



POLYTECHNIQUE MONTRÉAL

INF6803-TRAITEMENTS VIDÉOS ET APPLICATIONS

---

## TP1 - Segmentation par soustraction d'arrière-plan

---

*Auteurs:*

Desmet Laurent 1865564,  
Zins Pierre 1863527

Janvier, 2017

## Contents

<b>1</b>	<b>Présentation de la méthode VIBE et implémentation</b>	<b>2</b>
1.1	Présentation VIBE . . . . .	2
1.2	Implémentation . . . . .	2
<b>2</b>	<b>Résultats</b>	<b>3</b>

# 1 Présentation de la méthode VIBE et implémentation

## 1.1 Présentation VIBE

L'objectif de ce TP est de nous familiariser avec l'implémentation de méthodes de segmentation par soustraction d'arrière-plan. Il existe deux sous-familles :

- Les méthodes paramétriques, qui s'appuient sur des distributions de probabilités (moins de mémoire nécessaire)
- Les méthodes non paramétriques, qui s'appuient sur l'histogramme complet des pixels (plus de mémoire nécessaire). ViBe fait plutôt parti de cette catégorie.

Dans le soucis de limiter la mémoire nécessaire à l'implémentation de ces méthodes, ViBe, permet de ne garder qu'un certain nombre de pixels pour constituer son arrière-plan (typiquement 20). Cette quantité est considérée comme suffisamment élevée pour considérer l'arrière-plan robuste et fiable. L'idée générale de cette méthode est donc la suivante :

- Pour une nouvelle trame  $t$  quelconque, un pixel  $(i,j)$  quelconque, on calcule la distance euclidienne entre ce pixel et tous les éléments individuels qui appartiennent à la distribution d'arrière-plans pour la position de ce même pixel que l'on appellera dans la suite  $\Omega$ .
- Nous avons donc 20 distances. Si au moins deux distances sont en-dessous d'un seuil que l'utilisateur a défini (généralement ce dernier vaut 20), alors on considère que le pixel appartient à l'arrière-plan.
- Sinon, on le classe comme avant-plan.

Pour pouvoir adapter l'arrière-plan au changement de luminosité progressif lors de la journée ou pour tout autre raison, l'utilisateur peut ajouter le pixel de la trame  $t$  dans  $\Omega$  si ce dernier a été considéré comme arrière-plan selon une certaine probabilité indépendante des pixels voisins et des trames passés (généralement 1/16). En fonction de la vitesse d'adaptation de l'arrière-plan, l'utilisateur peut décider d'augmenter ou non cette probabilité. Plus cette dernière augmente, plus la vitesse d'adaptation de l'arrière-plan est rapide.

Enfin, on peut également mettre à jour le modèle des voisins, en ajoutant le pixel courant à  $\Omega$  de l'un de ses voisins choisi aléatoirement, lorsque le pixel est d'arrière-plan. A nouveau, cela se fait suivant la probabilité du processus de mise à jour (en général 1/16).

## 1.2 Implémentation

Notre implémentation se base principalement sur les 2 méthodes données dans le code de départ *initialize()* et *apply()*. Nous avons ajouté 2 structures de données *intensity* et *descriptors* qui vont stocker pour chaque pixel de l'image plusieurs "samples", parmi les pixels à la même position dans les trames précédentes. *intensity* va contenir des intensités (RGB) des pixels, et *descriptors* contiendra la valeur d'un descripteur LBP calculée sur les 8 voisins directs d'un pixel.

Voici une description rapide des deux méthodes principales de notre implémentation.

**ViBe\_impl::initialize()** Dans cette méthode, nous avons initialisé le modèle d'arrière-plan d'intensité, avec  $m_N$  valeurs aléatoires parmi les voisins directs de chaque pixel. Pour l'initialisation du modèle

d'arrière-plan des descripteurs, nous avons simplement calculé la valeur du descripteur LBP du pixel et l'avons ajouté  $m\_N$  fois dans le modèle. En général  $m\_N$  vaut 20.

**ViBe\_impl::apply** Dans cette méthode, nous parcourons chaque pixel de la trame. Pour chacun d'entre eux, nous allons vérifier l'intensité à partir du modèle d'arrière-plan des intensités (*checkIntensity()*). On calcul la distance L2 entre l'intensité du pixel courant et les intensités de tous les pixels du modèle d'arrière-plan. Si la distance est inférieure à  $m\_R$  pour au moins  $m\_nMin$  pixels alors le pixel appartient à l'arrière-plan. Sinon, il sera d'avant-plan. En général,  $m\_nMin$  et  $m\_R$  valent respectivement 2 et 20.

