Q1 Calculer et afficher pour chaque image de la vidéo, sous la forme d'une image, l'histogramme 2d correspondant à la probabilité jointe des composantes chromatiques (u, v) du codage Y uv des images couleur

Dans cette tâche, nous avons calculé l'histogramme 2D correspondant à la probabilité jointe des composantes chromatiques (u, v) du codage Yuv des images couleur pour chaque image de la vidéo. Pour ce faire, nous avons utilisé la fonction « calcHist2D » de la bibliothèque OpenCV pour calculer l'histogramme 2D et la fonction « imshow » pour l'afficher sous forme d'image (figure 1).

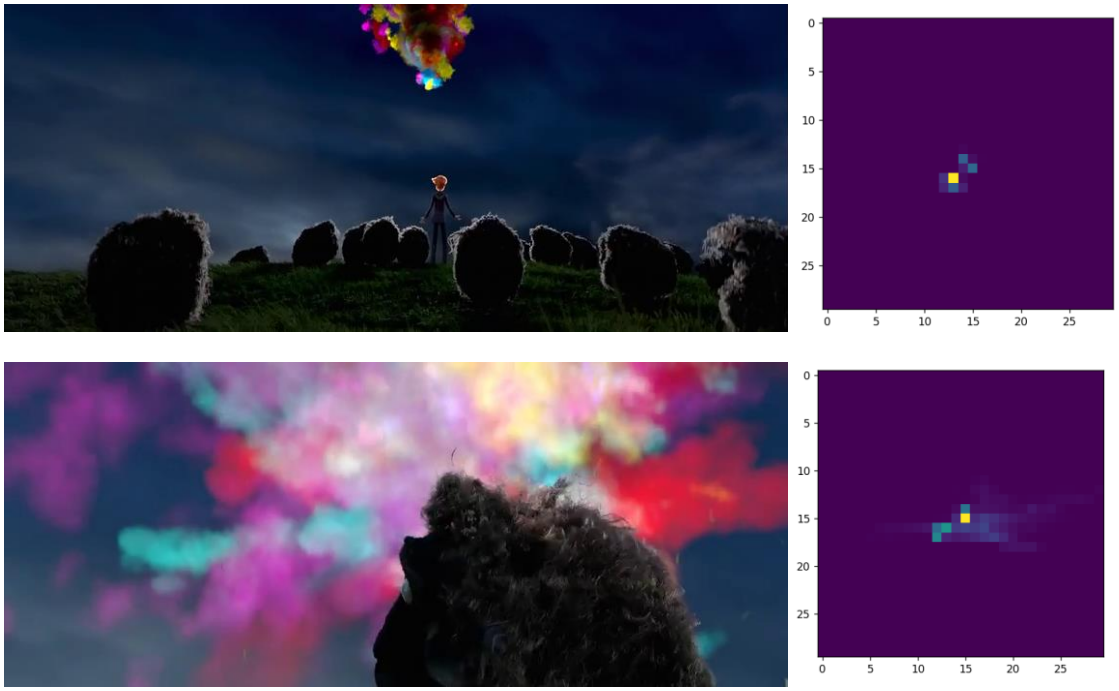


Fig 1. L'histogramme 2d correspondant à 2 images de la vidéo

Nous avons remarqué que le logarithme de l'histogramme est bien mieux affiché visuellement (figure 2).

Fig 2. Logarithme de l'histogramme 2d correspondant à l'images de la vidéo

Observer les variations de l'histogramme au cours d'une séquence et proposer des premières mesures pour la détection des changements de plan

Nous utilisons « cv2.compareHist » avec d'une fonction de distance «cv2.HISTCMP_BHATTACHARYYA ». En observant les variations de l'histogramme au cours de la séquence, nous avons remarqué que l'histogramme peut changer de manière significative lorsque le plan de la scène change (figure 3, figure 4). Nous avons donc proposé des premières mesures pour la détection des changements de plan, qui consistent à comparer l'histogramme courant avec l'histogramme précédent et à détecter les changements significatifs.

