Indexation de vidéo couleur non compressée par le contenu à l'aide de la structure d'arbre R

Karim Abdelkader Henni et Lynda Zaoui

Département Informatique, Faculté des sciences, Université Abdelhamid Ibn Badis Mostaganem
Laboratoire Signaux, Données, Systémes, Université des Sciences et de la Technologie Mohamed Boudiaf, B.P.1505 EL M'NAOUAR-ORAN (ALGERIE)

Henni.2005@gmail.com
Zaoui _lynda@yahoo.fr

Résumé. Depuis le début des années 1990, l'indexation et la recherche par le contenu sont devenues un pôle très actif de la recherche; elle est devevue indipensable aux bases de vidéos afin de répondre aux besoins de plusieurs secteurs comme la télésurveillance, la télévision...etc

Cet article décrit notre approche de recherche et d'indexation de vidéos par le contenu, en suivant deux principaux points :

Dans un premier point nous proposons d'appliquer une structure d'indexation appelée "arbre R" pour indexer une vidéo en procédant à son découpage en images individuelles, puis à sa segmentations temporelle et enfin nous sélectionnons les images clefs, l'ensemble de ces images formant le résumé vidéo. Ce premier point correspond à la phase off-line du prototype.

Dans un deuxième point, l'utilisateur a la possibilité d'introduire au système une image requête, et sur la base de calcul de distance de similarité le système renvoie des résultats. Ce deuxième point correspond à la phase on-line du prototype.

Mots clés. Indexation de vidéos par le contenu, arbre R, distance de similarité, segmentation temporelle, image clef, plan, résumé vidéo.

1 Introduction

Avec les progrès de l'informatique et des télécommunications (Internet ADSL, réseaux mobiles,…etc.), les bases de vidéos sont apparues. Ces bases sont amenées à stocker des millions de vidéos, ce qui a engendré de nouveaux besoins dans plusieurs domaines dont nous citons par exemple :

- Le secteur de la télévision s'intéresse à accélérer la recherche dans les archives de télévision.
- Les téléspectateurs ont besoin d'avoir une idée générale sur la vidéo sans devoir la visualiser entièrement (ç.à.d avoir un résumé de la vidéo).

- Dans le domaine de la télésurveillance les utilisateurs ont besoin de détecter et de faire le suivi des objets dans des séquences vidéos filmées.
- Le grand public désire retrouver un plan spécifique contenant une image X d'un film ou d'un journal télévisé.

Quelle que soit la base de données sur la quelle l'utilisateur travaille, ce dernier a toujours recours à une interrogation pour avoir une information précise ; la recherche d'une vidéo dans une base de plusieurs milliers de vidéos nécessite un temps considérable ; pour accélérer le processus, le concept d'*indexation* est alors employé.

Au début l'indexation consistait à donner une description textuelle de la vidéo. Ce type de méthodes (basées sur des annotations textuelles) présente des inconvénients. En effet, la description textuelle demande une intervention humaine ; et celle-ci a deux limites principales : le temps et le jugement; sans parler des limitations liées à la langue (ou au jargon) utilisée pour l'indexation.
D'une part, présenter le contenu d'une vidéo demande beaucoup de temps ("voiture bleue qui suit une voiture rouge" par exemple). Pour des ensembles d'une centaine de vidéos cela est encore imaginable mais actuellement les volumes étudiés sont plutôt de l'ordre du millier voir de la centaine de milliers de vidéos. Une intervention humaine pour décrire de tels ensembles semble donc impossible.

Par conséquent, et ne pouvant pas se baser entièrement sur une description textuelle, la recherche de vidéos tente aujourd'hui d'extraire les informations directement des vidéos et d'une manière automatique. Ce type de travaux est couvert par le terme : "*indexation de vidéos basée sur le contenu*". Dans ce type de méthodes (méthodes basées sur le contenu visuel) deux étapes sont importantes : une première étape est la segmentation en groupes d'images appelés "plans". Les plans sont des suites d'images filmées sans interruption. A la fin de chaque plan il existe une coupure. Par l'analyse des images consécutives il est possible de détecter automatiquement un changement de plan. La deuxième étape est d'associer des concepts aux plans détectés, les plans sont souvent représentés par des images clefs qui peuvent être analysés pour obtenir une description du contenu [1].
Il existe deux approches pour la détection des changements des plans : détection sur les vidéos compressée et détection sur les vidéos non compressées.
La détection des plans pour la vidéo non compressée est basée sur l'application de quelques mesures capables d'évaluer les changements entre images successives.
La détection des plans pour la vidéo compressée est basée sur la comparaison de certains coefficients spécifiques au codage MPEG.

