



Indexation vidéo par contenu

Content Based Video Retrieval (CBVR)

A. ELHASSOUNY

GL
ENSIAS

Master B2dS

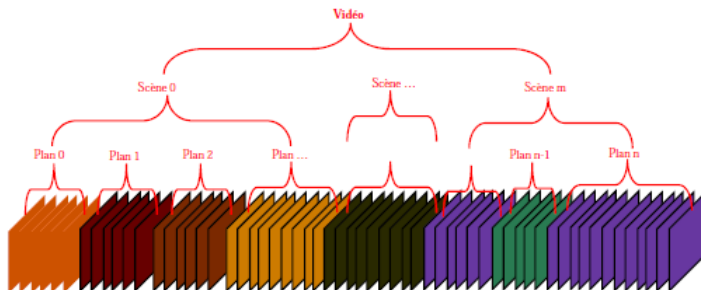


PLAN

- 1 Indexation vidéo par contenu
 - CBVR : Introduction
 - Analyse de la vidéo
 - Segmentation de la vidéo
 - Création de résumé de la vidéo

Qu'est ce qu'un document vidéo ?

- Un document vidéo est constitué
 - D'un signal audio : une ou plusieurs bandes son
 - D'un signal vidéo qui est constitué d'un nombre important de frames défilées de plus de 20 fps (frames par sec)
 - Vidéo = suite d'images + son + texte
- Le modèle cinématographique du signal vidéo



Qu'est ce qu'un document vidéo ?

Le modèle cinématographique du signal vidéo

- Les frames (frame) : image de la vidéo représentant le contenu visuel du plan
- Un plan
 - Il est défini par une séquence de frames durant laquelle l'acquisition du signal n'a pas subi d'interruption
- Une scène
 - Elle regroupe par définition l'ensemble des plans consécutifs se situant dans le même espace et au même instant, ou ayant un lien sémantique étroit
 - Elle représente la plus petite unité sémantique d'un film
 - Exemple (un dialogue, une conversation téléphonique, une action observée de plusieurs points de vue, deux évènements se produisant en parallèle, etc)
- Une vidéo peut être décomposée en plusieurs scènes, et chaque scène peut être composée d'un ou de plusieurs plans, et un plan de plusieurs frames

Qu'est ce qu'un document vidéo ?

Représentation d'une vidéo numérique

Le flux de données dans la vidéo numérique est alors défini par :

$$F = Nb * f_y * f_x * f_t (\text{bits/s})$$

- On caractérise donc la vidéo numérique par les paramètres suivants :
 - f_t : Fréquence d'image (frames per seconds (fps))
 - f_y : Lignes par frame
 - f_x : Pixels par lignes
 - Nb : Bits à encoder, ce paramètre encode la valeur d'intensité (couleur) d'un pixel
- Une vidéo numérique peut être obtenue par la discrétisation et l'encodage de la vidéo analogique ou en utilisant directement une série d'acquisition numérique (capteur numérique)

► OpenCV : modification temps réel d'un flux vidéo

Indexation de la vidéo

Applications

- la télévision numérique
- les archives de télévision
- les bibliothèques numériques
- Internet
- les vidéothèques d'entreprise
- l'éducation, la formation
- la culture, les loisirs

Indexation de la vidéo

Applications

- Programme télévision
 - exemple : quelques minutes après le décès d'un personnage public, les télévisions préparent des dossiers rétrospectifs
- Caméras de surveillance
 - Les entreprises ont des grandes quantités de données vidéos mais si on ne sait pas ce que ces vidéos contiennent, ou comment retrouver une information, cela est inutile ...
- Films
 - Les assistants de production passent des heures à visionner des vidéos à la recherche de bons segments
- ...le problème n'est pas d'avoir le matériel vidéo mais de le trouver !

Indexation de la vidéo

Requête ?

