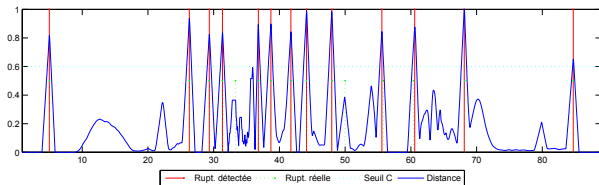


# DÉTECTION DE RUPTURE DANS UNE VIDÉO

Soutenance de projet d'EDTS

7 janvier 2015



Ya Huang   Rémi Mussard   Thomas Robert

Institut National des Sciences Appliquées de Rouen

# Sommaire

- 1 Extraction de caractéristiques
- 2 Détection de rupture
- 3 Conclusion
- 4 Bibliographie

Extraction de caractéristiques

# Différents types de rupture – Définition

## Définition

Il existe plusieurs types de rupture, les différents types de rupture influencent le choix de la taille de fenêtre, la précision du résultat, etc.

- Changement d'image ;
- Fondu d'image ;
- etc.

# Rupture – Exemples



Figure 1: Avant



Figure 2: Après



Figure 3: Avant



Figure 4: Après

# RGB – Définition

## Définition

La matrice RGB contient les informations des trois couleurs des frames d'une vidéo. L'information de chaque couleur est représenté par 256 niveaux.

La structure de cette matrice est en deux dimensions :

- Première dimension :
  - Taille :  $(3 \text{ couleurs}) \times (256 \text{ niveau})$  ;
  - Nombre d'occurrence de chaque niveau de couleur.
- Seconde dimension : Différents frames.

# RGB – Conclusion

## Avantages :

- Simple à mettre en œuvre ;
- Relativement robuste à l'apparition de nouveaux objets sur une scène.

## Inconvénients :

- Peu robuste aux transition avec un effet de fondu.

# YCbCr – Définition

## Définition

YCbCr est une manière de représenter l'espace colorimétrique. Y représente la densité de la lumière. Cb et Cr sont les deux informations de chrominance.  $Cb = (Y - \text{bleu})$ ,  $Cr = (Y - \text{rouge})$ .

- Première dimension :
  - Taille :  $(3 \text{ couleurs}) \times (256 \text{ niveau})$  ;
  - Nombre d'occurrence de chaque niveau de couleur.
- Seconde dimension : Différents frames.



# YCbCr – *Conversion RGB YCbCr*

## Conversion

$$\begin{cases} Y = 0.299 \times R + 0.587 \times G + 0.114 \times B \\ Cb = -0.1687 \times R - 0.3313 \times G + 0.5 \times B + 128 \\ Cr = 0.5 \times R - 0.4187 \times G - 0.0813 \times B + 128 \end{cases}$$

$$\begin{cases} R = Y + 1.402 \times (Cr - 128) \\ G = Y - 0.34414 \times (Cb - 128) - 0.71414 \times (Cr - 128) \\ B = Y + 1.772 \times (Cb - 128) \end{cases}$$

# YCbCr – Exemples

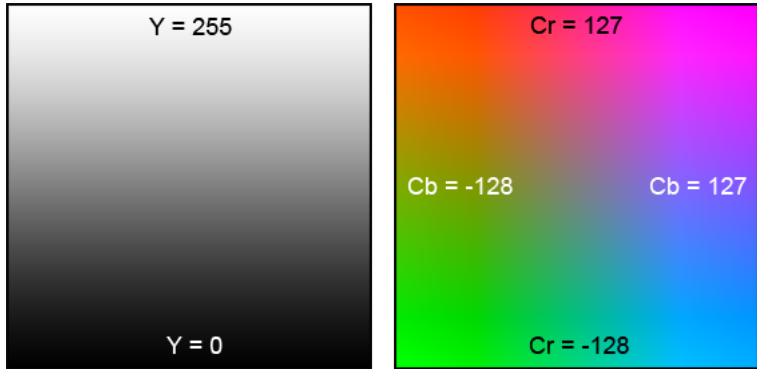


Figure 5: YCbCr

# YCbCr – Conclusion

## Avantages :

- Simple à mettre en œuvre, plus efficace que codage RGB ;
- Relativement robuste à l'apparition ou la disparition d'un objet sur une scène.