2 Structure de l'arbre R

L'arbre R (noté AR) est une structure hiérarchique, dérivée de l'arbre B et utilisée pour indexer les objets spatiaux ou géométriques. Les objets spatiaux sont représentés par des rectangles englobant minimum (noté REM) sur une image [2], ceci est intéressant ; en effet on arrive à décrire des objets très complexes avec un

nombre limité d'octets[1], puisque à chaque objet de cet espace on associe uniquement les coordonnées du REM et une description de l'objet [3].

2.1 Caractéristiques des arbres R

L'arbre R, comme l'arbre B, est un arbre équilibré de degré (ou ordre) *m* et présente les caractéristiques suivantes [2], [4], [5] :

- Un noeud interne d'arbre R permet d'adresser les noeuds du niveau suivant de l'arborescence.
- Un noeud feuille référence un ensemble d'objets spatiaux.
- Un noeud (interne ou feuille) est associé à une partie rectangulaire de l'espace de sorte que :
  - Le rectangle associé à un noeud feuille est le rectangle englobant minimum des *rem* des objets spatiaux référencés par le noeud feuille.
  - Le rectangle associé à un noeud intermédiaire est le rectangle englobant minimum des rectangles englobants minimums associés à chacun de ces fils.
  - La racine de l'arborescence est associée au rectangle englobant minimum de l'espace tout entier.
- Chaque noeud contient des entrées qui sont des couples (*rem, oid*) tels que :
  - Pour les noeuds internes : rem correspond au rectangle englobant minimum associé à un noeud fils, dont l'adresse est *oid*.
  - Pour les noeuds feuilles : rem est le rectangle englobant minimum d'un objet spatial dont l'adresse est *oid*.
- Chaque nœud, à l'exception de la racine, contient un nombre d'entrées compris entre *m* et *M*, où *m* est le nombre minimal d'entrées par noeud et *M* est le nombre maximal d'entrées, tels que $0 \leq m \leq [M/2]$; $[M/2]$ étant égale à la partie entière de $(M/2)$.
- La racine de l'arbre R possède au moins deux fils sinon c'est une feuille.
- Toutes les feuilles apparaissent au même niveau.
- A chaque objet est associé une entrée composée du REM et de sa description, celle-ci peut se faire à l'aide de mots clés.
- A chaque rectangle de l'image on associe une clé, c'est-à-dire une valeur entière permettant de définir sa position et ses dimensions. Ainsi le parcours dans l'arbre se fera par simple comparaison des clés.

[1] Un rectangle à deux dimensions peur être représenté par 4 nombres de 4 octets, chacun. Si un pointeur prend 4 octets, chaque entrée demande 20 octets.

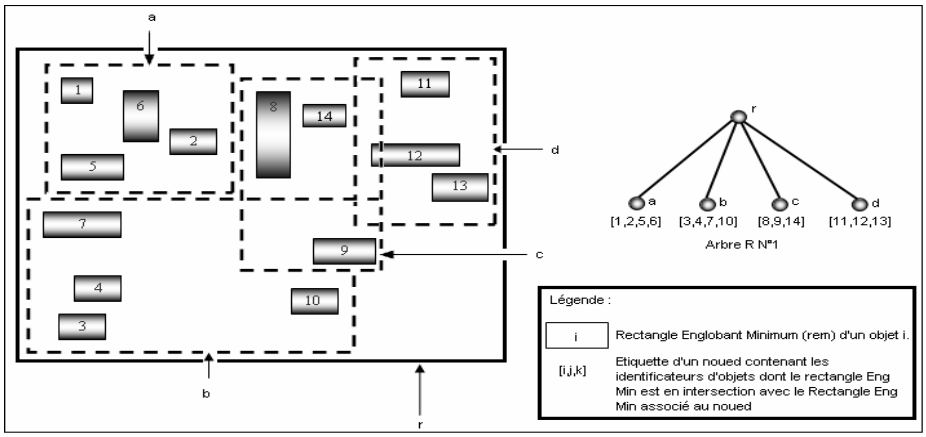

Fig. 1. Un arbre R.

