



Université Mohammed V
Faculté des Sciences
Rabat



Laboratoire de Recherche En Informatique et Télécommunication
(LRIT)

Master Recherche En Informatique et Télécommunication

Traitement de la vidéo (Partie 2)

Détection des changements de plans dans une vidéo en comparant différents histogrammes

Mini-Projet

réalise par :

CHETOUANI Yassin

AIT YACHOU Marwane

encadré par :

Pr. OUSGUINE Said

Année Universitaire 2019-2020

Table des Matières

INTRODUCTION GÉNÉRALE	3
1 INTRODUCTION AU TRAITEMENT ET L'ANALYSE DE VIDÉO	4
I. Introduction	4
II. Vidéo numérique et analogique	4
1. Vidéo Numérique	5
2. Sous-échantillonnage	6
III. Segmentation sémantique	6
IV. Conclusion	6
2 CHANGEMENT DE PLAN DANS UNE VIDEO	7
I. Introduction	7
II. Méthodes utilisé pour détecter un changement de plan.	7
1. méthode de différence pixel à pixel	7
2. méthode utilise une comparaison d'histogrammes	8
3. méthodes basées sur l'estimation de mouvement	8
III. Comparaison entre deux images en utilisent les histogrammes	8
1. Histogramme	8
2. Comparaison entre deux histogrammes	8
3. distance de Bhattacharyya	9
4. Conclusion	9
3 code et simulation sous Jupyter notebook	10
I. package	10
II. Transformer la video en plusieurs Frame	10
III. mettre les frames de video en ordre dans une liste	11
IV. Fonction qui construire l'histogramme de notre frames	11
V. Fonction qui normalise les histogrammes	11
VI. fonction qui calcule la distance de Bhattacharyya entre deux histogrammes	11
VII. fonction qui compare les histogrammes de deux images	12
VIII. fonction qui compare les histogrammes de deux frames succesives	12
IX. détection de changement du plan	12
WEBOGRAPHIE	13

Table des figures

1.1	<i>Image variant avec le temps</i>	4
1.2	<i>Vidéo Analogique</i>	5
1.3	<i>Segmentation sémantique</i>	6
2.1	<i>Histogramme d'une image de niveau de gris</i>	8

INTRODUCTION GÉNÉRALE

La vidéo regroupe l'ensemble des techniques permettant l'enregistrement ainsi que la restitution d'images animées, accompagnées ou non de son, sur un support électronique et non de type photochimique. Le mot « vidéo » vient du latin « vidéo » qui signifie : « je vois ». C'est l'apocope de vidéophonie ou vidéo-gramme. Le substantif vidéo s'accorde en nombre, cependant, l'adjectif reste toujours invariable. Selon les rectifications orthographiques du français en 1990, l'adjectif suit désormais la règle générale : l'adjectif s'accorde en nombre.

L'histoire du numérique dans la vidéo commence véritablement de 1972 à 1982. À l'origine équipements de synchronisation, les appareils se sophistiquèrent avant d'entrer dans le milieu professionnel. Dès lors, les industriels prirent conscience de l'avènement de ce nouveau phénomène et présentèrent des normes en matière de numérisation. Une certaine anarchie numérique régna alors sur le marché ce qui força la main au CCIR (Comité consultatif international des radiocommunications) à normaliser un format vidéo en composantes numériques compatible dans le monde entier : cette norme c'est le CCIR 601. Elle spécifie les paramètres de codage de signaux à numériser (échantillonnage, quantification...). Dès lors les innovations ne cessèrent de s'enchaîner pour permettre aujourd'hui à la vidéo numérique de se généraliser dans les centres de production, chaînes TV et régie de post-production pour assister le montage vidéo.

Chapitre 1

INTRODUCTION AU TRAITEMENT ET L'ANALYSE DE VIDÉO

I. Introduction

Une vidéo est une succession d'images à une certaine cadence. L'œil humain a comme caractéristique d'être capable de distinguer environ 20 images par seconde. Ainsi, en affichant plus de 20 images par seconde, il est possible de tromper l'œil et de lui faire croire à une image animée. On caractérise la fluidité d'une vidéo par le nombre d'images par secondes (en anglais frame rate), exprimé en FPS (Frames par second, en français trames par seconde).