## Inconvénients :

- Peu robuste aux transitions avec un effet de fondu.

# Texture – Définition

## Définition

*Région spatiale d'une image présentant une organisation spatiale homogène des niveaux de luminance.*

Structure en deux dimensions :

- Première dimension : primitives formant la texture ;
- Seconde dimension : organisation spatiale de ces primitives entre elles.

# Texture – Types

## Types de texture :

- **Macrotecture** : texture structurée, pour laquelle on peut extraire facilement un motif de base et les lois d'assemblage des primitives entre elles ;
- **Microtexture** : texture aléatoire, avec un aspect désorganisé, mais qui donne une impression visuelle relativement homogène.

# Texture – Exemples



Figure 6: Exemple microtexture

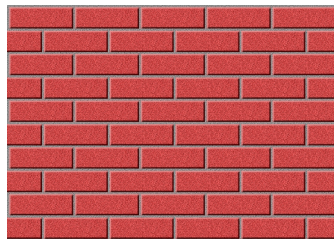


Figure 7: Exemple macrotexture

# Texture – *Matrice de cooccurrence*

## Principe :

Méthode statistique basée sur le niveau de gris d'une image (de 1 à 8 au maximum). Cette méthode compte le nombre de paires de pixels similaires dans une image et stocke les résultats dans une matrice de cooccurrence (taille  $n \times n$  avec  $n$  le niveau de gris maximum).

# Texture – Matrice de cooccurrence

## Exemple

$$I = \begin{pmatrix} 1 & 8 & 1 & 8 \\ 2 & 1 & 1 & 8 \\ 5 & 4 & 3 & 7 \end{pmatrix}, C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

avec  $I$  l'image en niveau de gris et  $C$  la matrice de cooccurrence.



# Texture – Conclusion

## Avantages :

- Simple à mettre en oeuvre ;
- Relativement robuste aux changements d'intensité d'une couleur ;
- Relativement robuste aux transitions avec un effet de fondu.

## Inconvénients :

- Peu robuste aux changement d'angle de caméra ;
- Peu robuste à l'apparition de nouveaux objets sur une scène.

# Flux optique – Définition

## Définition :

Le flux optique (ou flot optique) représente le mouvement apparent des objets présents sur une image et est basé sur la luminance.

Pour un pixel donné, la relation suivante est vérifiée entre deux images consécutives :

$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t)$  avec  $I$  la luminosité de l'image.

# Flux optique – *Ecriture différentielle*

## Equation du flux optique :

En supposant que le mouvement petit et constant, on obtient :

$$\frac{dl(x,y,t)}{\Delta t} = \frac{\partial l(x,y,t)}{\partial x} \cdot \frac{\Delta x}{\Delta t} + \frac{\partial l(x,y,t)}{\partial y} \cdot \frac{\Delta y}{\Delta t} + \frac{\partial l(x,y,t)}{\partial t}$$

Soit :

$$I_x \cdot V_x + I_y \cdot V_y = -I_t$$

avec  $V_x$  et  $V_y$ , le flux optique (ou les composants  $(x, y)$  de la vitesse) et  $I_x$ ,  $I_y$  et  $I_t$  les dérivées partielles de l'intensité de l'image.

