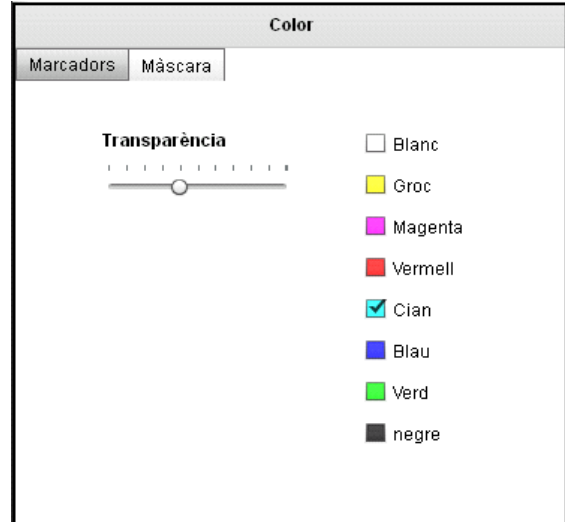


Figura 66: Configuració del color de la màscara

## 2. Mòdul de configuració:

Aquest mòdul compta amb dues pestanyes per configurar els descriptors visuals i el pinzell: Aquesta pàgina té dues pestanyes.

- **Descriptors:** La major part dels descriptors visuals per regions implementats pel servei de consulta per exemple basada en regió de la UPC es basen en l'estàndard MPEG-7. El seu valor per defecte és 0.5 i l'usuari pot variar la rellevància de cadascun individualment en la cerca amb un rang de 0 a 1. Alguns d'aquests descriptors són compartits amb la cerca per imatge sencera i són: Dominat Color, Texture Edge Histogram, Shape CSS, posició i mida.
- **Pinzell:** Permet configurar el gruix de la línia tant de la traça al mode traça com del rectangle al mode rectangle. També dona la possibilitat de configurar el radi del punt que es dibuixa al mode rectangle i, per últim, permet variar la tolerància d'esborrat la qual actua sobre el radi d'incidència de la goma.

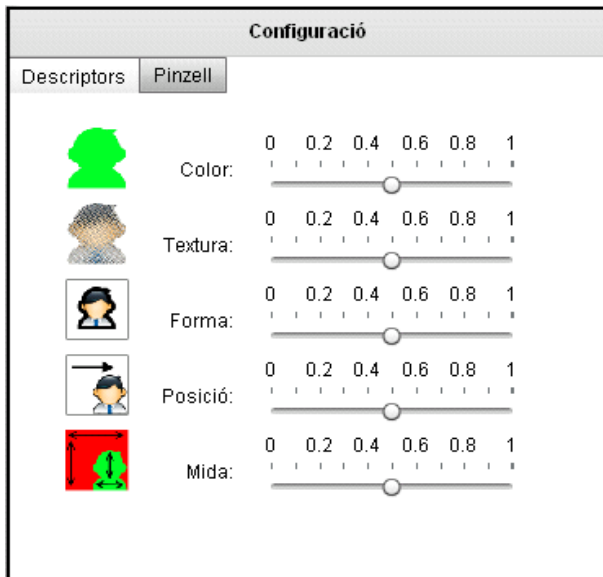


Figura 67: Configuració del descriptors de regió

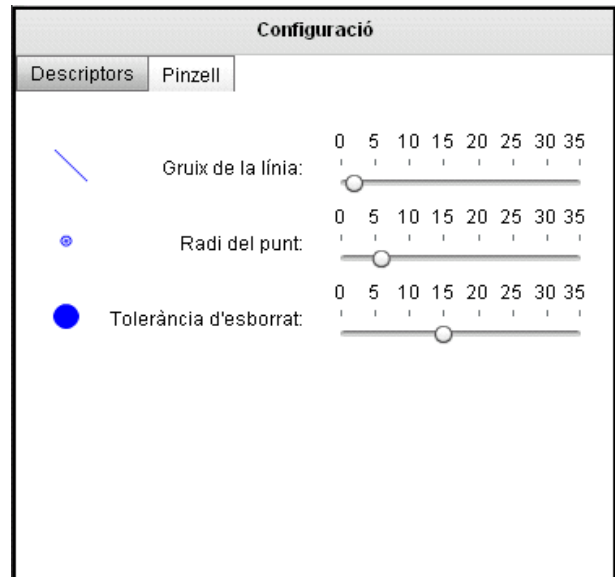


Figura 68: Configuració del pinzell

### 3. Mòdul d'ajuda:

El mòdul d'ajuda consta de tres pàgines que es mostraran segons el mode que s'estigui utilitzant en cada moment.

Quan s'obre el Digitrace no hi ha cap mode prefixat i la pàgina d'ajuda guia a l'usuari en l'elecció d'aquest mode. Un cop escollit el mode, a la pàgina d'ajuda s'especifiquen els passos que s'han de seguir per la selecció de la regió d'interès.

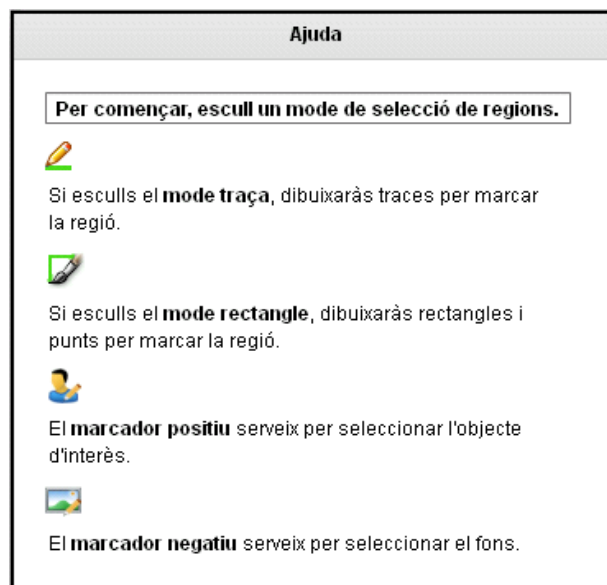


Figura 69: Pàgina d'inici del mòdul d'ajuda

Per últim, el **botó de cerca** envia tots els marcadors més els descriptors visuals cap al servidor tancant el Digitrace. A continuació es reben les imatges resultants de la consulta per mostrar-les a l'usuari a dins el mòdul de *keyframes* del Digitation.

### **6.3 Integració en el sistema**

La integració de tot el sistema des de la segmentació interactiva fins a la cerca per regions un cop seleccionada la regió de suport, no s'ha pogut dur a terme completament a causa, bàsicament, de dos problemes:

- El motor de cerca per regions de la universitat no respon correctament, és a dir, ofereix resultats inestables. Per aquest motiu, no ha sigut possible experimentar amb el servei de cerca per regions ni amb els pesos dels descriptors visuals i treure'n conclusions en aquesta memòria. A més, aquest motor de cerca no està ben integrat en el repositori del Fedora Commons de la UPC. Per tant, fins que no se solucioni aquest problema des de la universitat, el Digitrace no serà capaç de respondre a les cerques.

A la Figura 70 es mostra una captura del cercador de la UPC, el GOS, responent a una cerca per regió.

- El mètode Post encara no s'ha assolit en les comunicacions entre la UPC i la CCMA. Per aquest motiu, el Digitrace no és capaç d'oferir, ara per ara, la màscara resultant d'una segmentació interactiva amb el mode traça.

A causa d'aquests dos impediments, el Digitrace actualment resta a l'espera de la seva resolució per poder respondre a les cerques per regió. Pel contrari, les cerques per imatge i la detecció de text funcionen sense problemes així com la segmentació interactiva pel mode rectangle.

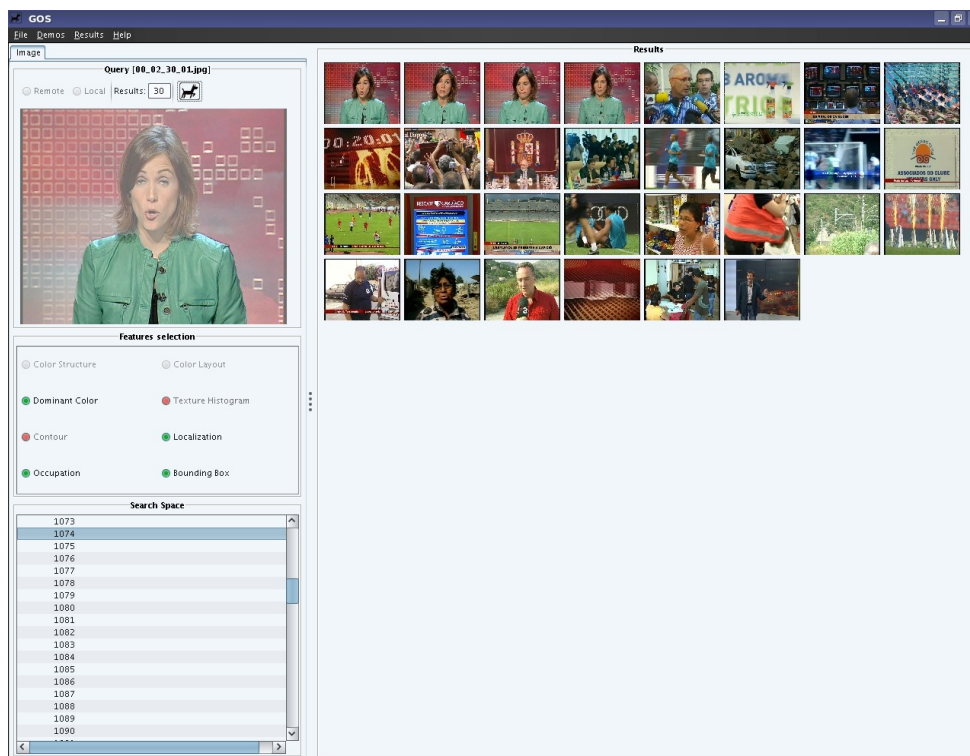


Figura 70: Captura del GOS responen a una cerca per regió

La segmentació interactiva pel mode traça sí ha estat integrada amb èxit. Es mostra un exemple a la Figura 75.