D'autre part la vidéo au sens multimédia du terme est généralement accompagnée de son, c'est-à-dire de données audio.

la vidéo c'est une image variant avec le temps, sa ve dire en ajoute une 3 ème dimension à l'image, c'est la variable de temps, et qui possède des propriétés spéciale, un lien que l'on qualifiera de mouvement.



FIGURE 1.1: *Image variant avec le temps*

II. Vidéo numérique et analogique

On distingue généralement plusieurs grandes familles d'"images animées".

Le cinéma, consistant à stocker sur une pellicule la succession d'images en négatif. La restitution du film se fait alors grâce à une source lumineuse projetant, à partir d'une copie en positif, les images successives sur un écran.

subsection Vidéo Analogique

La vidéo analogique, représentant l'information comme un flux continu de données analogiques, destiné à être affichées sur un écran de télévision (basé sur le principe du balayage. Il existe plusieurs normes pour la vidéo analogique. Les trois principales sont :

PAL/SECAM (Phase Alternating Line/Séquentiel Couleur avec Mémoire) : utilisé en Europe pour la télévision hertzienne, permet de coder les vidéos sur 625 lignes (576 seulement sont affichées car 8 % des lignes servent à la synchronisation). à raison de 25 images par seconde à un format 4 :3 (c'est-à-dire que le rapport largeur sur hauteur vaut $4/3$).

Or à 25 images par seconde, de nombreuses personnes perçoivent un battement dans l'image. Ainsi, étant donné qu'il n'était pas possible d'envoyer plus d'informations en raison de la limitation de bande passante, il a été décidé d'entrelacer les images, c'est-à-dire d'envoyer en premier lieu les lignes paires, puis les lignes impaires. Le terme "champ" désigne ainsi la "demi-image" formée soit par les lignes paires, soit par les lignes impaires. L'ensemble constitué par deux champs est appelé trame entrelacé. Lorsqu'il n'y a pas d'entrelacement le terme de trame progressive est utilisé.



FIGURE 1.2: *Vidéo Analogique*

Grâce à ce procédé appelé "entrelacement", le téléviseur PAL/SECAM affiche 50 champs par seconde (à une fréquence de 50 Hz), soit 2×25 images en deux secondes.

NTSC : La norme NTSC (National Television Standards Committee), utilisée aux États-Unis et au Japon, utilise un système de 525 lignes entrelacées à 30 images/sec (donc à une fréquence de 60Hz). Comme dans le cas du PAL/SECAM, 8% des lignes servent à synchroniser le récepteur. Ainsi, étant donné que le NTSC affiche un format d'image 4 :3, la résolution réellement affichée est de 640x480.

1. Vidéo Numérique

La vidéo numérique consiste à afficher une succession d'images numériques. Puisqu'il s'agit d'images numériques affichées à une certaine cadence, il est possible de connaître le débit nécessaire pour l'affichage d'une vidéo, c'est-à-dire le nombre d'octets affichés (ou transférés) par unité de temps.

Ainsi le débit nécessaire pour afficher une vidéo (en octets par seconde) est égal à la taille d'une image que multiplie le nombre d'images par seconde.

La vidéo numérique caractérise par les paramètres suivants :

- f_t : Fréquence d'image (frames per seconds (fps)).
- f_y : Lignes par frame.
- f_x : Pixels par lignes.
- N_b : Bits à encoder.

Ce dernier paramètre encode la valeur d'intensité (couleur) d'un pixel.

2. Sous-échantillonnage

Étant donné que l'œil est peu sensible aux variations de chrominance, la technique dite de sous-échantillonnage en chrominance (en anglais chroma subsampling), appelée également décimation, consiste à supprimer des informations de chrominance dans un groupe de 4x4 pixels.

III. Segmentation sémantique

La segmentation est essentielle pour les tâches d'analyse d'image. La segmentation sémantique décrit le processus d'association de chaque pixel d'une image à une étiquette de classe (telle qu'une fleur, une personne, une route, un ciel, un océan ou une voiture).