# Flux optique – *Gradient et filtre de Sobel*

## Principe :

- Définition des matrices de convolution :  $C_1 = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}$  et

$$C_2 = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

- Calcul des dérivées partielles ( $G_x$  et  $G_y$ ) :  $G_x = C_1 * \text{Img}$  et  $G_y = C_2 * \text{Img}$
- Combinaison des gradients horizontaux et verticaux :

$$G = \sqrt{G_x^2 + G_y^2}$$

# Flux optique – *Méthode de Lucas-Kanade*

## Principe :

On suppose que le déplacement d'un pixel  $p$  est petit et constant dans son voisinage :

- Sélectionner une fenêtre centrée sur le pixel  $p$  dont on cherche la vitesse ;
- Réécrire l'équation du flux optique comme un système d'équation ;
- Résoudre ce système d'équations par la méthode des moindres carrés.

# Flux optique – *Conclusion*

## Avantages :

- Relativement robuste aux mouvements ;
- Relativement robuste aux changements d'intensité de couleur.

## Inconvénients :

- Peu robuste aux transitions avec un effet de fondu ;
- Peu robuste lors de l'arrivée ou de la disparition d'un objet dans la scène.

# Détection de rupture

# Introduction

## Données

- Matrice de signal  $X \in \mathbb{R}^{n \times p}$
- $n$  observations dans le temps, 1 par image
- $p$  caractéristiques extraites des images

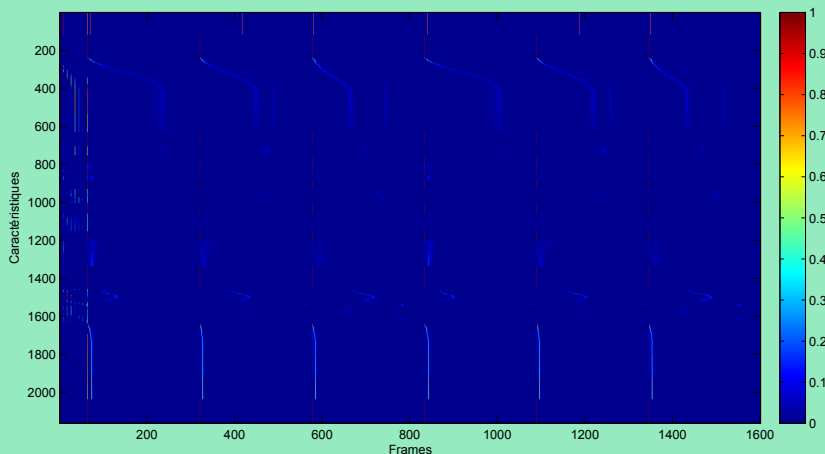
## Objectif

Déterminer les ruptures dans la vidéo au travers des caractéristiques extraites, dont la distribution est supposée stationnaire pour une séquence donnée.



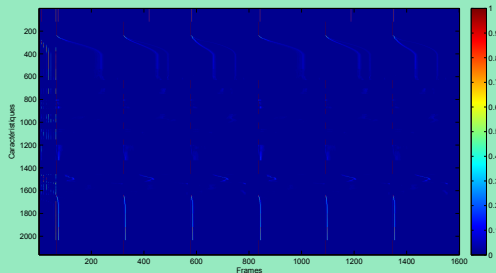
# Prétraitement des caractéristiques

## Caractéristiques obtenues



# Prétraitement des caractéristiques

## Caractéristiques obtenues



## Remarque

Peu de valeurs éloignées de 0

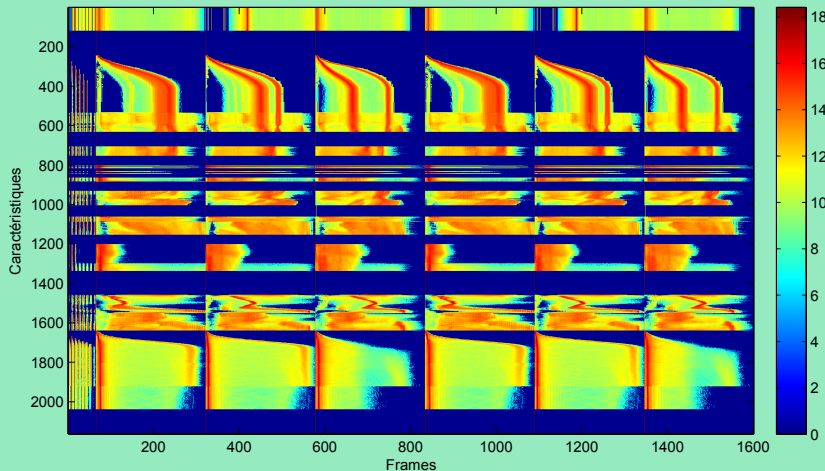
⇒ Prétraitement de la matrice de caractéristiques