2.2 Distance de R-Similarité

M. Rukoz, M. Manouvrier , G. Jomier dans [6] ont défini une distance générale Δ entre images basées sur les arbres quaternaires, définie par :

$$\Delta(i, j) = \frac{\sum_k c_k \, \delta_k(i, j)}{\sum_k c_k} \qquad (1)$$

– *k* est l'identificateur d'un nœud pris parmi l'union des identificateurs de nœuds apparaissant dans les arbres quaternaires des images *i* et *j*.
– $\delta_k(i, j)$ distances entre nœuds d'arbres quaternaires.
– c_k est un coefficient positif représentant le poids du nœud *k* dans le calcul de la distance Δ. Le choix de chaque poids c_k dépend des besoins de l'utilisateur, c'est-à-dire de l'importance que l'utilisateur souhaite donner à certains quadrants de l'image par rapport à d'autres dans le calcul de la distance Δ.

Remarque 1. $\Delta(i, j) = 0$ signifie que "tous les nœuds homologues *k* des arbres quaternaires *i* et *j*, de poids c_k non nul, ont une distance δ_k nulle ($\forall$k tel que $c_k \neq 0$: $\delta_k(i, j) = 0$)" et $\Delta(i, j) = 1$ signifie que "tous les nœuds homologues *k* des arbres quaternaires *i* et *j*, de poids c_k non nul, ont une distance δ_k égale à 1 ($\forall$ k tel que $c_k \neq 0$: $\delta_k(i, j) = 1$)".

En faisant référence aux arbres quaternaires, les arbres R ont aussi des nœuds, ces noeuds ne correspondent pas aux quadrants retrouvés par un découpage quaternaire mais plutôt aux rectangles englobants minimums des objets spatiaux, nous pouvons définir une distance basée sur les arbres R à partir de la distance Δ.

Nous supposerons que tous les REM ont le même poids dans le calcul de la distance $c_k = 1 \ \forall$ k et donc :

$$\Delta(i, j) = \frac{\sum_k \delta_k(i, j)}{\sum_k c_k} \quad (2)$$

Comme les noeuds d'arbre R ne contiennent pas des informations sur les sous arbres dont ils sont racines ; dans notre cas la distance δ_k ne peut prendre que deux valeurs : 0 ou 1. Si les deux régions à comparer sont identique alors $\delta_k = 0$ sinon $\delta_k = 1$.
Dans $\sum_k \delta_k(i, j)$ on aura que des REM différentes.
$\sum_k c_k$ nous donnera en fait le nombre de REM pris parmi l'union des REM qui apparaissent dans les deux images *i* et *j*.
Notre distance est appelée "*distance de R-similarité*" ; $R(i, j) \in [0, 1]$ et est définie par l'équation suivante :

$$R(i, j) = \frac{| S(i, j) |}{| U(i, j) |} \quad (3)$$

Avec $| S(i, j) |$: nombre de nœuds et de rem différents entre les images *i* et *j*.
$| U(i, j) |$: nombre total (sans doublon) des identificateurs des nœuds et des rem apparaissant dans les arbres R des images *i* et *j*.

3 Notre approche

Notre approche d'indexation et de recherche de vidéos par le contenu comporte cinq étapes incontournables décrites dans ce qui suit :

3.1 Découpage en images

Il s'agit dans cette étape de restituer les images individuelles qui ont formé la vidéo grâce aux algorithmes et aux techniques de programmation ainsi que des logiciels de traitement de vidéo et de capture d'écran existant sur le marché comme Virtual dub, OSS video decompiler… ; ces images nous permettront par la suite de définir des plans d'images en utilisant des distances de similarité.

3.2 Segmentation spatiale et construction de l'arbre R

La segmentation d'images constitue un élément clé entre traitement et analyse de l'image, elle permet de passer du support de l'image à une liste d'objets décrits en terme de régions [7].