- Requête sur :
 - un ensemble de vidéos ?
 - une seule vidéo ?
- Requête : sous quelle forme ?
 - usage des meta données ?
 - description textuelle d'une scène ?
 - une image d'un film ?
 - l'image d'un acteur du film ?
 - requête vidéo



Indexation de la vidéo

Indexation de la vidéo par Meta-données

- Meta-données
 - Auteurs, réalisateurs, producteurs, acteurs, autres professionnels associés
 - Date de création
 - Informations sur les conditions de production
- Données textuelles
 - Critiques du film
 - Sous-titres
- Flux de données associé = le son
 - Événements sonores
 - Transcription

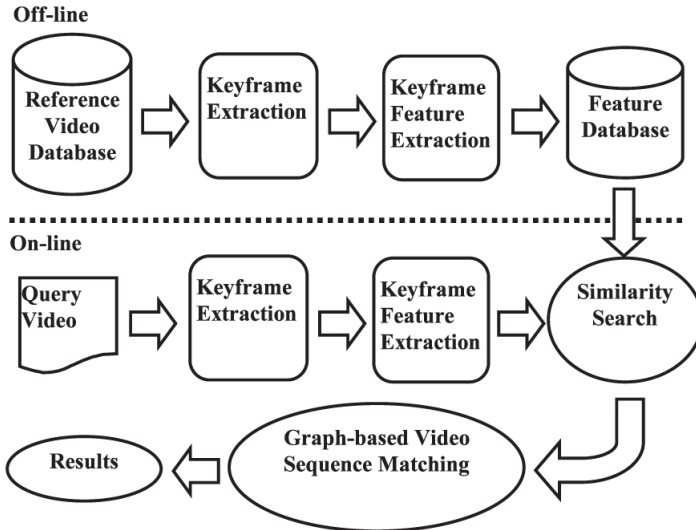
Indexation de la vidéo

Indexation de la vidéo par contenu ?

- Requête image
 - plus difficile (car moins d'information dans la requête, et pas de description spatio-temporelle possible)
- Requête vidéo (souvent un plan)
 - Indexation spatio-temporelle : indexation au niveau des plans plutôt que des images composant la vidéo
 - Segmentation temporelle en "plans" (au sens cinématographique)
- ... en plus ... difficultés de l'indexation dues aussi :
 - À la quantité de données : 100Mo à 120 Go par heure de programme TV
 - Aux formats de stockage de ces données (codeurs vidéo)
 - Coût ??
 - Grande redondance de l'information au cours du temps