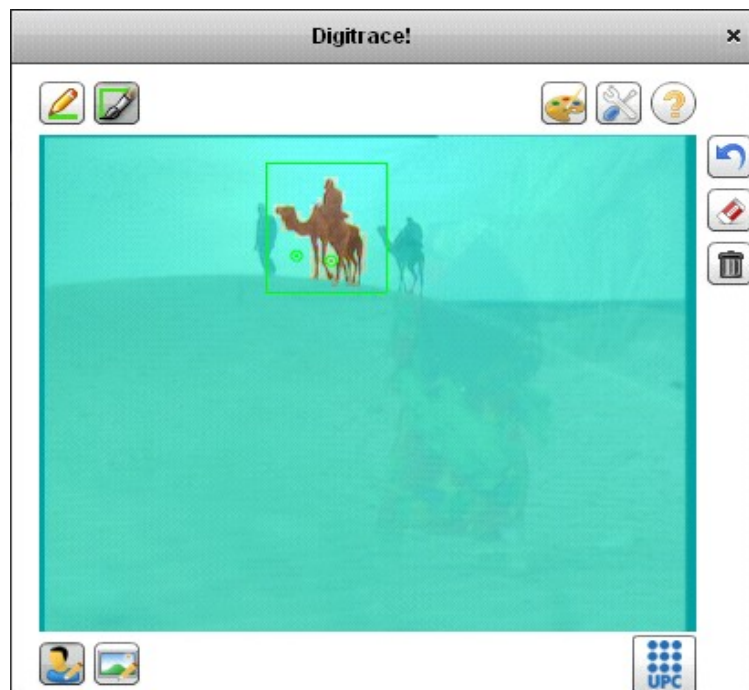
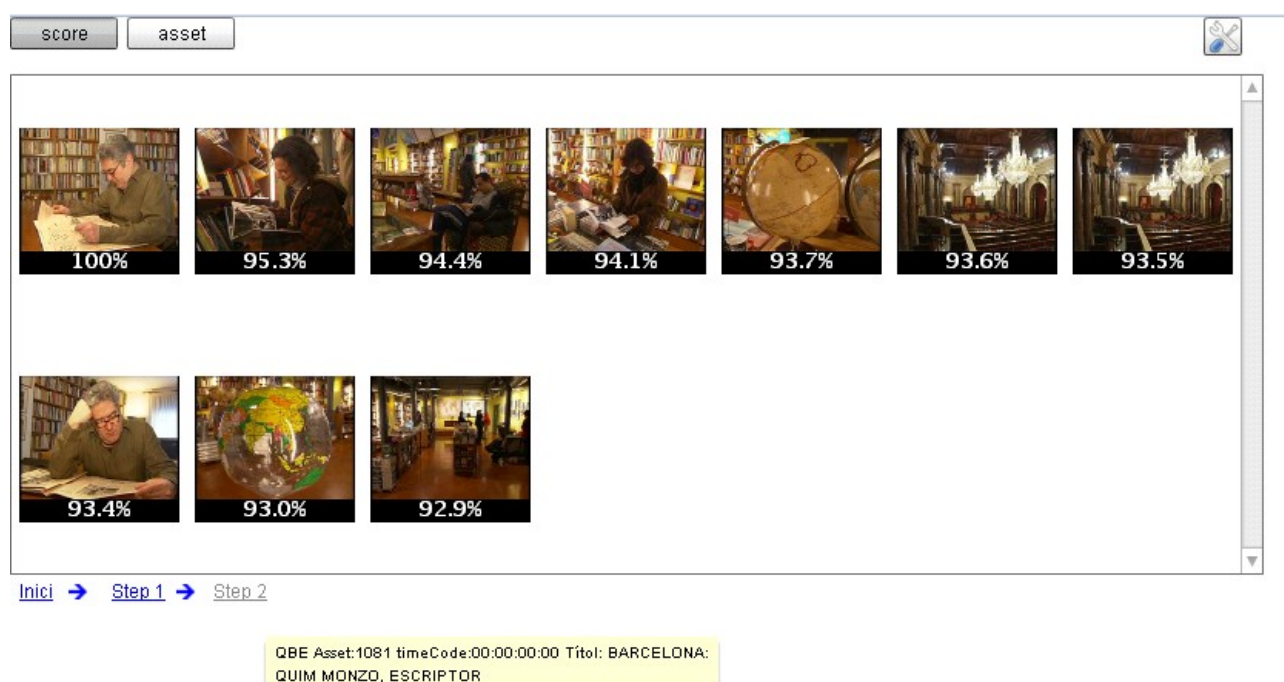


Figura 71: Segmentació interactiva al mode traça

La cerca per imatge global sí ha estat integrada completament amb èxit i s'han pogut fer alguns experiments amb els pesos dels descriptors des de la pròpia interfície. Els primers resultats de les cerques són bastant coherents. Els resultats més allunyats són més inestables degut, principalment, a que la cerca s'està fent sobre un nombre petit d'assets.

A la Figura 72 es mostra un exemple de consulta, la imatge de partida és la primera que es mostra. S'ha donat rellevància als pesos dels descriptors de textura i de color. Per aquest motiu, trobem resultats amb els mateixos colors dominants i proporcions semblants i, a més, els contorns de les imatges resultants coincideixen amb els de la imatge de partida.



*Figura 72: Resultats d'una cerca per imatge*

A la Figura 73 es mostra un altre exemple. Aquest cop es dona principal rellevància als descriptors de color i disposició. Com s'observa, els primers resultats de la cerca comparteixen les característiques de color dominant i disposició de colors amb la imatge base.



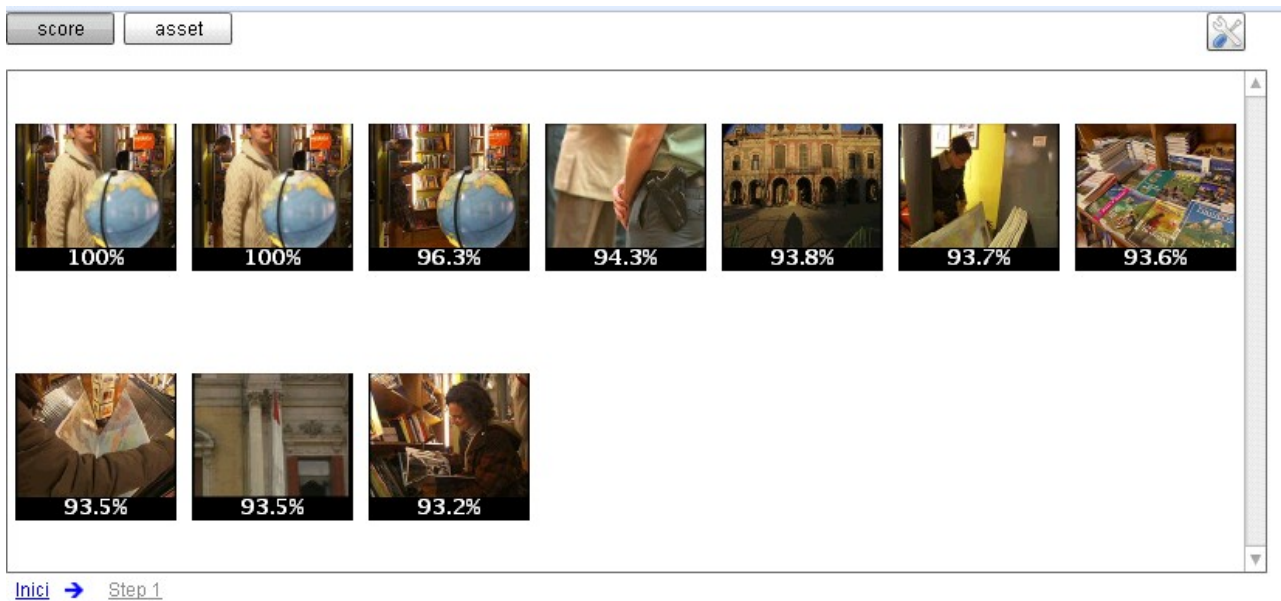


Figura 73: Resultats d'una cerca per imatge

Per últim, a la Figura 74 es mostren els resultats d'una cerca donant importància al color i la dispersió. Per aquest motiu, s'observen resultats amb tons semblants i, a més, tots els resultats mostren una dispersió semblant de cada color a dins de la imatge.

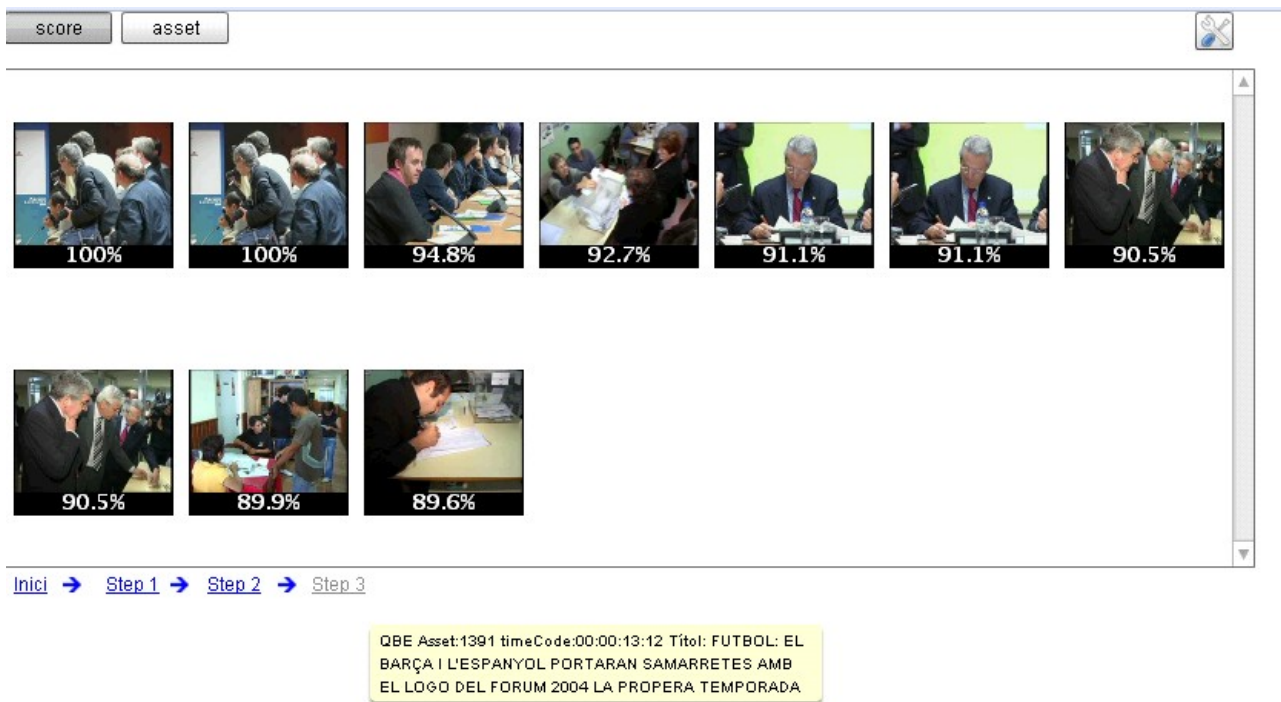


Figura 74: Resultats d'una cerca per imatge

## 7 Conclusions

### 7.1 Grau d'assoliment dels requeriments

Per començar, el primer requeriment plantejat abans de començar va ser la implementació d'una eina en un llenguatge específic i amb una programació modular, permetent la seva integració amb altres sistemes existents, així com la comunicació remota amb altres parts del sistema.