Dans cette étape il s'agit de segmenter chaque image individuellement (qui est le résultat de l'étape précédente) en utilisant un des algorithmes de segmentation existants.

Cette étape sert à construire l'arbre R pour chaque image segmentée spatialement, et par conséquent à calculer les distances de R-Similarité entre les images, elle est transparente à l'utilisateur et elle a une utilité pour la segmentation temporelle.

3.3 Segmentation temporelle

La segmentation temporelle est une étape très importante pour les approches d'indexation et de recherche de vidéos par le contenu.

La majorité des approches utilisent des critères de bas niveau comme la couleur, la texture ou le mouvement [8], [9] pour segmenter la vidéo en plusieurs unités de base appelées "*plans*", quatre catégories de méthodes existent détaillées ci dessous :

- Les méthodes basées sur les différences pixel à pixel détectent un changement de plan en calculant une différence entre les pixels de l'image à l'instant *t* et ceux de l'image à l'instant *t+1*. Si le nombre de pixels différents est supérieur à un seuil, alors on considère que l'on est en présence d'un "*cut*".
- Les méthodes à base d'histogramme comparent deux images successives en s'appuyant sur leurs histogrammes respectifs. Une différence des deux histogrammes est calculée et comparée à un seuil. Un changement de plan est détecté si la différence obtenue est supérieure au seuil. Du fait de l'utilisation d'histogrammes, il est possible de ne pas détecter un *cut* si les deux images concernées ont un histogramme similaire mais un contenu différent.
- Les méthodes basées sur une estimation du mouvement utilisent l'information de mouvement comme critère principal pour la détection des changements de plan. Les mouvements sont estimés pour chaque pixel d'une image obtenue à l'instant *t*, et sont comparés avec ceux de l'image correspondant à l'instant *t+1*. Un nombre trop important de mouvements incohérents entre les deux images successives implique alors la détection d'un changement de plan.
- Les méthodes basées sur les blocs sont des méthodes intermédiaires entre les méthodes basées pixels (locales) et les méthodes basées histogrammes (globales).

3.3.1 Notre algorithme

Le plan d'images est l'unité technique sur laquelle se base notre approche d'indexation et de recherche de vidéo par le contenu, nous proposons dans ce qui suit un algorithme qui permet de segmenter une vidéo en plans à l'aide des arbres R ; l'algorithme est décrit dans ce qui suit :

1- Obtenir l'ensemble F des images individuelles avec l'étape 1.
2- Construction de l'arbre R pour chaque image de l'ensemble F avec l'étape2.
3- Le 1er plan de la vidéo contient uniquement la 1^{ere} image de l'ensemble F.
4- On calcule la distance de R similarité entre l'image suivante et la 1^{ere} image du plan courant.
5- Si la distance est inférieure au seuil (ce dernier est défini de manière empirique sur un jeu de tests) alors cette image fait partie de ce plan et.
5.1- Tant qu'on a pas atteint la fin de l'ensemble F, on passe à l'image suivante de la séquence et.
5.2- Revenir au 5.
6- Sinon l'image courante génère un nouveau plan et aller à l'étape 4.
7- Dés qu'on atteint la fin de l'ensemble F, on s'arrête.

3.4 Sélection de l'image clef

Après avoir déterminé les plans dans l'étape précédente, nous devons extraire dans cette étape les caractéristiques visuelles de chaque plan. Ces caractéristiques sont définies dans une ou plusieurs images appelées "*images clefs*" (appelées parfois images représentatives, ou images caractéristiques).

Différentes approches ont été proposées jusqu'ici pouvant être classées en trois catégories :

- Approches se basant sur les différences visuelles entre images : ces approches utilisent donc les caractéristiques brutes obtenues sur les images (couleur, texture, mouvement).
- Approches se basant sur les configurations du mouvement : ces approches sélectionnent les images représentatives en fonction de l'intensité ou des variations du mouvement.
- Une approche différente consiste à créer une mosaïque du plan. Une mosaïque est construite à partir des images du plan et fournit une représentation complète et compacte de la vidéo. Cette approche est beaucoup plus rare car les mosaïques sont difficiles à construire dans le cas général [10].

L'image clef dans notre approche est la première image de chaque plan car le plan contient les images similaires à cette image ; si la distance de similarité est supérieure au seuil nous concluons qu'elle est différente de la première image et forme à son tour un autre plan.