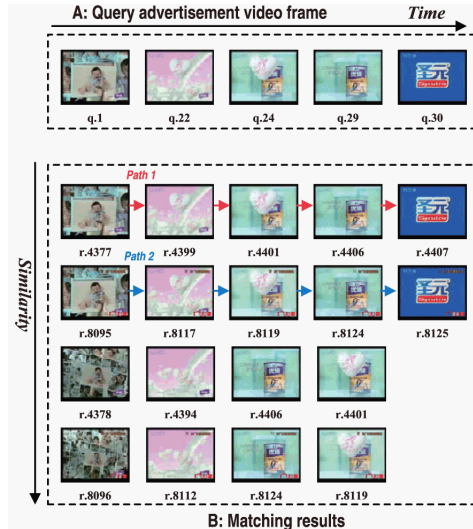
Indexation de la vidéo

Schéma de CBVR



Indexation de la vidéo

Schéma de CBVR



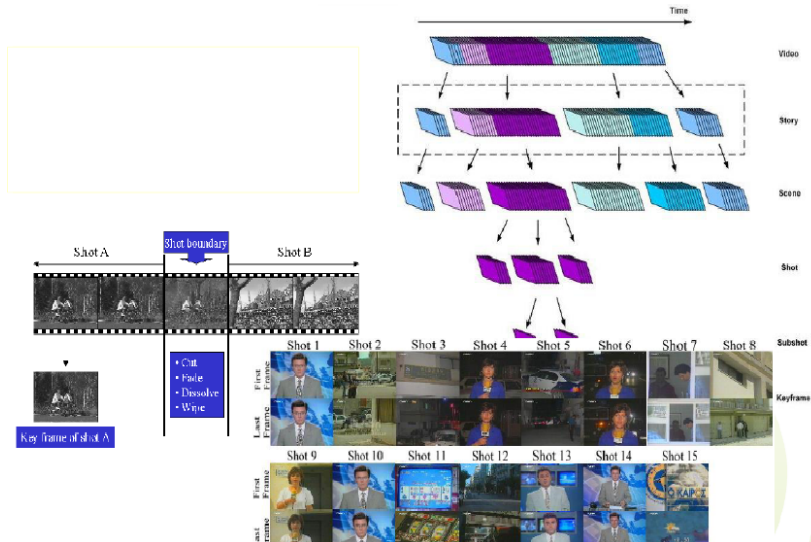
Analyse de la vidéo numérique

Segmentation en plan et en scène

- La segmentation en plans est l'identification automatique, par des méthodes informatiques, des bornes des plans dans une vidéo.
 - Cela consiste à repérer automatiquement les points de montage définis à l'origine par le réalisateur, en mesurant les discontinuités entre les images successives de la vidéo
 - Permettra ensuite la sélection d'images-clés
- Frame clé (key frame) : Image de la vidéo représentant grossièrement le contenu visuel du plan

Analyse de la vidéo numérique

Segmentation en plan et en scène



Segmentation en plan et en scène

Structure d'une vidéo - transitions entre plans

- Comment détecter les transitions entre plans, i.e. discontinuités dans le flux vidéo ?
- Comment détecter les discontinuités dans le flux vidéo ?
 - transition brusque : cut
 - transitions progressives entre plans
- La détection de rupture de plan est basée sur la segmentation (partitionnement) de la séquence vidéo en une succession de frames représentant une action spatiale et temporelle continue

Segmentation en plan et en scène

Structure d'une vidéo - transitions entre plans

Les images au voisinage d'une transition sont très différentes

- On cherche à repérer les discontinuités dans le flux vidéo

De façon générale, on effectue les étapes suivantes :

- 1 Extraire une signature (primitive globale) sur chaque image
- 2 On définit ensuite une distance (ou mesure de similarité) entre les signatures
- 3 Entre frame successifs, l'application de cette distance, sur l'ensemble du flux vidéo, produit un signal unidimensionnel
- 4 On cherche alors les variations extrêmes (pics du signal) qui correspondent aux instants de faible similarité

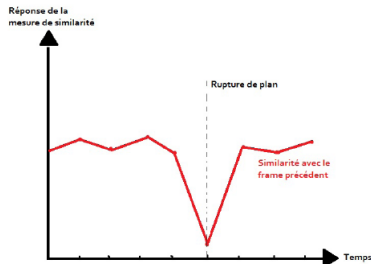
Segmentation en plan et en scène

Intensité

Imaginons la technique la plus simple :

- Signature : intensités des images.
- Mesure de similarité : distance moindres carrés :

$$D(t) = \sum_x \sum_y I(x, y, t) - I(x, y, t-1))^2$$



La technique est très sensible aux :

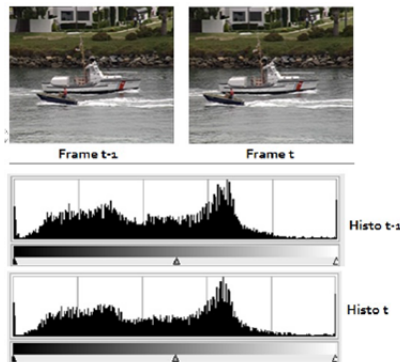
- Mouvements d'objets ou de caméra, bruit, ombrage, changement d'illumination

Il faut donc trouver des nouvelles primitives ou mesures plus robustes.

Segmentation en plan et en scène

Histogramme

- Un plan contient grossièrement les mêmes objets en mouvement sur un fond statique
- La distribution de l'intensité à l'intérieur des frames d'un même plan est donc semblable
- On représente cette distribution par l'histogramme d'intensité des frames



Segmentation en plan et en scène

Histogramme - Mesure de similarité

Comment calculer une mesure de similarité avec des histogrammes ?