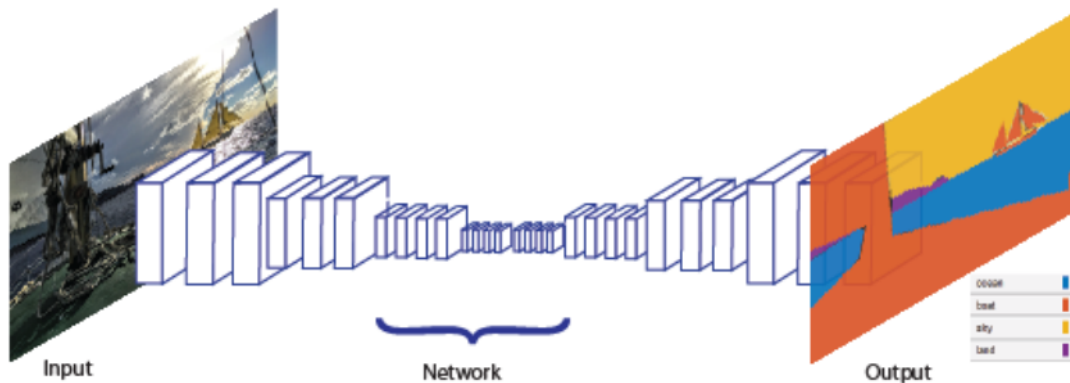


FIGURE 1.3: *Segmentation sémantique*

IV. Conclusion

Le traitement de la vidéo, permet d'approfondir le champ d'expertise de l'analyse et du traitement d'image en ajoutant la 3^{ème} dimension c'est la variable temporelle.

Chapitre 2

CHANGEMENT DE PLAN DANS UNE VIDEO

I. Introduction

L'émergence de la vidéo comme média d'information s'accompagne d'un besoin croissant d'outils pour segmenter ces séquences d'images dans le domaine temporel. Cette segmentation temporelle a pour but de détecter les changements de plan, généralement caractérisés par des cuts ou par d'autres effets plus complexes. Elle est par exemple nécessaire dans des applications telles que la gestion de bases de données multimédia, la création automatique de résumés de films ou de séquences télévisées, la compression de séquences vidéo, ou encore le suivi d'objet en temps-réel dans des scènes dynamiques.

Nous sommes particulièrement intéressés par ce dernier type d'application, c'est-à-dire le suivi d'objet en temps-réel. Une détection préalable des changements de plan est nécessaire pour assurer une bonne qualité de suivi et une exécution en temps-réel.

Après avoir rappelé les principales solutions existantes, nous allons présenter ici notre méthode de détection de changement de plan en détaillant les principales étapes de l'algorithme.

II. Méthodes utilisé pour détecter un changement de plan.

La plupart des méthodes de détection de changements de plan peuvent se regrouper en plusieurs catégories : différence pixel à pixel, comparaison d'histogrammes, estimation de mouvement. Ces catégories vont être présentées brièvement.

1. méthode de différence pixel à pixel

Les méthodes de différence pixel à pixel comparent les pixels de deux images successives. Si le nombre de pixels différents est supérieur à un seuil, alors un changement de plan est détecté. Ces méthodes sont assez sensibles au bruit et aux mouvements importants dans la scène.

2. méthode utilise une comparaison d'histogrammes

Le deuxième type de méthodes utilise une comparaison d'histogrammes : si les histogrammes de deux images successives sont suffisamment différents, alors on affirme qu'un changement de plan est présent dans la séquence. L'inconvénient de ces méthodes est lié à la non-prise en compte de l'information spatiale contenue dans les images.

3. méthodes basées sur l'estimation de mouvement

Enfin, les méthodes basées sur l'estimation de mouvement estiment le flux optique dans deux images successives. Une comparaison du nombre de mouvements différents entre les deux images à un seuil permet de déterminer la présence d'un changement de plan. Ces méthodes sont cependant coûteuses en temps de calcul.