Les images clefs sont les images les plus riches en information par rapport aux autres images. L'ensemble de ces images forme ce que l'on appelle "*résumé vidéo*".

En fait ces quatre étapes correspondent à la phase off-line du prototype, la dernière étape correspond à la phase on-line.

3.5 La recherche dans les vidéos

Dans les bases d'images, les recherches consistent à savoir si l'image requête (l'image introduite par l'utilisateur dans la phase on-line) est stockée dans cette base d'images ou bien à extraire de cette base toutes les images similaires à l'image requête au sens d'une distance de similarité.

Les bases de vidéos ne sont aujourd'hui qu'à leur début, nous pouvons par conséquent imaginer quelques définitions de recherche dans les vidéos selon les domaines d'application de l'indexation et des besoins des utilisateurs, par exemple la recherche dans les vidéos consiste en :

- La recherche de l'exacte réplique de la vidéo introduite par l'utilisateur dans la base de vidéos.
- Rechercher des vidéos proches de la vidéo requête.
- Détection des objets particuliers (visages, véhicules, bâtiments,….) dans les bases de vidéo.
- Rechercher dans la base une vidéo contenant l'exacte réplique d'une image requête ou bien un ensemble d'images similaires à l'image requête.

Nous avons tracé en bas un schéma explicatif montrant les deux phases de notre système d'indexation et de recherche de vidéo par le contenu pour les deux derniers types de recherche énoncés plus haut.

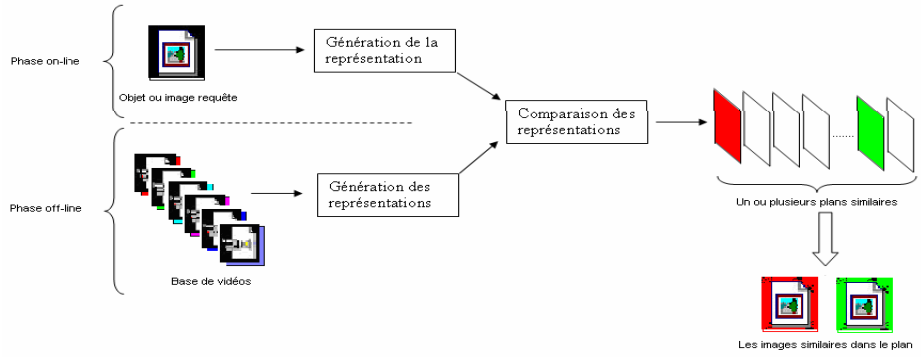

Fig. 2. Principe de fonctionnement d'un système d'indexation de vidéos par le contenu.

4 Résultats d'expérimentation

Nous avons évalué notre approche sur une séquence vidéo couleur non compressée (d'une forte variation du contenu) au format AVI ayant les caractéristiques suivantes :

- Taille (size) : 128 x 128
- Longueur (lenght) : 0 :07 :000 (7 secondes)
- Images (frames) : 105
- FPS (frames per seconds) : 15.00

La 1^{ére} étape consiste à découper la vidéo en images individuelles, les résultats sont illustrés par la figure suivante :