- On calcul une distance entre les histogrammes normalisés de chaque deux frames successifs de la vidéo et on pose un seuil pour l'une des distances suivantes :
 - Distance euclidienne

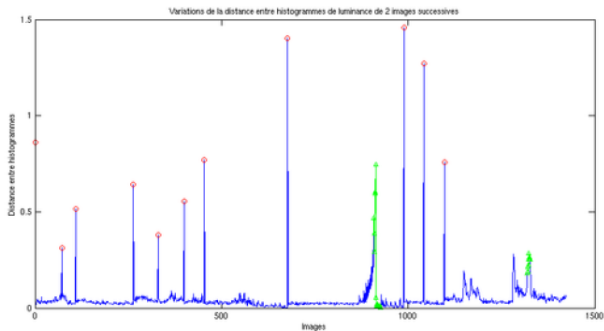
$$D_e(H_t, H_{t-1}) = \sum_{u=0}^m (H_t(u) - H_{t-1}(u))^2$$

- Intersection d'histogramme

$$D_i(H_t, H_{t-1}) = 1 - \frac{\sum_{u=0}^m \min(H_t(u), H_{t-1}(u))}{\min(\sum H_t(u), \sum H_{t-1}(u))}$$

Segmentation en plan et en scène

Histogramme



Avantage de l'histogramme d'intensité :

- Insensible au mouvement des objets : représentation qui ignore les arrangements spatiaux du contenu des frame
- Insensible aux petits mouvements de camera : la représentation est insensible aux secousses de caméra

Création de résumé

Pourquoi ?

On veut fournir des informations pertinentes et concises afin d'aider l'utilisateur à indexer, à naviguer ou à organiser des fichiers vidéos plus efficacement.

- Résumé statique (video summary) : sélectionner les images les plus représentatives de la vidéo
 - Ces images appelées images clés se présentent en général sous la forme d'un scénarimage (storyboard)
- Résumé dynamique (video skimming) : version courte de la vidéo originale
 - Ex : construction automatique d'une bande-annonce du film

Création de résumé

Méthodes

On peut regrouper les familles de résumés statiques en quatre catégories distinctes :

- méthodes reposant sur l'échantillonnage
- méthodes reposant sur les plans
- méthodes reposant sur les scènes
- ... autres méthodes : méthode avancée (sémantique, etc).

Création de résumé

Résumé statique par échantillonnage

- Dans le résumé statique par échantillonnage, il faut choisir les images clés en sous-échantillonnant uniformément ou aléatoirement la séquence originale.
- Inconvénient :
 - Certaines parties de la vidéo ne seront pas représentées
 - Redondance de certaines images clés avec un contenu similaire.
- Cette approche n'a pas d'attache au contenu de la vidéo.

Création de résumé

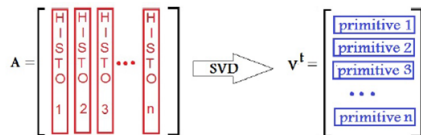
Résumé statique par plan

- Dans le résumé statique par plan, la détection des plans est réalisée pour mieux ajuster la sélection des images clés au contenu de la vidéo.
- Pour ce faire, on va extraire des images-clés des différents plans extraits de la vidéo.
- Cas 1 : Première image
 - On extrait la première image du plan comme image clé ou les première et dernière images du plan.
- Cas 2 : Utilisation de l'histogramme
 - La première image du plan est sélectionnée comme image clé. Puis, si la distance entre l'histogramme d'intensité de la dernière image clé sélectionnée et l'image courante est supérieure à un seuil, alors l'image courante est la nouvelle image clé.

Création de résumé

Résumé statique par scène

- L'idée est d'assigner chaque image à un groupe, puis à réunir les groupes les plus similaires de manière itérative jusqu'à un critère d'arrêt.
- Une matrice est créée où chaque colonne contient un vecteur caractéristique (l'histogramme) associé à un groupe d'images de la vidéo.