També es demanava el disseny d'una interfície de cerca basada en regions: el Digitrace s'encarrega de cobrir aquesta necessitat permetent l'ús de dos modes diferents per la selecció de regions: el mode rectangle i el mode traça.

A més, la interfície global multimodal és capaç d'accedir a la informació continguda al repositori de la CCMA així com de comunicar-se amb el mòdul de la UPC per dur a terme les cerques per imatge o per regió o bé per demanar resultats sobre una imatge al detector de text. Aquesta interfície ha sigut possible gràcies a la incorporació d'un nou mòdul de *keyframes* al Digitation, per tant, tal com es demanava, la interfície ha estat integrada a l'entorn del Digitation.

També s'han incorporat uns botons de configuració que garanteixen la llibertat de l'usuari a l'hora d'escollir els pesos que han de tenir els descriptors per fer la cerca tant si es tracta d'imatge global com de regions.

Per últim, podem afirmar que les característiques intrínseques que ha de garantir una RIA queden cobertes amb aquesta interfície ja que s'ha tingut molta cura amb el disseny i la claredat de la interfície en tot moment. Concretament al Digitrace, tots dos modes oferts per la segmentació interactiva han estat pensats perquè l'usuari pugui

seleccionar una regió amb la mínima interacció possible i la màxima eficàcia.

La interfície també ofereix la màxima informació de forma senzilla i intuïtiva prometent una interacció clara i sense grans complicacions. Aquesta última afirmació es pot justificar amb diferents elements de la interfície com els menús, opcions de configuració i instruccions.

## **7.2 Punts forts de la interfície**

Com a punt fort de la interfície, cal remarcar que comparteix les qualitats de les interfícies d'escriptori i les interfícies web que s'han trobat al mercat actual per a la segmentació interactiva. Per tant, la nostra interfície integra l'accessibilitat d'un entorn web amb l'aplicació d'una tècnica de segmentació automàtica, la qual ofereix una segmentació interactiva molt més ràpida i eficaç que les segmentacions manuals.

Un altre punt a esmentar és la originalitat de la proposta de segmentació interactiva, ja que el nostre treball s'ha centrat en desenvolupar un criteri suficientment restrictiu com per poder assegurar uns resultats força coherents sense necessitat d'utilitzar marcadors de fons. Tot això garantint un esforç petit en la interacció de l'usuari en la selecció de la regió.

## **7.3 Conclusions tècniques**

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

Per una banda, centrant-nos en la **segmentació interactiva**, es pot afirmar que la tècnica per arbres de particions és molt més complexa i més acurada que les altres tècniques exposades a l'estat de l'art i avaluades als resultats. Específicament, s'ha fet un **estudi comparatiu** dels dos modes implementats sobre la nostra proposta de

segmentació interactiva. Amb aquest estudi s'ha pogut avaluar la usabilitat de cadascun dels modes i s'ha comprovat que són tècniques complementaries.

El mode traça resulta força adequat per a imatges homogènies que donen lloc a arbres de particions menys dispersos. El punt a destacar d'aquest mode és l'esforç mínim que garanteix a l'usuari si es compleix amb la premissa exposada. Això és conseqüència del criteri de propagació definit pel mode traça, el qual permet dibuixar un petit marcador sobre l'objecte i que aquest s'expandeixi ràpidament.

El mode rectangle, en canvi, és molt útil per seleccionar objectes amb molts colors i contorns, que donen lloc un arbre amb molts segments petits. Amb el mode rectangle, se seleccionen sense problema aquests segments garantint que cap d'ells quedarà exclòs de la regió final sempre que estigui contingut a dins del rectangle. També és el mode més adequat per seleccionar objectes que contenen alguna fulla que embruta l'arbre als nodes inferiors. Per últim, el mode rectangle és de gran ajuda per a arbres molt dispersos, és a dir, arbres a on trobem les particions que conformen la regió final disperses per diferents branques i ajuntant-se amb altres fulles de la partició que formen part del fons.

Per l'altre banda banda, **a nivell d'interfície**, la integració de totes les opcions a dins de la pròpia eina de la Corporació, el Digiton, facilita l'acceptació per part dels usuaris, que no hauran de fer l'esforç de conèixer una eina des de zero. D'aquesta manera, amb una eina molt propera podran provar les capacitats que els hi ofereix la cerca per exemple i entendre les possibilitats que ofereix.

Si més no, l'eina compta amb algunes mancances que dificulten, ara per ara, la seva explotació. La interfície està preparada per fer les crides per a la cerca per exemple i per a rebre els resultats d'aquesta cerca però l'UPSeek encara no és capaç d'oferir els resultats per a la cerca per regió. Aquest pas és completament necessari abans de pensar en l'explotació de l'eina.

Altres debilitats menys evidents, són els resultats inestables de les cerques per exemple i la lentitud de resposta del motor de cerca, fruit de la seva manca d'algorisme d'indexació eficient. La CCMA, que està molt interessada en posar en marxa aquest servei, manifesta la seva preocupació per la falta de rapidesa en les cerques. Aquest és un punt d'inflexió per poder portar a un context real l'eina desenvolupada ja que, si els resultats triguen molt de temps en mostrar-se, l'usuari no veurà factible la incorporació d'aquestes noves opcions de cerca en la seva rutina.

Pel que fa a la utilitat, i ja entrant en una opinió més personal, la cerca per exemple és una forma diferent d'expressar una idea que, a vegades, pot ser difícil de trobar en paraules. Amb una cerca per exemple, per tant, es poden fer cerques de conceptes difícils d'expressar amb paraules. A més, augmenta la possibilitat de trobar detalls d'una imatge que no han estat introduïts a les descripcions textuais que s'emmagatzemen com a metadades d'un asset.

Tota aquesta reflexió em porta a afirmar que, personalment, opino que, un cop resoltes les debilitats de l'eina, el futur de la cerca per exemple en les empreses del sector audiovisual és bastant clara, ja que cobreix una necessitat que actualment no està resolta.

#### ***7.4 Conclusions personals***

Personalment, aquest projecte m'ha aportat una experiència de treball en equip i una visió del món laboral en el sector audiovisual, més concretament en la recerca en aquest àmbit. També ha suposat la oportunitat de treballar amb una gran empresa com és la CCMA i conèixer les relacions internes i col·laboracions entre aquesta i la universitat.

El fet de formar part d'un projecte d'aquestes dimensions m'ha obert un món desconegut en l'àmbit de la recerca del processat de la imatge, no només en els sistemes CBIR sinó en altres àmbits.

Centrant l'atenció en el projecte, el constant diàleg tant amb l'empresa com amb la universitat m'ha situat com a medidora entre totes dues i això m'ha aportat una visió general de tot el procés necessari per crear una eina com aquesta, sent particip, de forma més o menys directe, de totes les passes d'aquest procés: des de l'arbre binari de particions fins a les proves finals de test amb la pròpia interfície resultant passant per la programació de cada botó i cada opció del Digitrace així com el disseny dels dos modes de segmentació interactiva que s'han explicat.

Per tant, la interfície resultant ha sigut fruit d'un esforç conjunt i d'un diàleg constant donant lloc a un sistema complex de comunicació entre la universitat i l'empresa per tal de relacionar el procés d'anàlisi i processament de la imatge, situat a la UPC, amb la interfície gràfica, situada a la CCMA.

A més a més, he pogut crear una eina amb una utilitat més enllà del projecte de final de carrera, ja que segurament tindrà una continuïtat i podrà ser aprofitada, almenys en part, per ser presentada a un usuari final més endavant. Aquesta afirmació es demostra amb la prova d'usuari que es pretén fer al juny per veure la resposta dels futurs usuaris de l'eina i poder conèixer de primera mà les virtuts i mancances que hi troben.

## **7.5 Treball futur**

Hi ha una sèrie de propostes que no s'han pogut assolir per manca de temps però que són prioritàries per poder obtenir una bona resposta dels usuaris perquè faciliten les cerques aportant més informació dels *keyframes* que es mostren en pantalla en cada moment:

- S'hauria d'incorporar la màscara utilitzada per la cerca basada en regions a dins de la pàgina que mostra els resultats per tal que l'usuari pugui comparar la regió de cerca amb els resultats de la consulta. S'han discutit diferents opcions però, per tal

de poder aprofitar l'estructura proporcionada pel Digition, s'ha decidit utilitzar el mateix mòdul de *keyframes* per mostrar, com a primer resultat de la cerca, el propi *keyframe* exemple amb la màscara a sobre. També seria força interessant que es mostrés la màscara, no només a sobre de la imatge de consulta, sinó a totes les imatges resultants de la cerca. D'aquesta manera es podria veure quina regió ha establert l'algoritme com a resultat de la cerca a cada miniatura del mòdul de *keyframes*.

- També és necessari afegir informació de forma més dinàmica al mòdul de metadats situat a la dreta del Digition. Aquest mòdul proporciona la informació textual associada a l'àsset resultant d'una cerca textual i donant per suposat que el mòdul de *keyframes* mostra tots els *keyframes* d'un sol àsset. Ara que s'han incorporat les cerques per imatge, els *keyframes* resultants d'una cerca poden provenir de qualsevol àsset i, per tant, s'hauria de crear un mòdul de metadada que s'actualitzés segons el *keyframe* seleccionat o bé segons la posició del ratolí sense necessitat de seleccionar.
- Caldria fer un agrupament dels resultats de la cerca per tal de mostrar el màxim contingut possible i que aquest tingui una rellevància.

Hi ha altres propostes que, simplement, són idees més o menys definides que podrien dur-se a terme en un futur. Algunes d'elles són les següents:

- Pensar una altra manera de que l'usuari interactuï amb els pesos dels descriptors de forma menys directa com, per exemple, assumint diferents configuracions segons el tipus de cerca (futbol, telenotícies, paisatges,...)
- L'empresa veu com a prioritària l'eficiència de la cerca que, ara per ara, és molt lenta per poder tenir un servei factible que l'usuari pugui utilitzar. Tot i així, aquest punt depèn de la universitat i encara no s'ha definit cap solució en concret.
- També seria una bona opció aprofitar el detector de text per fer cerques textuais a partir del text que es troba en una imatge.