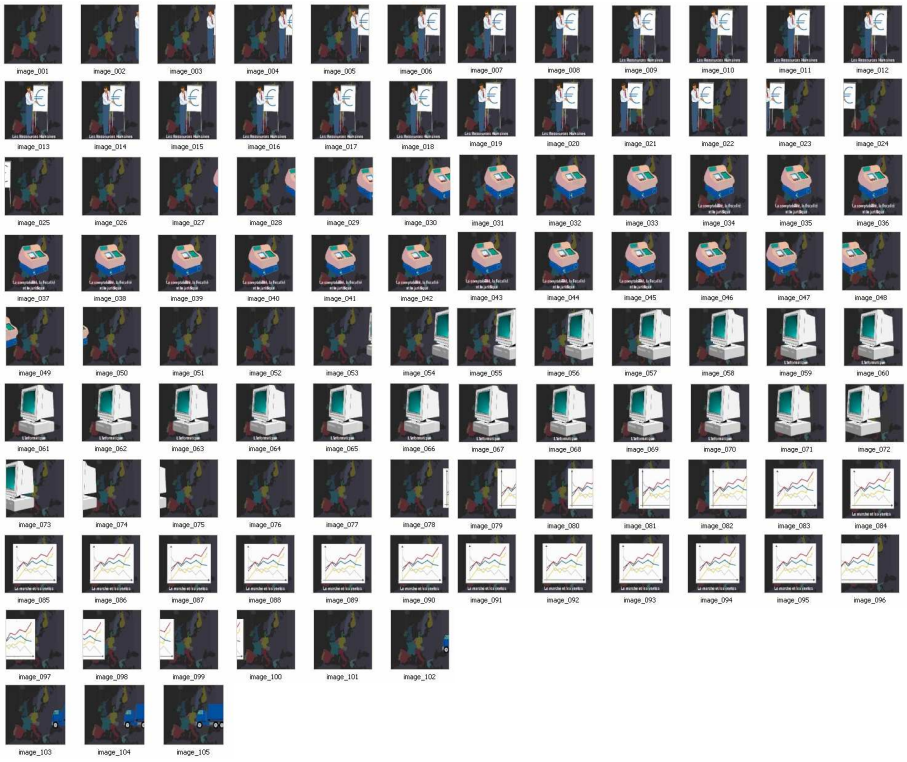

Fig. 3. Images individuelles de la vidéo de l'exemple.

La 2eme étape de notre approche n'est pas illustrée par le prototype car c'est une étape de codage, elle est transparente à l'utilisateur, dans cette étape chaque image est segmentée spatialement et stockée sur disque sous forme d'arbre R.

La 3eme étape consiste à segmenter la vidéo en plans en se basant sur les arbres R des images, les résultats sont les suivants :

Le plan 0 englobe les images : 1,2,…,20 (20 images).
Le plan 1 englobe les images : 21 (1 image).
Le plan 2 englobe les images : 22,23,…,29 (8 images).
Le plan 3 englobe les images : 30,31,…,33 (4 images).
Le plan 4 englobe les images : 34,35,…,52 (19 images).
Le plan 5 englobe les images : 53,54,…,70 (18 images).
Le plan 6 englobe les images : 71,72,…,77 (7 images).
Le plan 7 englobe les images : 78,79,…,82 (5 images).
Le plan 8 englobe les images : 83,84,…,95 (13 images).
Le plan 9 englobe les images : 96,97,…,101 (6 images).
Le plan 10 englobe les images : 102,103,…,105 (4 images).

La 4eme étape consiste à extraire les images clefs des plans, ce qui donne un résumé de la vidéo, les résultats sont illustrés par la figure suivante :

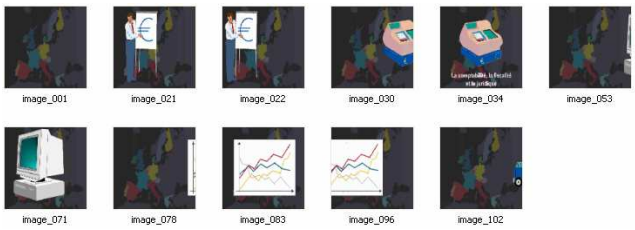

Fig. 4. Résumé vidéo.

Ces étapes correspondent à la phase off-line du prototype, l'utilisateur a la possibilité d'interroger la base lors de la phase on-line, nous avons introduit dans notre prototype deux images requêtes, une qui fait partie de la vidéo et l'autre qui n'en ne fait pas, les deux images requêtes sont illustrées par la figure suivante :

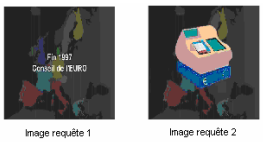

Fig. 5. Deux images requêtes introduites au système d'indexation.

Lorsque l'utilisateur introduit l'image requête 1 dans notre prototype, le système calcule la distance de similarité entre chaque image individuelle de la vidéo de la base de données et l'image requête et extrait les images individuelles les plus similaires à cette image requête, ces images similaires peuvent appartenir aux différents plans, si parmi les distances calculées, n'existe pas une distance nulle alors le système affiche à l'utilisateur que l'image ne figure pas dans la vidéo.