# Prétraitement des caractéristiques – *Résultat*

**INSA**

INSTITUT NATIONAL  
DES SCIENCES  
APPLIQUÉES  
ROUEN

## Caractéristiques après prétraitement ( $\log(10^9 X + 1)$ )

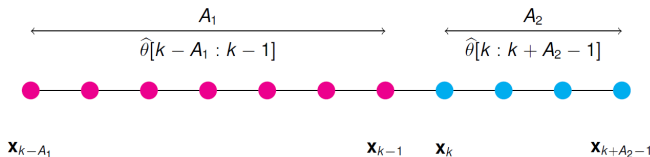


# Méthodes essayées

## Méthodes essayées

- Méthode par dérivée filtrée utilisant la  $p$ -valeur  
*Source : [Bertrand11] et [Herault14]*
- Méthode de détection de changement de noyau  
*Source : [Desobry05]*

# Dérivée filtrée – *Principe*



## Principe

- Parcours du signal avec deux fenêtré de taille  $A_1$  et  $A_2$
- Distance  $D(k)$  entre indicateurs  $\hat{\theta}$  sur chaque fenêtré
- Zones de rupture  $\mathcal{K}_i = \{k \text{ successifs t.q. } |D(k)| > C\}$
- Rupture en  $k_{r_i} = \arg \max_{k \in \mathcal{K}_i} |D(k)| \quad \forall i$

# Dérivée filtrée – Détermination de $C$

## Principe probabiliste basée sur la $p$ -valeur

- Hypothèse  $\mathcal{H}_0$  : il n'y a pas de rupture
- $M = \max_k |D(k)|$
- On veut  $C$  tel que  $\mathbb{P}(M > C \mid \mathcal{H}_0) < p$  avec  $p$  fixé

## Méthode d'estimation de $C$

- Création de  $\{X_{\mathcal{H}_0}\} = \{\text{randperm}(X)\}$  données sous  $\mathcal{H}_0$
- Estimation de  $\{M_{\mathcal{H}_0}\}$  sur ces données
- Estimation de la distribution de  $M$  sous  $\mathcal{H}_0$
- Estimation de  $C$  tel que  $\mathbb{P}(M > C \mid \mathcal{H}_0) < p$

# Changement de noyau – Principe

## Principe

- Cas particulier de la dérivée filtrée
- Indicateurs  $\hat{\theta}$  étant les paramètres d'un *one-class SVM* sur chaque fenêtre
- Mesure de distance particulière entre les kernels

# Changement de noyau – Distance

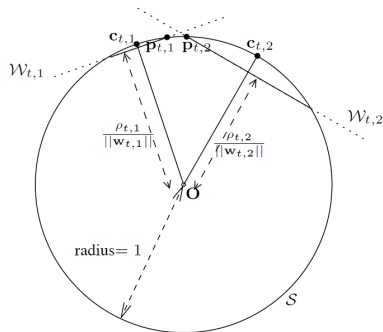
$$D = \frac{\widehat{c_1 c_2}}{\widehat{c_1 p_1} + \widehat{c_2 p_2}}$$

$$\widehat{c_1 c_2} = \arccos \left( \frac{\langle w_1, w_2 \rangle_{\mathcal{H}}}{\|w_1\|_{\mathcal{H}} \|w_2\|_{\mathcal{H}}} \right)$$

$$= \arccos \left( \frac{\alpha_1^\top K_{12} \alpha_2}{\sqrt{\alpha_1^\top K_{11} \alpha_1} \sqrt{\alpha_2^\top K_{22} \alpha_2}} \right)$$

$$\widehat{c_i p_i} = \arccos \left( \frac{\rho_i}{\sqrt{\alpha_i^\top K_{ii} \alpha_i}} \right)$$

$i, j = 1$  pour les données avant  $k$   
et  $2$  pour les données après.