Dans le premier cas, il faut également mettre à jour le modèle d'arrière-plan du pixel courant et celui d'un des voisins du pixel courant, choisi aléatoirement. La mise à jour se fait toujours avec la probabilité " $1/m\_nSigma$ " (en général 1/16).

**Améliorations** Nous avons proposé trois améliorations possibles :

- Appliquer un filtre médian, à la fin, une fois la méthode ViBe appliquée. Cela va réduire fortement le bruit et améliorer la performance globale de notre méthode.
- Appliquer des opérations morphologiques, dans le même but que le filtre médian. Nous effectuons une opération d'ouverture : érosion pour éliminer le bruit "blanc", puis dilatation pour éliminer le bruit "noir". Cependant, cette méthode, donne de moins bon résultats que l'application du filtre médian.
- Utiliser les descripteurs LBP et un test dessus pour la classification du pixel. Pour cela, nous calculons la valeur du descripteur LBP du pixel courant à partir de ses pixels voisins. Puis, comme avec les intensités, nous comparons cette valeur à celles présentes dans le modèle d'arrière-plan. A la place de la distance L2, nous utilisons cette fois la distance de Hamming. Si cette dernière est inférieure ou égale à un seuil (en général 3), les valeurs des 2 descripteurs sont considérées comme proches et leur valeur est prise en compte dans le calcul du descripteur. Ensuite, si la valeur du descripteur est proche d'au moins  $m\_nMin$  descripteurs du modèle d'arrière-plan, alors le pixel peut être considéré comme d'arrière-plan. Au niveau de la mise à jour du modèle d'arrière-plan des descripteurs, nous avons fait la même chose que pour les intensités, c'est à dire, la valeur du descripteur est ajoutée au modèle (à la place d'une autre choisie aléatoirement) avec une probabilité de " $1/m\_nSigma$ " (en général 1/16).

## 2 Résultats

Pour chacune des vidéos, nous avons calculé les métriques suivantes : precision, recall et f-measure.

- **Precision** : Dans notre cas, cela représente le rapport entre le nombre de pixel avant-plan correctement détectés par rapport au nombre total de pixels détectés comme avant-plan par notre méthode. On a donc un indice sur le nombre de pixels en réalité d'arrière-plan détectés comme avant-plan par notre méthode.
- **Recall** : Dans notre cas, cela représente le rapport entre le nombre de pixels avant-plan détectés correctement par notre méthode par rapport au nombre total de pixel d'avant-plan du groundtruth. Cela nous donne donc un indice sur la performance de détection de l'avant-plan pour notre méthode : combien de pixels d'avant-plan avons-nous bien détecté par rapport au vrai nombre de pixels d'avant-plan.

- **F-Measure** : Elle correspond plus ou moins à une moyenne pondérée de la Precision et de Recall. Elle donne une idée globale de la performance de notre méthode.

Pour chacune de ces trois mesures, leur valeur est comprise entre 0 et 1, et plus elle est proche de 1, meilleure est la performance.

	Highway		
	Precision	Recall	F-Measure
ViBe normal	0.820	0.916	0.865
Filtre médian	0.935	0.966	0.950
Op. morpho.	0.901	0.979	0.938
LBP descriptors	0.886	0.994	0.937

	Pedestrian		
	Precision	Recall	F-Measure
ViBe normal	0.669	0.969	0.791
Filtre médian	0.925	0.980	0.952
Op. morpho.	0.781	0.993	0.874
LBP descriptors	0.749	0.996	0.855

	Fountain		
	Precision	Recall	F-Measure
ViBe normal	0.116	0.919	0.205
Filtre médian	0.650	0.946	0.770
Op. morpho.	0.294	0.963	0.450
LBP descriptors	0.401	0.973	0.568

A partir de toutes ces mesures, on peut remarquer que ViBe normal donne de bons résultats pour les deux premières vidéos, mais un résultat plutôt mauvais (sauf pour "Recall") pour la vidéo *fountain*. Pendant l'exécution, on remarque que cela est principalement dû au jet d'eau qui va provoquer une classification erronée d'un grand nombre de pixels en avant-plan. Ainsi, la précision va fortement baisser, ce qui réduira "F-Measure". Au contraire, "Recall" reste plutôt bon, puisque l'on classe comme avant-plan la plupart des vrais pixels d'avant-plan de groundtruth.