Une SVD est réalisée pour réduire l'espace des caractéristiques.

double invert(InputArray src, OutputArray dst, int flags=DECOMP_SVD), DECOMP_SVD singular value decomposition (SVD) method

Création de résumé

Résumé statique par scène

- On crée la matrice d'histogrammes A avec tous les frames de la scène
- On réduit A avec la SVD (singular value decomposition) pour avoir l'espace de primitive réduit V^t
- Partitionnement : On choisie une image-clé de départ, puis
 - On classe en ordre les vecteur-primitives (V_p) selon la distance avec le vecteur-primitive de l'image clé S_1
 - Pour chaque V_p non classé, on trouve l'image ayant la distance minimum $d(S_{min})$
 - Si $d(S_{min}) < T$, on associe le V_p à l'image S_{min} . Sinon, on crée une nouvelle image clé S_c
 - On arrête lorsque tous les V_p sont associés à une image clé S_c
- On retourne les images clés S_c

Création de résumé

Résumé statique par scène

Remarques

- On travaille sur la totalité des frames de la scène
- Le temps de calcul peut être long

Solution :

- Approche hiérarchique (séquentielle)

Définition de résumé hiérarchique à deux niveaux

- Premier niveau : on applique l'approche précédente avec une séquence fixe d'images (20 et plus)
- Deuxième niveau : on applique l'approche précédente, mais en utilisant les images clés sorties du niveau précédent

Cette approche permet de paralléliser la première étape et de réduire le coût de la décomposition SVD.

OpenCV

Modification temps réel d'un flux vidéo

- Si vous avez une caméra branchée sur votre ordinateur ou une webcam, tous les traitements d'OpenCV pour des images fixes peuvent également être appliqués sur la vidéo provenant de cette caméra, ou d'un fichier.
 - Ouvrir le flux d'une vidéo / webcam / caméra
 - Définissez premièrement la variable qui va stocker ce flux ou récupère les frames : **VideoCapture *vcapture;** ou **CvCapture *capture;**
 - Ouvrir une vidéo : **VideoCapture capture("video/Effidence.avi");//capture(0)** ou **capture = cvCreateFileCapture("/chemin/de/votre/vidéo/test.avi");** et **capture = cvCaptureFromCAM(int)**
 - Récupérer une image : par exemple, dans une boucle on récupère une image envoyée par la caméra en utilisant l'opérateur **vcapture >> frame** ou par la fonction **Mat cvQueryFrame (CvCapture *)**

OpenCV

Modification temps réel d'un flux vidéo

```
#include <opencv2/opencv.hpp>
#include <iostream>
using namespace cv;
using namespace std;

int main(){
    CvCapture* capture;
    Mat frame, img_gris;
    capture = cvCaptureFromCAM( -1 );
    if( !capture )
        return 1;
    namedWindow( "Video", CV_WINDOW_AUTOSIZE );
    while (1){
        frame = cvQueryFrame( capture );
        if( !frame.empty() ){
            // traitement de l'image
            cvtColor( frame, img_gris, CV_RGB2GRAY );
            inshow( "Video", img_gris );
        }
        // On attend 10 ms
        int key = cvWaitKey(10);
        // Si la touche 'échap' code ASCII '27' est appuyée
        // on quitte la boucle pour terminer le programme
        if( key == 27 ) break;
    }
    return 0;
}
```

OpenCV

Libérer la mémoire utilisée

Étape (très) importante, libérer toute la mémoire que vous avez utilisé en créant ces divers objets

- Libérer la fenêtre
 - Une seule fonction suffit à fermer toutes les fenêtres :
`cvDestroyAllWindows()`
 - Toutefois dans le cas où vous ne souhaiteriez que fermer une seule et unique fenêtre, vous pouvez utiliser cette dernière :
`cvDestroyWindow("nomFenetre")`
- Libérer la variable d'image : **`cvReleaseImage(unedImage)`**
- Libérer la capture vidéo : (entre autre pour que la webcam/caméra ne soit plus utilisée par le programme). **`cvReleaseCapture(capture);`**
- **Remarque** : lorsque vous récupérez une image d'un flux vidéo vous n'avez pas besoin de la « libérer » dans la boucle while, ni même après. OpenCV gère ceci automatiquement et vous n'avez qu'à votre charge de désallouer une image créée par vos soins