## 8 Bibliografia

- [1] Ramon Salla, Aplicació rica d'internet per a la consulta amb text i imatge al repositori de vídeos de la Corporació Catalana de Mitjanç Audiovisuals. Projecte Final de carrera, UPC, Barcelona, Desembre 2009  
<http://upcommons.upc.edu/pfc/bitstream/2099.1/8766/1/PFC.pdf>
- [2] IMARS és un producte comercial de IBM evolucionat d'una proposta inicial del 1996: el MARS. Michael Ortega, Yong Rui, Kaushik Chakrabarti, Sharad Mehrotra, Thomas S. Huang. Supporting similarity queries in MARS. Seattle, Washington, United States, 1997.  
DOI= <http://doi.acm.org/10.1145/266180.266394>
- [3][ Lux, M. 2009. Caliph & Emir: MPEG-7 photo annotation and retrieval. In *Proceedings of the Seventeen ACM international Conference on Multimedia* (Beijing, China, October 19 - 24, 2009). MM '09. ACM, New York, NY, 925-926.  
DOI= <http://doi.acm.org/10.1145/1631272.1631456>
- [4] Xavier Giró-i-Nieto · Neus Camps · Ferran Marques (2010) , GAT: a Graphical Annotation Tool for semantic regions.
- [5] Kevin McGuinness · Noel E. O'Connor (2010) , Pattern Recognition.
- [6] <http://losingfight.com/blog/2007/08/28/how-to-implement-a-magic-wand-tool/>
- [7] Garrido L, Salembier P (2000) Binary partition tree as an efficient representation for image processing segnetation and information retrieval.
- [8] 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).
- [9] Tomasz Adamek (2006) Using contour information and segmentation for object registration, modeling and retrieval.
- [10] Carles Ventura, Image-based query by example using MPEG-7 visual descriptors Projecte Final de Carrera, ETSETB, UPC, 2010.
- [11] Manjunath, B. S.; Salembier, Philippe; Sikora, Thomas (abril de 2002).



*Introduction to MPEG-7: Multimedia Content Description Interface*. Wiley & Sons. [ISBN 0-471-48678-7](#).

[12] Delcor Ballesteros, Jordi i Pérez Noriega, Verónica. Descripción, indexación, búsqueda y adquisición de secuencias de vídeo mediante descriptores MPEG-7, Projecte Final de Carrera, EET, UPC, 2006

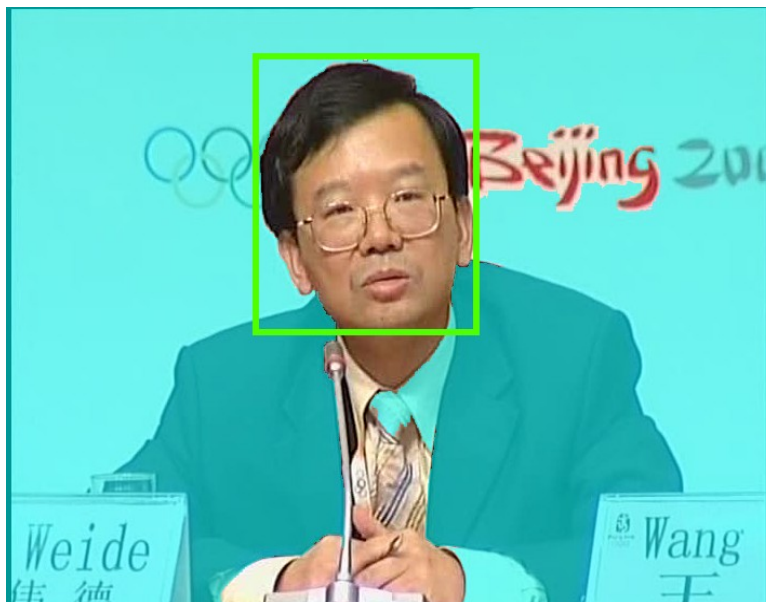
[13] Gloria Morales Subirà. Avaluació de sistemes de color per l'obtenció de colors dominants. Projecte Final de Carrera, EET, UPC, Gener 2010.

# Annex

## ***Annex I: Exemples de segmentació interactiva***

A continuació es poden trobar un seguit d'exemples d'extracció de cares, objectes i logotips mitjançant els dos modes de segmentació interactiva oferts en la nostra interfície: mode rectangle i mode traça.

L'objectiu de la primera part d'aquest annex és evidenciar el motiu que ha portat a escollir el criteri d'expansió pel mode traça i el criteri de no expansió pel mode rectangle, tal com s'indica a l'apartat de disseny.



*Figura 75: Exemple 1 del mode rectangle amb expansió*

A la Figura 55 s'observa com el mode rectangle selecciona la cara de la fotografia de forma molt neta. A la Figura 75, en canvi, s'observa la mateixa fotografia però seguint un criteri d'expansió. El rectangle dibuixat s'ajusta molt més a la cara que el de la Figura 55 però, tot i així, moltes regions que no pertanyen a la cara s'agreguen a la regió resultant.

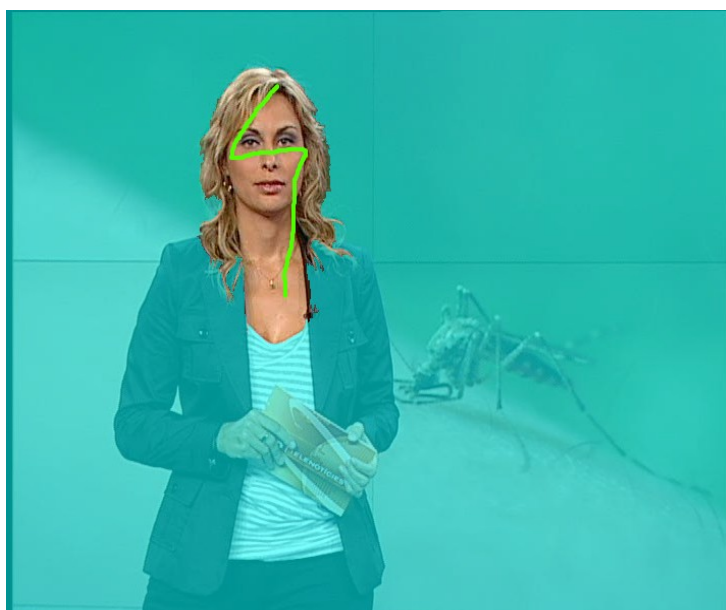
En el segon exemple de la Figura 76, s'observa com el rectangle intenta seleccionar el logotip del detergent però s'emporta moltes petites regions que haurien de

formar part del fons. En la Figura 59, pel contrari, es comprova que seguint el criteri de no expansió el resultat és molt més adequat.



*Figura 76: Exemple II del mode rectangle amb expansió*

Pel que fa al mode traça, a la Figura 77 s'observa una fotografia d'una presentadora, la regió a seleccionar en aquesta ocasió és la seva cara. Per tenir una idea de la partició de la cara, la Figura 78 mostra el tros de l'arbre de particions que conté la regió. Dibuint una traça com la que s'observa a sobre de la cara a la Figura 77, se seleccionen les fulles marcades en verd a la Figura 79. Seguint el criteri d'expansió, la propagació de l'arbre expandeix l'etiqueta cap a les fulles 14 i 16, marcades en blau.