- $\alpha_j$  : variables duales
- $w_i$  : poids de l'hyperplan
- $\rho_i$  : biais de l'hyperplan
- $K_{ij}$  : matrice de Gram



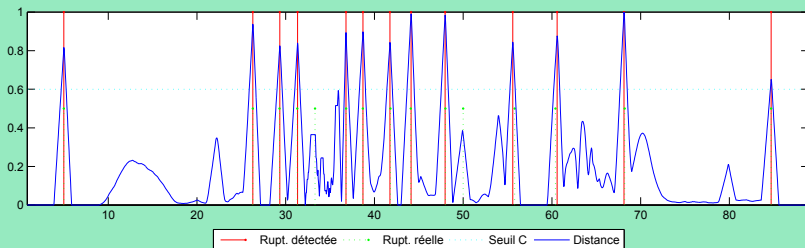
# Détermination des paramètres $A_i, C$

## Détermination des paramètres $A_i, C$

- Détermination de  $A_1$  en essayant différentes valeurs et en visualisant  $D$  ( $A_2 = A_1$ ).
- Détermination de  $A_2$  en essayant différentes valeurs et en visualisant  $D$ .
- Détermination de  $C$  visuellement.  
*Très mauvais résultats avec la méthode de la p-valeur avec 200 tirages aléatoires.*

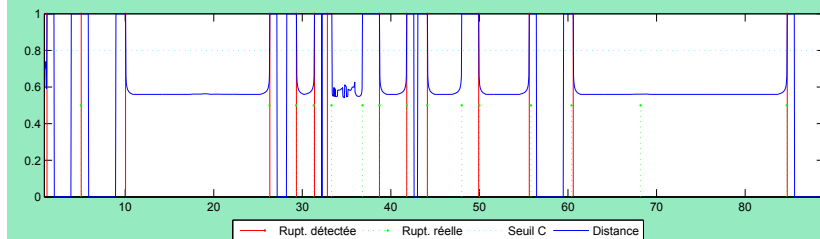
# Résultat final – *Différence de moyennes*

Distance et ruptures détectées,  $A_1 = 20, A_2 = 28, C = 0,6$



# Résultat final – *Distance one-class SVM*

Distance et ruptures détectées,  $A_1 = 20$ ,  $A_2 = 28$ ,  $C = 0,8$



# Conclusion

# Conclusion

- Extraction de diverses caractéristiques :
  - Histogrammes RGB & YCbCr
  - Matrices de cooccurrence
  - Gradient de l'image
- Détection de rupture par dérivée filtrée avec distances :
  - Différence des moyennes
  - Changement de kernel
- Résultats relativement bons mais pas parfaits
- Axes d'améliorations :
  - Ajout de caractéristiques
  - Autre méthode de détection ?

# Bibliographie

# Bibliographie I



Pierre Raphael Bertrand, Mehdi Fhima, and Arnaud Guillin.  
Off-line detection of multiple change points by the filtered  
derivative with p-value method.

*Sequential Analysis*, 30(2) :172–207, Avril 2011.



Frédéric Desobry, Manuel Davy, and Christian Doncarli.  
An online kernel change detection algorithm.

*Signal Processing, IEEE Transactions on*, 53(8) :2961–2974,  
2005.



Romain Hérault.

Cours « Dérivée filtrée par la méthode p-valeur », INSA  
Rouen, Décembre 2014.