Au niveau des améliorations apportées, le filtre médian propose l'amélioration la plus performante. Les opérations morphologiques présentent elles aussi de bons résultats, légèrement moins bon que le filtre médian.

L'ajout des descripteurs LBP n'améliore pas la méthode. Seul "Recall" est légèrement amélioré, mais la précision baisse fortement. Cela s'explique par le fait que l'on ajoute une condition à vérifier afin de classer un pixel comme "arrière-plan". Ainsi, les pixels auront plus tendance à être classifiés en "avant-plan" et donc une plus grosse proportion de pixel en vérité "avant-plan" seront détectés, d'où l'amélioration de "Recall". Dans le même temps, la précision est affectée car désormais on détecte trop de pixel d'avant-plan. Théoriquement, les descripteurs LBP devraient toutefois permettre d'améliorer la méthode ViBe. Grâce à eux, il est possible de tenir compte des textures. Notre implémentation est légèrement différente de celle proposée dans l'article [1], ce qui explique peut-être nos résultats.

Au niveau des paramètres de notre méthode, voici les valeurs de bases que nous avons utilisées.

- nombre de pixels dans le modèle d'arrière-plan : 20
- distance entre les couleurs pour matching : 20

- nombre minimum de matches pour une classification en tant qu'arrière-plan : 2
- probabilité de mise à jour du modèle d'arrière-plan : 1/16

Concernant le nombre de matches nécessaires pour la classification en arrière-plan, le fait de l'augmenter aura tendance à classer plus de pixels en avant-plan. D'après les métriques obtenues (pour une valeur de "5"), nous remarquons que la précision de notre méthode baisse fortement. Nous avons aussi essayé avec une valeur de "1" qui va favoriser une classification en arrière-plan. Dans ce cas, pour les 2 premières vidéos, Recall et F-Measure diminuent, ce qui est logique. Au contraire, dans la vidéo "fountain", nous obtenons d'excellentes valeurs (Precision = 0.970, Recall = 0.914, F-Measure = 0.914). Ceci s'explique par le fait que le nombre de pixels injustement classifiés en avant-plan seront désormais en arrière-plan. Ce paramètre doit donc être adapté en fonction de la vidéo, afin d'obtenir les meilleurs résultats.

Au niveau de la distance entre les couleurs pour le matching :

- réduire la valeur favorise une classification en avant-plan, et fait baisser les performances
- augmenter la valeur va favoriser la classification en arrière-plan. Nous avons essayé avec une valeur de 50. Comme précédemment, les résultats sont moins bons pour les 2 premières vidéos, mais bien meilleurs pour la 3ème vidéo (Precision = 0.997, Recall = 0.712, F-Measure = 0.831)

Pour le nombre de pixels à considérer dans le modèle d'arrière-plan, nous avons remarqué qu'augmenter cette valeur n'améliore aucune des mesures de manière significative. Seul le temps de traitement augmente ce qui n'est pas un point positif.

Enfin, pour la probabilité de la mise à jour du modèle, nous avons essayé avec une probabilité plus élevée (1/2). Dans ce cas, nous obtenons d'excellents résultats pour les 3 vidéos (toutes les métriques proches de 0.95). La plus grosse amélioration concerne la troisième vidéo dans laquelle on retrouve le jet-d'eau. Il est donc compréhensible qu'une mise à jour plus fréquente donne de meilleurs les résultats, puisque notre modèle d'arrière plan va s'adapter plus vites aux changements. Inversement, réduire la probabilité de mise à jour fait baisser les performances globales de notre méthode.

Finalement, voici les métriques pour notre méthode avec la meilleure amélioration (filtre médian) et les meilleures valeurs des paramètres.

	Precision	Recall	F-Measure
Highway	0.936	0.965	0.950
Pedestrian	0.952	0.978	0.965
Fountain	0.943	0.935	0.939

## Références

- [1] Pierre-Luc St-Charles Guillaume-Alexandre Bilodeau. Improving background subtraction using local binary similarity patterns.