*Figura 77: Exemple I del mode traça amb expansió*

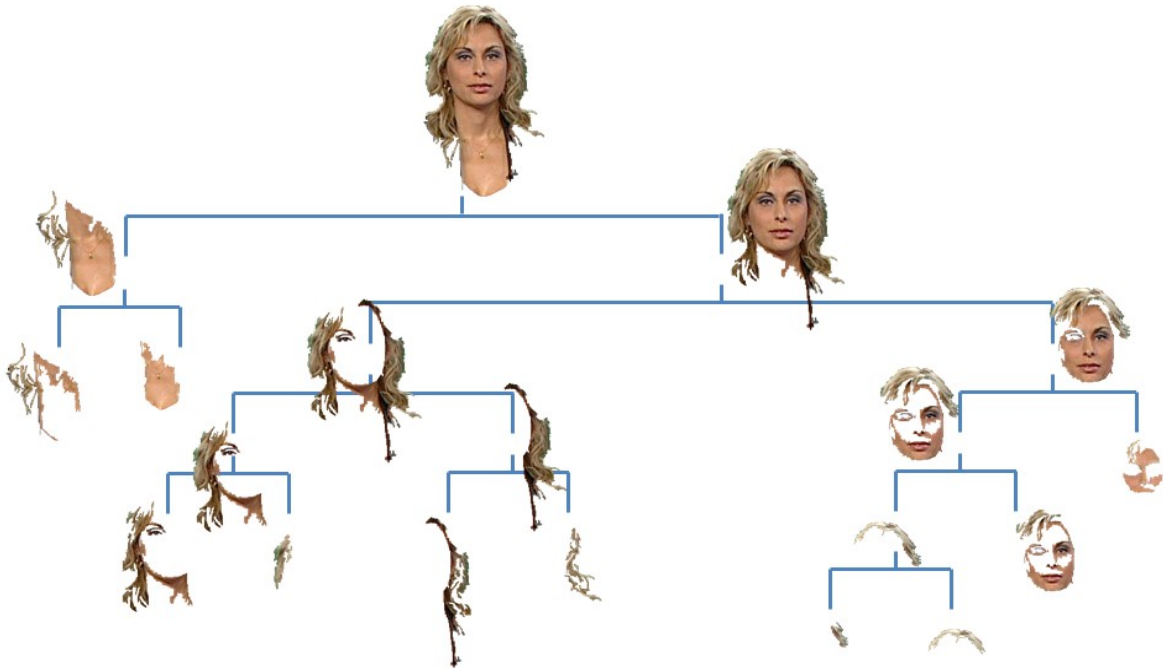


Figura 78: Partició d'un fragment de la imatge

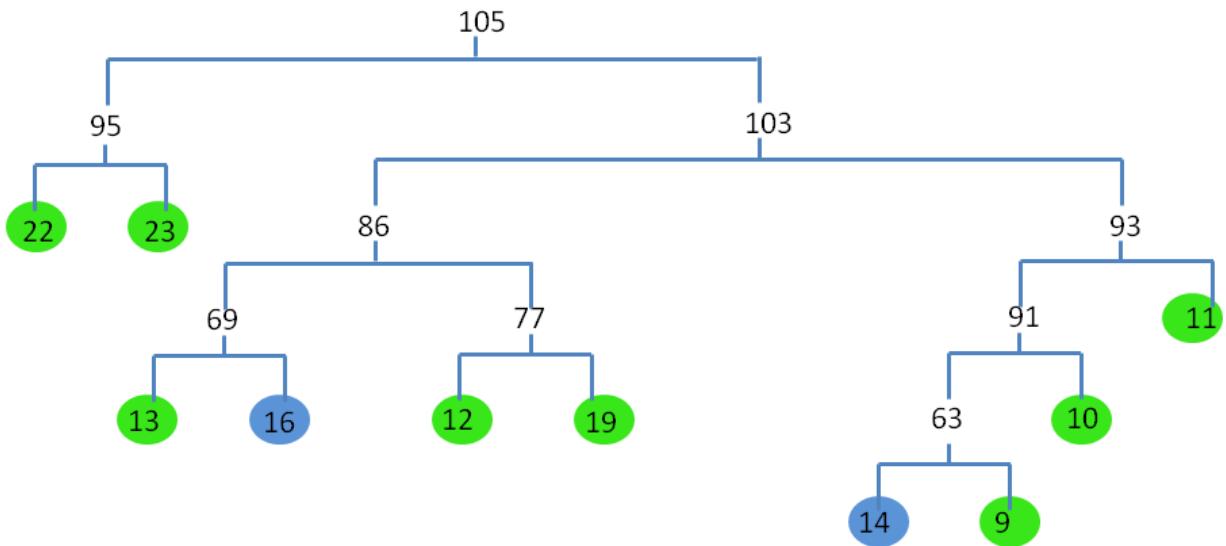


Figura 79: Fulles seleccionades i propagació a l'arbre

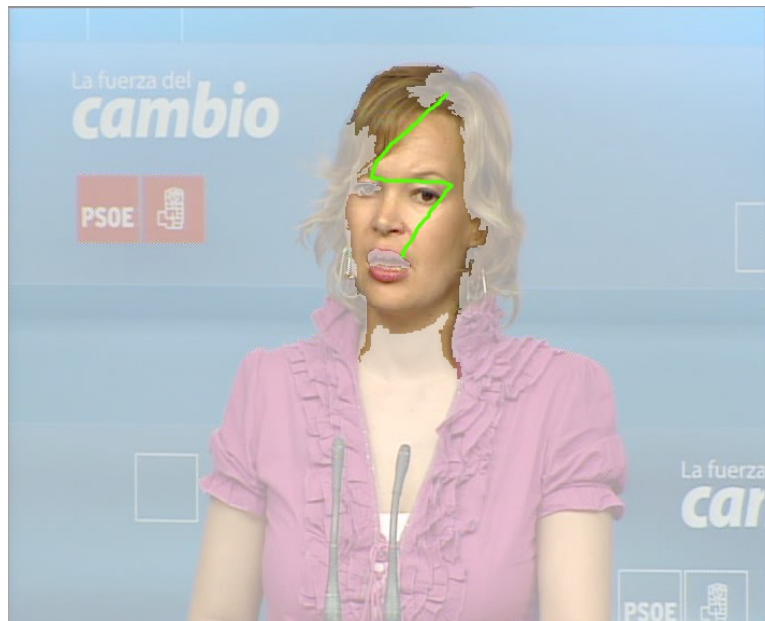
En canvi, aplicant el mode traça sense expansió, Figura 80, trobem petites regions que componen el cabell de la presentadora que no s'adhereixen a la regió (fulles 14 i 16). Aquest fet dificultarà la feina a l'usuari, el qual haurà d'anar seleccionant les petites regions una per una.



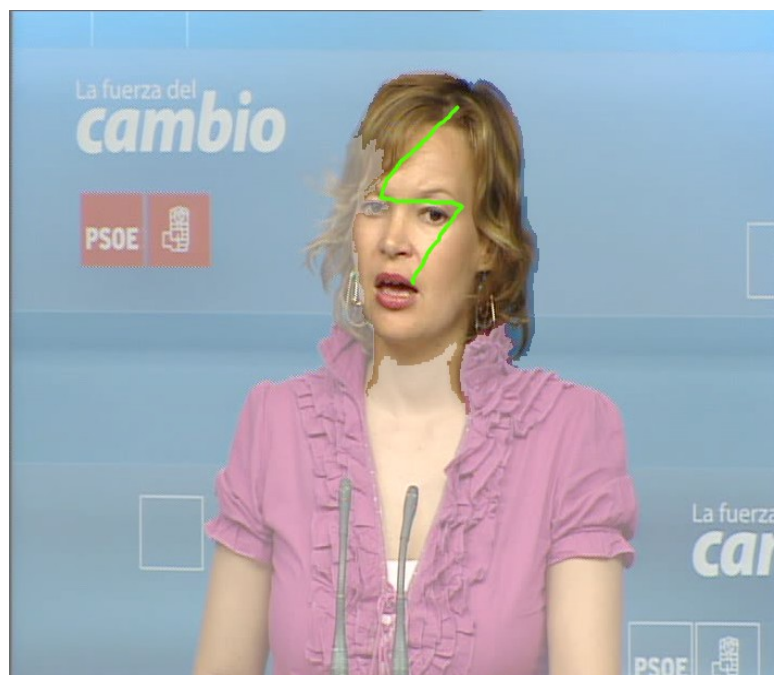
*Figura 80: Exemple 1 mode traça sense expansió*

Com s'observa en l'arbre de la Figura 78, la regió de la cara està formada per moltes regions petites que parteixen de diferents branques. Per aquest motiu, és molt fàcil que l'usuari, que no coneix la partició inicial, no passi per totes les fulles amb la traça: en aquest cas les fulles 14 i 16 queden sense etiquetar en un primer pas. Si se segueix el mode amb expansió resulta més probable que l'etiqueta s'expandeixi per aquestes fulles sense etiquetar que es troben entre les fulles marcades.

Un segon exemple confirma que el criteri amb expansió dona millors resultats pel mode traça: la Figura 81 deixa veure com s'intenta seleccionar la cara de la Leire Pajín però la traça no s'expandeix suficient. A la Figura 82, seguint un el criteri d'expansió i amb la mateixa traça, els resultats són més esperançadors i només caldrà seleccionar la regió del cabell que no ha estat afegida.

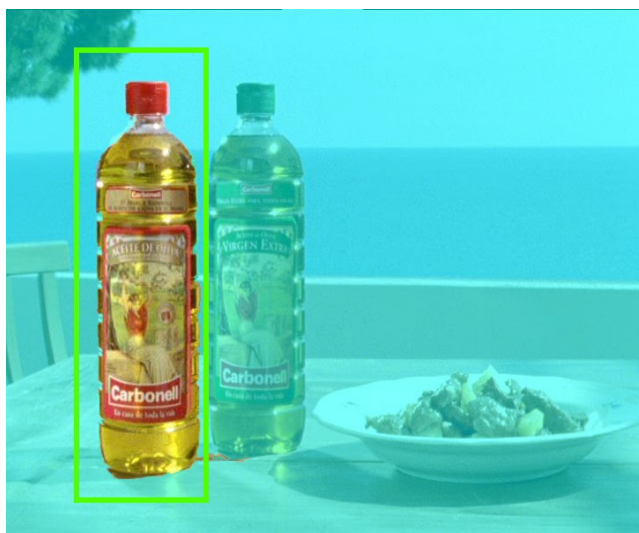


*Figura 81: Exemple II del mode traça sense expansió*



*Figura 82: Exemple II del mode traça amb expansió*

A continuació es mostren més exemples, un cop ja definits el mode rectangle sense expansió i el mode traça amb expansió, que deixen palesa l'efectivitat d'aquests dos mètodes:





## ***Annex II Artícles del Bitsearch***

Aquests darrers mesos s'han estat publicant els avenços de la nostra feina centrant l'atenció en els arbres binaris de particions, l'elecció dels modes de segmentació interactiva i també alguns aspectes de la comunicació entre el mòdul de la UPC i la CCMA. A continuació es mostren aquests articles, tot i que també es poden trobar en el bloc oficial: <http://bitsearch.blogspot.com/>

**Friday, May 7, 2010**

### **[Interactive segmentation running with rectangles](#)**

Just 19 days left to hand over my bachelor thesis and it seems like time runs faster. Things are beginning to work and we can already perceive the end. It's really rewarding to finally see the results of many hours of work and it's a pleasure for me to show them to you.

The point is that the communication between our interface for interactive segmentation and the server from the UPC for [rectangle mode](#) has succeeded. Now we can run the tool and work with markers getting a mask over the image to graphically visualize the selected region for each marker drawn. Then, for each point drawn, the client will send all the coordinates again and show the new mask.

It is said that a picture is worth a thousand words, so here it is our interface working with rectangle mode:



As we draw the rectangle, the client sends its coordinates to the server, who processes the image with its partition previously calculated and returns the mask, which is showed as a alpha over the image. This way the user can easily visualize the region he is marking and try to hone the result with points if necessary.

**Sunday, April 25, 2010**

### **Defining communication for interactive segmentation using scribbles**

As we stated in the [last post](#), our interface will use two different modes for interactive segmentation of an image: rectangle and scribble. Now it's time to define the procedure for scribble mode. Scribble mode is quite simple than rectangle mode as it consists on drawing scribbles over the image in order to select the object with positive markers and background with negative markers. Each time user draws a new scribble, client will ask server for an image which will contain the resulting mask for the selected object. This way, the user can see the region marked and hone his search until the region of interest matches with the resulting mask.

The difficulty remains in how to send the results of each search to the client in an efficient and feasible way. Last meeting helped us to define the packets of information we will send between client and server. From the limitation that we are working with a server with any memory to save previous results, packets from client to server need to contain all the necessary information to calculate the region from markers starting from scratch. Firstly we considered to send markers as a PNG image every time but there was a problem: we needed to know which scribble was drawn before, we needed to know marker's order.

Now on, we thought of different options and decided. These were our conclusions for the client packets:

- We needed a parameter for the **label**, which will define if the last scribble was a **negative** or a **positive** marker.
- Then the named **scribble**, which will be sended as a PNG image.
- Finally, two arrays containing the IDs of all positive and negative selected leaves: **bpt\_neg** and **bpt\_pos**. These arrays will contain the last label (the last marker drawn) for each leave.

This way, we just need to send a single image for each search and, at the same time, server will recover labels for each leave from the last search and will avoid recalculating all labels every time.

Therefore, server will send to the client:

- The **two arrays** (bpt\_neg and bpt\_pos). Client will work as an external memory for the server but won't need to process this information in no time.
- The **alfa** of the object selected sended as a PNG image of two colors (object and background). This alfa will be used by the client interface to show to the user the region selected as object with the markers drawn.

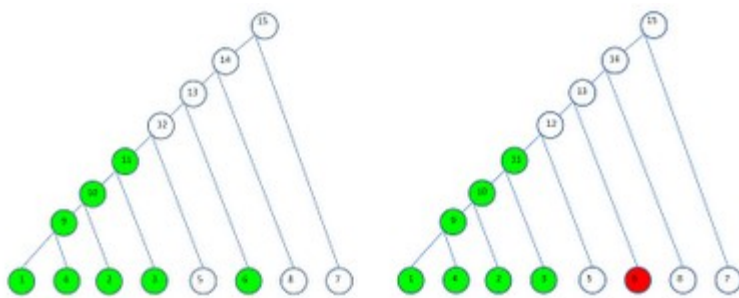
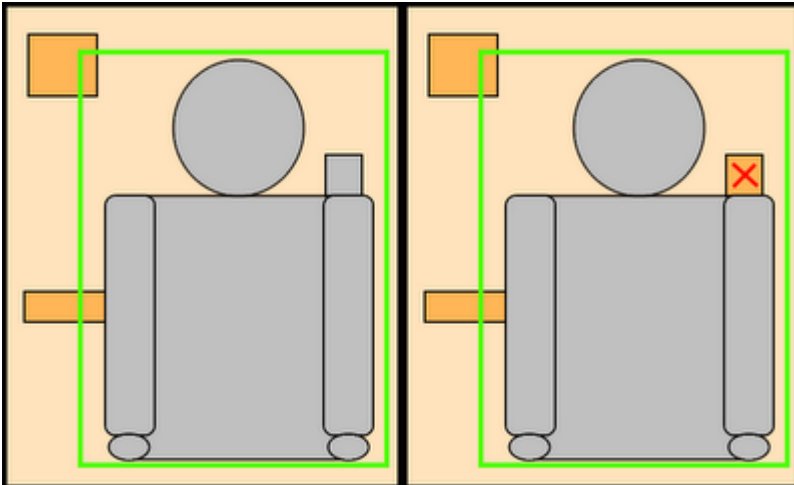
**Sunday, April 18, 2010**

### [Testing interactive segmentation with rectangles and points](#)

As I have been saying in my previous posts, the interface we are developing will allow an interactive segmentation of images with the aim of selecting a region of the global image. Our interface will hold two diferents modes: rectangle and scribble, which are defined by their markers: **rectangle** mode uses rectangles and points and **scribble** mode uses scribbles.