III. Comparaison entre deux images en utilisant les histogrammes

On a plusieurs méthodes qui servent à comparer deux images et dire ce qui se ressemble et ce qui diffère, mais nous sommes intéressés d'avoir comparé deux images en utilisant la comparaison des histogrammes de ces deux images.

1. Histogramme

L'histogramme représente la distribution des intensités (ou des couleurs) de l'image. C'est un outil fondamental du traitement d'images, avec de très nombreuses applications. L'histogramme est défini comme une fonction discrète qui associe à chaque valeur d'intensité le nombre de pixels prenant cette valeur. La détermination de l'histogramme est donc réalisée en comptant le nombre de pixels pour chaque intensité de l'image.

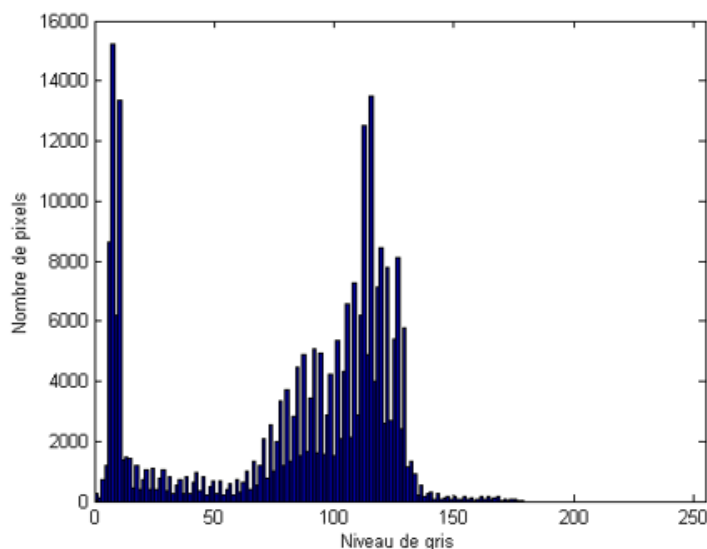


FIGURE 2.1: *Histogramme d'une image de niveau de gris*

2. Comparaison entre deux histogrammes

On va représenter chaque image en utilisant des histogrammes, en utilisant trois histogrammes 1D : un pour chaque RVB canal - il est sûr d'utiliser juste la couleur et d'ignorer la texture et les histogrammes de bord (une approche alternative utilise un histogramme 3D unique pour chaque image, mais je l'évite

car il ajoute une complexité supplémentaire). Par conséquent, On va comparer les histogrammes pour voir comment ils sont similaires, et si la mesure de similitude passe une certaine valeur de seuil alors On peut dire avec confiance que les images respectives sont visuellement les mêmes, On comparerions les histogrammes de canal correspondants de chaque image (par exemple, image L'histogramme rouge de l'image 1 avec l'histogramme rouge de l'image 2, puis l'histogramme bleu de l'image 1 avec l'histogramme bleu de l'image 2, puis les histogrammes verts, donc On ne compare pas l'histogramme rouge de l'image 1 avec l'histogramme bleu de l'image 2, ce serait juste idiot).

Pour Comparer deux histogramme est ce qu'il ressemble ou non On doit calculer la distance de Bhattacharyya.

3. distance de Bhattacharyya

La distance de Bhattacharyya est une mesure de la similarité de deux distributions de probabilités discrètes. Elle est reliée au coefficient de Bhattacharyya, qui est une mesure statistique du recouvrement de deux ensembles d'échantillons. Cette mesure est la plus utilisée pour la mise en correspondance entre deux observations basées sur l'histogramme de couleur. Cette mesure est régulièrement utilisée dans des problèmes de classification, en particulier dans le domaine de la vision par ordinateur.

Pour deux distributions de probabilité discrète p et q définies sur le même espace de probabilité, la distance de Bhattacharyya est calculée par :

$$D_B(p, q) = -\ln(BC(p, q))$$

Et le coefficient de Bhattacharyya c'est :

$$BC(p, q) = \sum \sqrt{p(x)q(x)}$$

Pour des distributions de probabilité continues, le coefficient est défini par :

$$BC(p, q) = \int \sqrt{p(x)q(x)} dx$$

Dans le deux cas, $0 \leq BC \leq 1$. La distance de Bhattacharyya n'obéit pas à l'inégalité triangulaire, au contraire de la distance de Hellinger, définie à partir de la distance de Bhattacharyya par $\sqrt{1 - BC}$.