L’utilisateur a la possibilité de voir les détails de la recherche, c’est-à-dire les images individuelles les plus similaires à la requête et les plans auxquels elles appartiennent.

Table 1. Détails de recherche pour l’image requête 1

N_PL	N_IMG	DISTANCE
0	1	0.375
2	26	0.375
4	51	0.375
4	52	0.375
6	76	0.375
6	77	0.375
9	101	0.375

Pour l’image requête 2, le système procède de la même manière, il trouve que la distance de similarité est nulle et dans ce cas là il affiche à l’utilisateur que l’image figure dans la vidéo.

Table 2. Détails de recherche pour l’image requête 2

N_PL	N_IMG	DISTANCE
3	33	0

En fait ces résultats sont obtenus avec la valeur de 0.9 du seuil, nous avons testé notre prototype sur le même exemple mais avec des valeurs différentes du seuil.

Table 3. Nombre d’images clefs en fonction des valeurs du seuil

Seuil	Nombre d’images clefs
0.1	65
0.2	60
0.3	53
0.4	43
0.5	31
0.6	25
0.7	20
0.8	15
0.9	11

Avec un petit seuil le nombre d’images clefs est relativement grand car un petit changement du contenu d’une image à une autre provoque la définition d’une nouvelle image clef, mais plus le seuil est grand moins il y a de chances de tomber sur une image qui varie beaucoup du contenu au point de définir un nouveau plan ce qui amène à avoir moins d’images clefs.

Un bon résumé de vidéo ne veut pas dire moins d'images clefs possibles, pour notre exemple un résumé de 2 images clefs ne donnerait pas un aperçu significatif et général de la vidéo ; en contre partie un résumé de 65 images clefs (avec un seuil de 0.1) exclut complètement la notion de résumé, ce n'est plus un résumé mais plutôt une partie de la vidéo, tout dépend en fait du contenu de la vidéo.

5 Conclusion

A travers cet article nous avons proposé une approche de recherche et d'indexation de vidéos par le contenu en utilisant la structure d'arbre R. Le prototype développé comporte deux principales phases : phase off-line qui consiste à indexer toutes les vidéos de la base par cette structure ce qui correspond à un autre codage de la vidéo et phase on-line dans laquelle l'utilisateur a la possibilité d'effectuer des recherches fines au sein de ces vidéos ; en effet dans cette phase l'utilisateur introduit une image ou une vidéo requête et le système l'indexe selon notre structure. Maintenant que les vidéos (y compris la vidéo requête) sont représentées par la même structure, le principe consiste à calculer la distance de similarité et d'extraire de la base les vidéos similaires à la vidéo requête selon le seuil qui a été défini.

Références

1. M. Scuturici "Contribution aux techniques orientées objets de gestion des séquences vidéo pour les serveurs Web". Thèse de Doctorat en Informatique. Institut National des Sciences Appliquées de Lyon. 2002.
2. M. Manouvrier "Objets similaires de grande taille dans les bases de données". Thèse de Doctorat en Informatique. Université Paris IX-Dauphine. 2000.
3. H. Abed : "Représentation et stockage des images similaires par des structures arborescentes". Mémoire de Magister en Informatique. Université des Sciences et de la Technologie d'ORAN Mohamed Boudiaf. 2002.
4. A. Guttman. R-trees : "A Dynamic Index Structure For Spatial Searching". Dans proc . Of ACM SIGMOD Int .Symp. On the management of data, pages 45-57, 1984.
5. http://www.infres.enst.fr/people/saglio/etudes/Strasbrg.html
6. M. Rukoz, M. Manouvrier, G. Jomier "Distance de similarité d'images basées sur les arbres quaternaires". Dans CNRS – CONICIT. 2002. pages 1 à 20.
7. http://www.lis.inpg.fr/pages_perso/chassery/francais/recherche.html
8. http://www.afrif.asso.fr/archive/rfia2004/ARTICLES.html
9. http://bat710.univ-lyon1.fr/ligim/textes/equipes/IMAGE/CORESA03/articles/html
10. F. Souvannavong : "Indexation et recherche de plans vidéo par le contenu sémantique". Thèse de Doctorat en Informatique. Ecole Nationale Supérieur des Télécommunications (Paris). 2005.