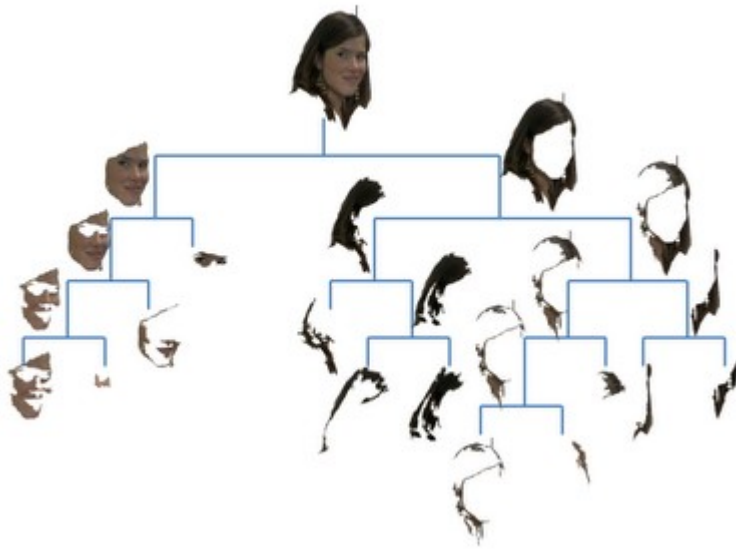
This time, we will focus on [rectangle mode](#). This mode will follow two steps as I explained [here](#): as the user draws a rectangle, we admit as object regions contained entirely within the rectangle. At this moment, every leave which is not labeled as object will be automatically labeled as background. Now on, the user has to hone the result with mouse clicks in order to select or deselect leaves until the selected object matchs with the region of interest. This method considers that every leave has one state (object or background) and every click is pointing a leave which is incorrectly labeled, that is to say, his state must be changed. The procedure is to change the state from object to background or vice versa.

Following the same example from the lastest posts, we get the results below:

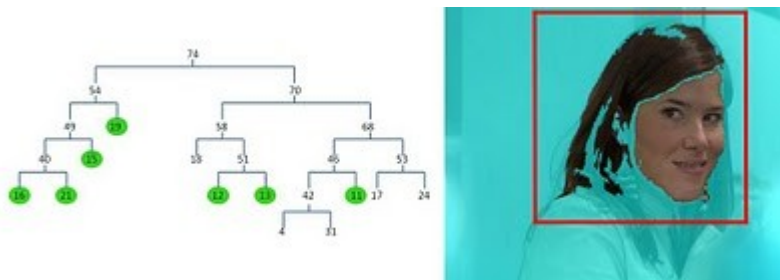


We can apply this methodology in a real example. Firstly, take a look of this piece of the BPT from the image:





If we draw a rectangle, the results are very rewarding:



Now we just need to adjust the result with 5 clicks and here is the final region:

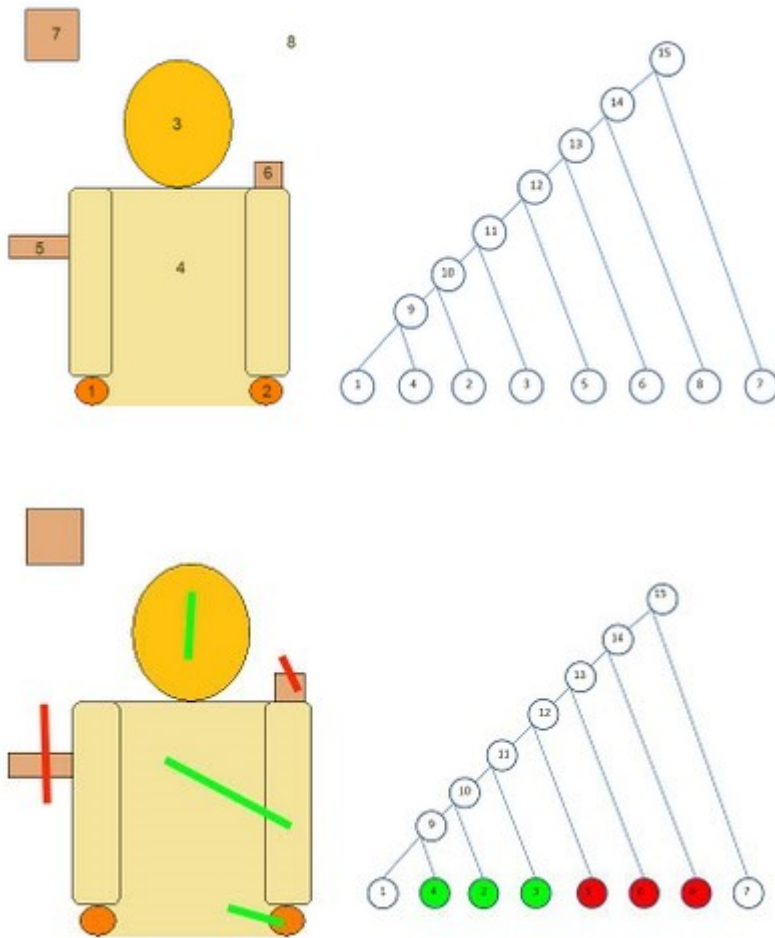


Saturday, April 10, 2010

**Binary Partition Trees by Luis Garrido and Tomasz Adamek**

During the last weeks I have been studying some proposals for interactive segmentation with [Binary Partition Trees](#) from other authors. Now I will introduce two different proposals:

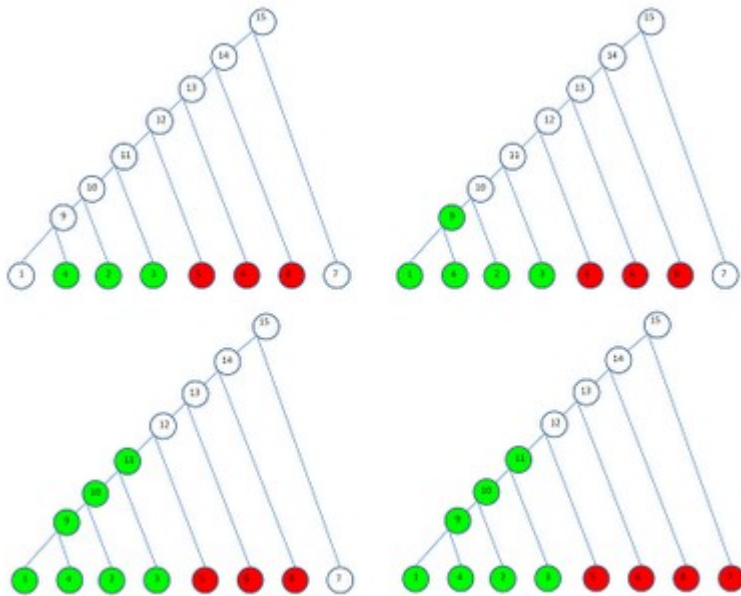
Both proposals begin with the same premise: as we can see in figures bellow, the conversion from markers to leaves is very simple. The BPT leaves touched by positive markers will be translated into positive leaves and the BPT leaves touched by negative ones will be negative leaves.



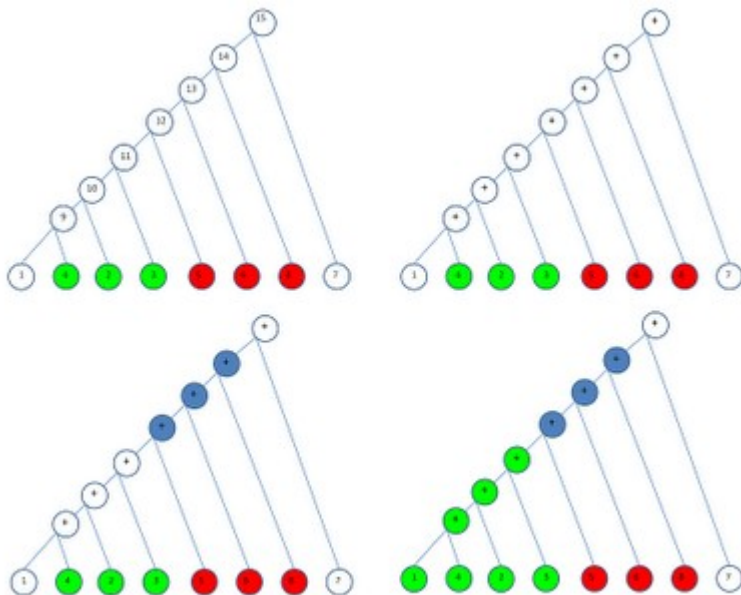
Garrido [1] expands every positive leaf to his parent whether his brother is not in conflict with him. In a binary partition tree, if the brother is not a negative node, the positive label will be expanded to the parent.

Take a look at the figure bellow that shows step by step the propagation of labels proposed by Garrido. The last step is to fill unlabeled nodes, which we call conflictive nodes. Firstly, conflictive nodes are assigned to the neighboring region but if more than one neighbor we distinguish different solutions: the region can be assigned arbitrarily but there are different parameters that can be taken into account such as similarity or some geometric criteria. In

this case, we have chosen the largest shared perimeter. Region 7 shares all his perimeter with region 8, consequently, region 7 is assigned to the background.



Adamek, meanwhile, proposes a method starting with a bottom-up and continues with a top-down propagaton. Bottom-up propagation is the upwards propagation from the first labeled node to the root. Then conflictive nodes are marked on the root, in the figure bellow conflictive nodes are painted blue. Adamek defines conflictive nodes as nodes whose children are in conflict, ie. with different labels.



The top-down propagation is the expansion of non-conflicting nodes to their childs (figure above, bottom right).

In this proposal also remain some unlabeled nodes. But Adamek does not use the binary partition tree to fill unlabeled nodes because he considers erroneus to use the tree to

solve its own limitations.