4. Conclusion

Pour comparer deux images en utilisant la méthode de comparaison des histogrammes, On va tout d'abord construire les histogrammes de ces images, Pour une image RVB (histogramme qui représente le Rouge, histogrammes qui représente le Vert et histogrammes qui représentent le Blue), puis comparer les histogrammes des deux images qui y ont la même couleur sa-ve-dire (comparer l'histogramme rouge de l'image 1 avec l'histogramme rouge de l'image 2 etc...), on va calculer la distance de Bhattacharyya entre deux histogrammes qui représente la même couleur, puis additionner les distances des 3 couleurs et les diviser par 3 pour avoir la moyenne de comparaison, si la moyenne passe une certaine valeur de seuil alors On peut dire avec confiance que les deux images sont différentes, et pour détecter un changement de plan dans une vidéo il faut diviser la vidéo en plusieurs frames et comparer deux frames successive avec la méthode précédente si s'ils sont différents, donc automatiquement on a un changement de plan entre les deux frames.

Chapitre 3

code et simulation sous Jupyter notebook

I. package

```
from PIL import Image
from math import sqrt
from math import log
from math import fabs
import matplotlib.pyplot as plt
import cv2
import numpy as np
import os
```

II. Transformer la video en plusieurs Frame

```
# Lire la video dans un fichier
cap = cv2.VideoCapture('video_test2.mp4')

try:
    if not os.path.exists('data'):
        os.makedirs('data')
except OSError:
    print ('Error: Creating directory of data')
c=0
currentFrame = 0
currentFrame1= 0
currentFrame2= 0
while(True):
    # Capturer frame par frame
    ret, frame = cap.read()

    # Enregistre l'image du frame actuelle dans un fichier jpg
    name = './data/frame' + str(currentFrame) + str(currentFrame1) + str(currentFrame2) + '.jpg'
    print ('Creating...' + name)
    cv2.imwrite(name, frame)
    # Enregistre l'image du frame actuelle dans notre dossier pour lire les frames jpg
    name = './frame' + str(currentFrame) + str(currentFrame1) + str(currentFrame2) + '.jpg'
    print ('Creating...' + name)
    cv2.imwrite(name, frame)
    # Pour arrêter les images en double
    if currentFrame2 < 9:
        currentFrame2 += 1
    else:
        currentFrame2=0
        currentFrame1+=1
        c=1
    if currentFrame1 < 9 and c==1:
        c=0
    elif c==1:
        currentFrame1=0
        currentFrame+=1
        c=0
    # Lorsque tout est terminé, relâchez la capture
    cap.release()
    cv2.destroyAllWindows()
```

III. mettre les frames de video en ordre dans une liste

```
data=[]
for filename in os.listdir(f'data'):
    data.append(filename)
    # mettre les frames dans une liste
```

IV. Fonction qui construit l'histogramme de notre frames

```
def buildHistogram(im):
    pix = im.load()
    if im.mode == "RGB":
        R = [0]*256
        G = [0]*256
        B = [0]*256
        histo1=[]
        for i in range(0, im.size[0]):
            for j in range(0, im.size[1]):
                val = pix[i, j]
                R[val[0]] += 1
                G[val[1]] += 1
                B[val[2]] += 1
        histo = (R, G, B)
        return histo;
    elif im.mode == "HSV":
        H = [0]*360
        S = [0]*100
        V = [0]*100
        for i in range(0, im.size[0]):
            for j in range(0, im.size[1]):
                val = pix[i, j]
                H[val[0]] += 1
                S[val[1]] += 1
                V[val[2]] += 1
        histo = (H, S, V)
        return histo;
    else:
        print("Votre image n'est pas en RGB ou en HSV")
```