Adamek's methodology to fill unlabeled nodes proceeds as follows: Using Binary Partition Trees we can distinguish only two different regions: foreground and background. From here, we have to take a look to the unlabeled node and his neighbouring nodes.

If every node in contact with the unlabeled node has the same label, (object or background) then the node will be assigned to that label. If not, the distance between the unlabeled node and the closest adjacent node from each region in contact will be studied. This distance is calculated as Euclidean distance between average colours represented in LUV space for each two adjacent nodes. This way, we can find the confidence with which an unlabeled node can be assigned to background or foreground. This confidences will always add up to 1. The smallest confidence of each unlabeled node ( $R_i$ ) will be saved.

The idea is to find the smallest confidence by comparing the confidences ( $R_i$ ) of each unlabeled node. Then the "winner" is assigned to foreground or background depending on the last estimation  $R_i$ .

These calculations are repeated iteratively until all nodes are labeled.

In the example, node 7 will be assigned to the only region in contact: background.

[1] Garrido L, Salembier P (2000) Binary Partition Tree as an efficient representation for image processing segmentation and information retrieval.

[2] Tomasz Adamek (2006) Using contour information and segmentation for object registration, modeling and retrieval.

**Sunday, March 7, 2010**

### [Propagation of labels in the binary partition tree](#)

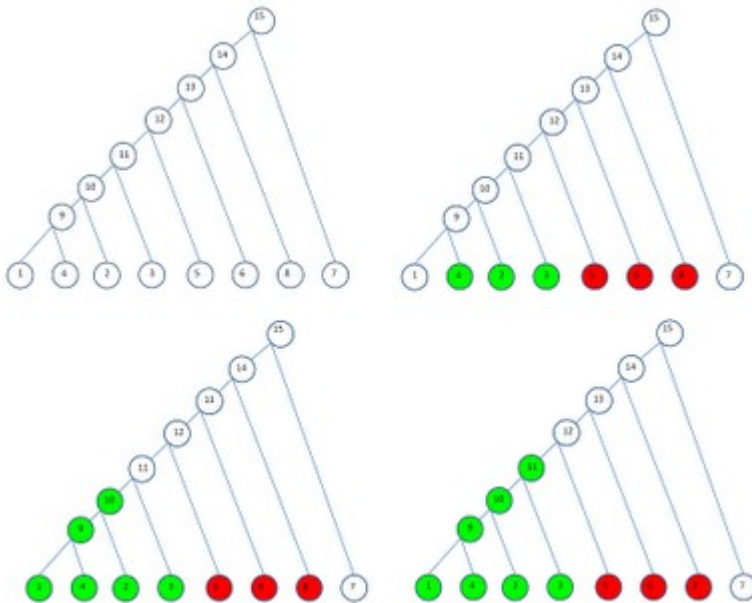
Focusing on propagation of labels, we assume some positive and negative leaves in the binary partition tree and we might decide where to position the neutral leaves.

Our algorithm works very restrictive so that the negative markers are not required to avoid a too large expansion in the branches of the tree. The criterion shall be as follows: for each positive node, its parent will be selected provided that the subtree of its brother has, at least, one positive node and no negative. This means that each negative node works as a barrier to the spread of the label.

Therefore, this system will be completely independent of the markers used to decide the label of each node. Consequently, the method will be valid for any marker.

Let's take a look at this example:





We will walk through the nodes and search for the first node that obeys the criterion: the first one, number 1, is not marked. The brother of the second one, number 4, does not match with its label. The third one, number 2, obeys the criterion. We can see the tree expansion from this node in the image above (bottom left). The tree on its right shows the last node in the branch converted into positive. Node number 11 can not keep expanding the positive label because its brother is a negative node.

### Interactive segmentation using rectangles and points

During the last weeks, we have decided to change a bit the method for the interactive segmentation. We will now do some tests mixing two markers: rectangles and points.

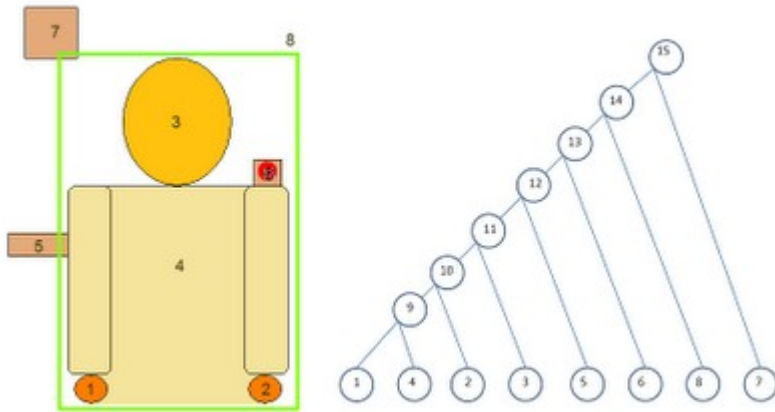
The reason for this change of mind is to achieve the better interactivity as possible for users. Rectangles seem to be very useful as we can mark a lot of regions just with a single mouse movement. The problem is the lack of precision if we want to remove small regions added to the first approximation of the object or, on the contrary, to add small regions that have been firstly discarded. This precision is provided by points (one mouse click), with which we can mark a single pixel from the image, that will correspond to one single leaf of the tree partition.

Remember that the methodology will be divided in two steps: [converting the markers into leaves](#) and [propagation of labels](#).

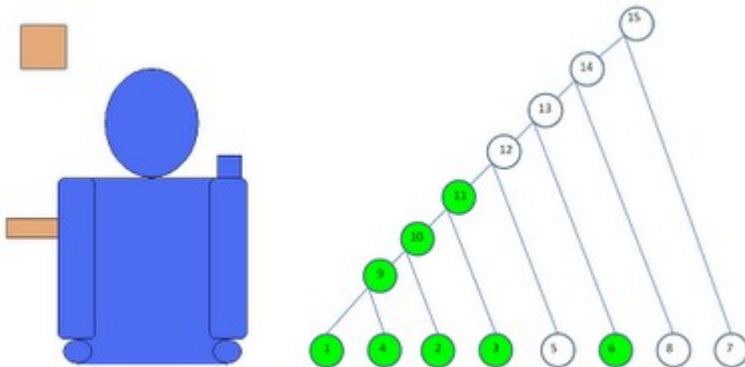
At the moment the idea is to restrict the number of markers to one rectangle and unlimited mouse clicks. From here, to convert the markers into leaves, we admit as object the regions contained entirely within the rectangle. Next step will be propagation of labels: to do so, the criterion explained [here](#) is changed a bit to become much restrictive because rectangles cover more regions than scribbles or points. Now, for each node, the label is only propagated to the parent if the two children are positive.

Now on, the user have to hone the result with mouse clicks in order to select or deselect leaves until the selected object matchs with the region of interest. To do this, each time the user makes a mouse click, the algorithm recalculates the result adding the negative leaves that will work as a barrier of the propagation or adding small regions that will become part of the whole object.

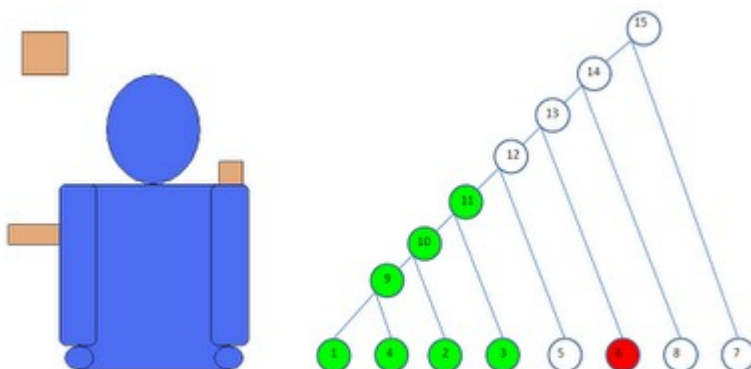
Lets see an example:



Firstly, the user draws the rectangle and the algorithm walks through the binary partition tree and gives a first result:

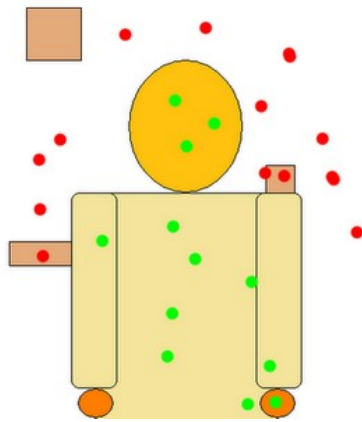


Then the user will mark with a point the brown design of the background, the algorithm will recalculate the resut and the interface will show the object bellow:



Saturday, February 27, 2010

[From markers to leaves in a binary partition tree](#)



As I explained in the [last post](#), we will use binary partition trees for the algorithm implementation of the interactive segmentation. We can divide the methodology in two steps: the first one will be converting the markers to leaves and the second one will be the propagation of labels.

This week, we will delve in the conversion from markers to leaves:

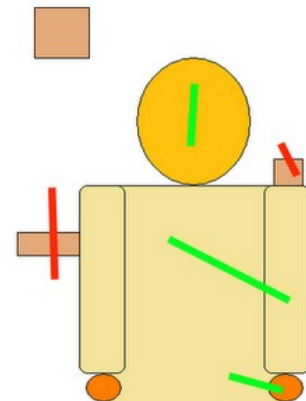
We can distinguish four types of leaves: positive, negative, neutral and leaves with positive and negative markers at a time.

As stated last week, we will use three types of markers: points, scribbles and rectangles.

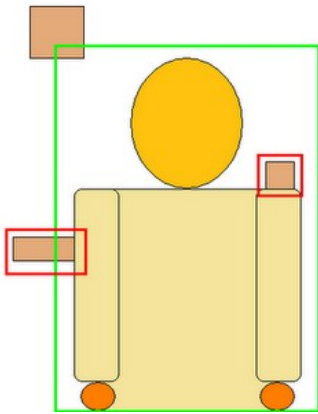
With points, every set of points have to be converted into leaves or single regions: If a region contains only positive points, it will be translated into a positive leaf. If a region contains only negative points, it will be translated into a negative leaf. And, finally, if the region has no point, it will be neutral.

The final consideration is: what if the same region has positive and negative points? In this case we may count the number of positive and negative points inside the region and reject it if negatives win or tie, that is to say, this leaf will be converted into a negative leaf. If there are more positive points, the leaf will be positive.

With scribbles, the proceeding will be quite similar: the regions containing only positive scribbles will be translated into positive leaves, regions containing only negative scribbles will be translated into negative ones and, at last, regions without any marker will be neutral leaves.



To decide where to place regions with positive and negative scribbles at a time, we will use the same concept as in the case of points. Therefore, we will look at the amount of pixels of both positive and negative scribbles inside the region and will decide according to it. If negative wins or ties, the leaf will be negative. If, on the contrary, positive wins, then the leaf will convert into positive.



At least, the third marker: the rectangle.