V. Fonction qui normalise les histogrammes

```
def normHisto(histo):
    r = []
    s = 0.0
    for h in histo:
        s += h
    for h in histo:
        r.append(h/s)
    return r
```

VI. fonction qui calcule la distance de Bhattacharyya entre deux histogrammes

```
def comparaisonHistoBhattacharyya(histoA, histoB):
    hA = normHisto(histoA) # normaliser l'histogramme A
    hB = normHisto(histoB) # normaliser l'histogramme B
    distance = 0.
    for i in range(0, 255):
        distance += (hA[i] * hB[i]) ** 0.5
    newDistance = -log(distance)
    # calculer la distance de Bhattacharyya
    return newDistance
```

VII. fonction qui compare les histogrammes de deux images

```
def comparaisonHisto3channels(histoA, histoB):
    resultsBatta = []
    moyenneBatta = 0
    for c in range(3):
        # le resultat du calcul de la distance de Bhattacharyya entre histoA[c] et histoB[c]
        resultsBatta.append(comparaisonHistoBhattacharyya(histoA[c], histoB[c]))
    for i in range(0, 3):
        moyenneBatta += resultsBatta[i]
    moyenneBatta /= 3. # calculer la moyenne des 3 distances de Bhattacharyya
    r=round((moyenneBatta * 100), 2)
    return r
```

VIII. fonction qui compare les histogrammes de deux frames successives

```
def execution():
    for i in range(len(data)):
        histo1=buildHistogram(Image.open(data[i])) #construire l'histogramme de l'image
        histo2=buildHistogram(Image.open(data[i+1])) #construire l'histogramme de l'image suivante
        compahisto=comparaisonHisto3channels(histo1, histo2) # comparer les histogrammes de deux images successives

        if compahisto >= 1:
            print("détection de changement du plan entre frame", i, "et frame", i+1)
        else:
            print("pas de changement de plan entre frame", i, "et frame", i+1)
    return 0
```

IX. détection de changement du plan

```
execution()
```

```
pas de changement de plan entre frame 0 et frame 1
pas de changement de plan entre frame 1 et frame 2
pas de changement de plan entre frame 2 et frame 3
pas de changement de plan entre frame 3 et frame 4
pas de changement de plan entre frame 4 et frame 5
pas de changement de plan entre frame 5 et frame 6
pas de changement de plan entre frame 6 et frame 7
pas de changement de plan entre frame 7 et frame 8
pas de changement de plan entre frame 8 et frame 9
pas de changement de plan entre frame 9 et frame 10
pas de changement de plan entre frame 10 et frame 11
pas de changement de plan entre frame 11 et frame 12
pas de changement de plan entre frame 12 et frame 13
détection de changement du plan entre frame 13 et frame 14
détection de changement du plan entre frame 14 et frame 15
détection de changement du plan entre frame 15 et frame 16
détection de changement du plan entre frame 16 et frame 17
détection de changement du plan entre frame 17 et frame 18
détection de changement du plan entre frame 18 et frame 19
détection de changement du plan entre frame 19 et frame 20
détection de changement du plan entre frame 20 et frame 21
détection de changement du plan entre frame 21 et frame 22
pas de changement de plan entre frame 22 et frame 23
pas de changement de plan entre frame 23 et frame 24
pas de changement de plan entre frame 24 et frame 25
pas de changement de plan entre frame 25 et frame 26
```

WEBOGRAPHIE

<https://paf.telecom-paris.fr/projets/detection-changements-plan-pour-sequencement-dune-video>

<https://forums.commentcamarche.net/forum/affich-6656681-comparer-deux-imagesanswers>

<https://perso.telecom-paristech.fr/gousseau/gretsi.pdf>

<https://people.irisa.fr/Sebastien.Lefevre/publis/forumIsis2000.pdf>

<https://tel.archives-ouvertes.fr/tel-00278073/document>

http://www.numdam.org/article/RSA_1993_414730.pdf

<https://github.com/romgille/Projet-traitement-d-image/blob/master/README.md>

<https://webdevdesigner.com/q/comparing-two-histograms-36520/>