In this case, the procedure is a bit different: starting on the initial partition, all segments contained entirely within the positive rectangle or rectangles will be positive leaves.

Those within the negative rectangles will be negative leaves. But if the above criteria has placed them as positive, then must be converted to negative leaves.

Finally, leaves with no markers will be translated on neutral leaves.

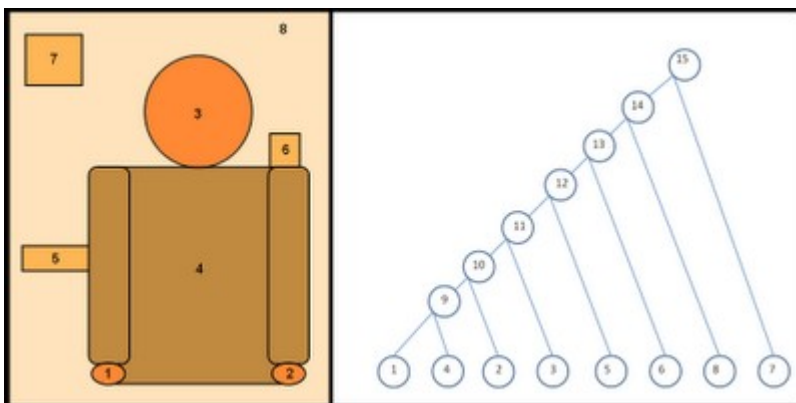
Sunday, February 14, 2010

### Binary Partition Trees

Now we have an idea about the markers we want to use. The next question is: how to do it? I might talk about the algorithm implementation: It will be based on Binary Partition Trees.

Partition trees firstly subject the image to a segmentation process depending on color, contour, texture, etc. that generates the initial partition. Secondly, single adjacent regions are combined into new regions according to their resemblance at different preseted points as color, texture, etc. This interative process leads to a region-based hierarchical representation of the image where leaves represent the different regions.

Specifically, we will use **BPT (Binary Partition Trees)** which are Partition Trees that combine neighboring regions two by two. Following the last post, there is an example below showing the initial partition of the anchorman with the white and brown background. The nodes located at the bottom represent the single regions and the nodes above are combinations of two of them. The node at the top represents the whole image. They were proposed by [Salembier and Garrido](#) in 2000.



Monday, February 8, 2010

### Interactive segmentation using scribbles, rectangles and points

My name is Pia and this days I'm working on my Bachelor Thesis. It will take 4 months and it will be developed at th office of CCMA-ASI in Esplugues, Barcelona.

Now it's time to present it: The aim is to exploit some tools they already have for image search. Nowadays, typical query by text is the base of these tools but the idea is to help them grow and improve step by step their robustness and complexity by combining improvement of algorithms, similar images, region selection and greater simplicity for users. This means working hard on the interface design. My work will focus on **interactive segmentation using scribbles, rectangles and points**.

#### **Implementation:**

Starting on this concept, I met Xavi and Neus last week to discuss the segmentation technique we wanted to implement for an easy and intuitive use. Let me expose the different options:

#### **Scribbles:**

The selection using scribbles consists on drawing lines with the mouse. The user will mark the parts of the image he is interested in, then the algorithm will take all the areas that the line is touching.

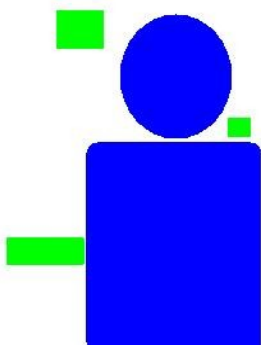
#### **Rectangles:**

With rectangles, the notion will be the same but now the region of interest will be every region contained entirely within the rectangle. That is to say, partially contained region will be rejected.

#### **Points:**

It could be interesting to select the areas we want to include in our search just with a set of points. This time the region of interest will be every region that contains any point.

There is something missing that way: how to dismiss an object that seems to be on the region of interest. We can create a new concept: the **negative or background markers**.



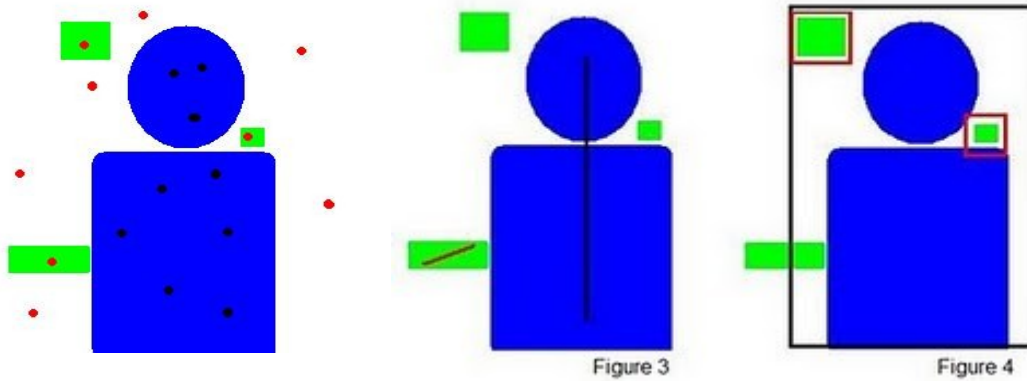
#### *Example:*

To understand this better, we will put an example: in this simplified drawing we see an anchorman. Behind him we find a white background with green designs. Supposing that we want to search the anchorman regardless of background, we may select him with our markers:

#### **Positive markers:**

The black line at *Figure 3* shows how to select the region using scribbles while the black rectangle at *Figure 4* shows how to select the region using rectangles and black points at *Figure 5*

show how to do it with points.



**Adding negative markers:**

In *Figure 3*, this green design at the left could be confused with the clothing of the anchorman. If we draw a negative trace on it, we are rejecting this region.

*Figure 4* shows the inclusion of two background figures within the rectangle of interest. If we draw new negative rectangles containing these figures we will avoid them.

In *Figure 5*, we can mark easily the regions we want to appear and the ones we don't want to. We should be careful with the ones which can be confused with clothing of the anchorman and don't forget to mark them with negative points.

### **Annex III: Fonts d'informació**

MXML vs AS3:

<http://en.wikipedia.org/wiki/MXML>

<http://www.madeinflex.com/2006/09/07/mxml-vs-as3/>

<http://flexar.blogspot.com/2009/01/04-introducin-mxml.html>

<http://opensource.adobe.com/wiki/display/flexsdk/MXML+2009mxml-vs-as3/>

<http://as3.es/>

<http://en.wikipedia.org/wiki/ActionScript>

[http://www.adobe.com/products/flex/flex\\_framework/](http://www.adobe.com/products/flex/flex_framework/)

Servei Web:

[http://es.wikipedia.org/wiki/Servicio\\_web](http://es.wikipedia.org/wiki/Servicio_web) (http Service)

<http://livedocs.adobe.com/flex/3/langref/mx/rpc/http/mxml/HTTPService.html>

Arquitectura distribuida:

<http://www.csi.map.es/csi/silice/Global69.html>

[http://es.wikipedia.org/wiki/Computaci%C3%B3n\\_distribuida](http://es.wikipedia.org/wiki/Computaci%C3%B3n_distribuida)

<http://www.fceia.unr.edu.ar/ingsoft/unidad14-4.pdf>

Programación por capas (multi-plataforma)

[http://es.wikipedia.org/wiki/Programación\\_por\\_capas](http://es.wikipedia.org/wiki/Programación_por_capas)

Objectes de valor:

<http://c2.com/cgi/wiki?ValueObject>

<http://www.yporqueno.es/blog/patron-value-object>

JSON:

<http://en.wikipedia.org/wiki/JSON>

CLUSTER:

[http://en.wikipedia.org/wiki/Cluster\\_analysis](http://en.wikipedia.org/wiki/Cluster_analysis)

<http://en.wikipedia.org/wiki/K-means>

Eines imatge:

[http://en.wikipedia.org/wiki/List\\_of\\_CBIR\\_Engines](http://en.wikipedia.org/wiki/List_of_CBIR_Engines)

FLEX:

[http://en.wikipedia.org/wiki/Adobe\\_Flex](http://en.wikipedia.org/wiki/Adobe_Flex)

[http://en.wikipedia.org/wiki/Adobe\\_Flash](http://en.wikipedia.org/wiki/Adobe_Flash)

[http://www.programacion.com/java/tutorial/ipintro/4/#4\\_modelomulticapa](http://www.programacion.com/java/tutorial/ipintro/4/#4_modelomulticapa)

<http://www.adobe.com/es/products/flex/>

<http://www.tufuncion.com/flex>

<http://flex.org/>

<http://flex.org/what-is-flex>

<http://www.desarrolloweb.com/articulos/que-es-adobe-air.html> AdobeAIR

ADOBE FLASH BUILDER

<http://www.adobe.com/products/flashbuilder/>

[http://en.wikipedia.org/wiki/Adobe\\_Flash\\_Builder](http://en.wikipedia.org/wiki/Adobe_Flash_Builder)

RIA.

[http://es.wikipedia.org/wiki/Aplicaciones\\_de\\_Internet\\_Ricas](http://es.wikipedia.org/wiki/Aplicaciones_de_Internet_Ricas)



[http://www.adobe.com/resources/business/rich\\_internet\\_apps/#open](http://www.adobe.com/resources/business/rich_internet_apps/#open)

HTTP:

<http://es.wikipedia.org/wiki/Http>

[http://en.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)

<http://www.w3.org/Protocols/>

[http://compnetworking.about.com/od/networkprotocols/g/bldef\\_http.htm](http://compnetworking.about.com/od/networkprotocols/g/bldef_http.htm)

[http://searchwindevelopment.techtarget.com/sDefinition/0,,sid8\\_gci214004,00.html](http://searchwindevelopment.techtarget.com/sDefinition/0,,sid8_gci214004,00.html)

[http://diveintopython.org/http\\_web\\_services/index.html](http://diveintopython.org/http_web_services/index.html)

[http://es.wikipedia.org/wiki/Localizador\\_uniforme\\_de\\_recursos](http://es.wikipedia.org/wiki/Localizador_uniforme_de_recursos) URL

<http://es.wikipedia.org/wiki/DNS> DNS

MIME:

<http://en.wikipedia.org/wiki/MIME>

<http://www.duamu.com/re/articulo/1346/id/549/articulos-que-son-los-mime-types.html>

ATM:

[http://es.wikipedia.org/wiki/Asynchronous\\_Transfer\\_Mode](http://es.wikipedia.org/wiki/Asynchronous_Transfer_Mode)

REST:

[http://en.wikipedia.org/wiki/Representational\\_State\\_Transfer](http://en.wikipedia.org/wiki/Representational_State_Transfer)

<http://www.xfront.com/REST-Web-Services.html>

[http://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm)

[http://en.wikipedia.org/wiki/Internet\\_media\\_type](http://en.wikipedia.org/wiki/Internet_media_type) MIME-TYPES