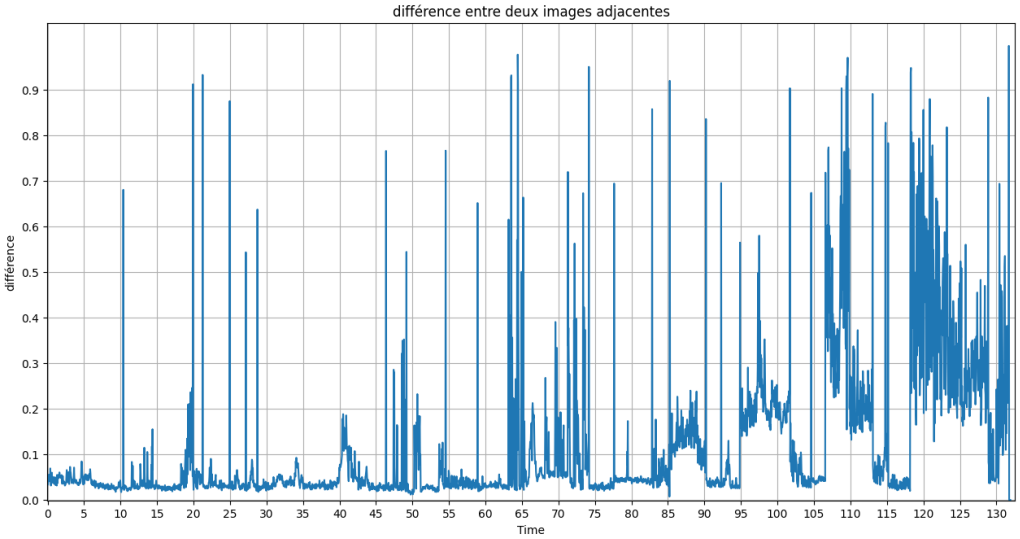
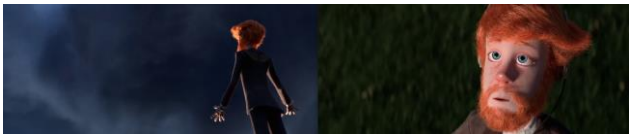


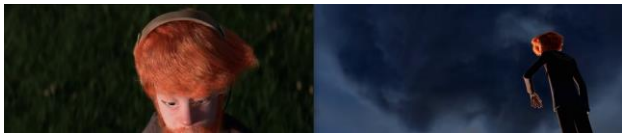
Fig 4. les variations de l'histogramme au cours d'une séquence

Nous avons également noté que cette méthode peut être sensible aux variations d'éclairage, qui peuvent modifier l'apparence de l'histogramme même si le plan de la scène ne change pas (figure 5). Pour résoudre ce problème, nous pourrions utiliser des techniques plus avancées de détection de changements de plan qui prennent en compte les variations d'éclairage.

variations = 0.6806764045080363

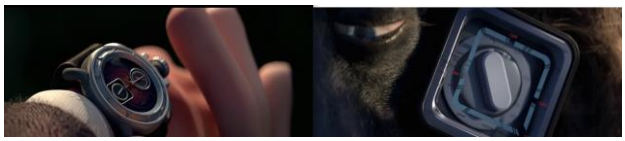


variations = 0.9122599744021581

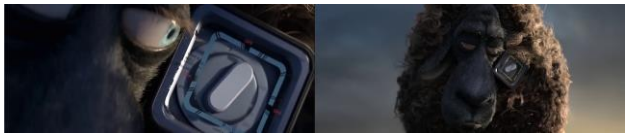


variations = 0.9327344897553256

variations = 0.8752107080348277

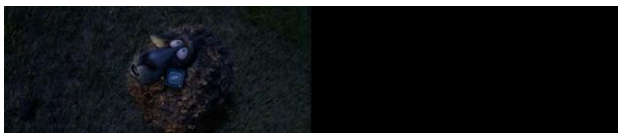


variations = 0.5435366459535745



variations = 0.6373721595473452

Fig 4. Les bons exemples de variations de l'histogramme au cours d'une séquence de plan

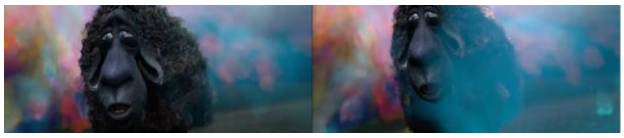


variations = 0.9269246371830758

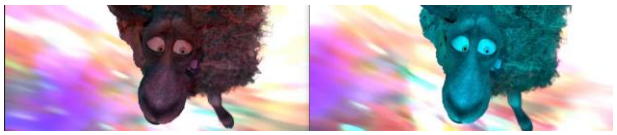
variations = 0.9318081611443215

variations = 0.5701944746808181

variations = 0.5627610816274995



variations = 0.48614587119861125



variations = 0.7668806801735438

Fig 5. Les mauvais exemples de variations de l'histogramme au cours d'une séquence de plan

Que faire dans le cas de vidéos monochromes ?

Dans le cas de vidéos monochromes, c'est-à-dire des vidéos en niveaux de gris où chaque pixel a une seule valeur de luminance, il n'est pas possible de calculer un histogramme 2D de composantes chromatiques (u, v) comme dans le cas de vidéos en couleur.

Cependant, il est toujours possible de détecter des changements de plan en utilisant d'autres mesures telles que la différence de luminance entre deux images consécutives ou la différence de gradients entre deux images consécutives. Ces mesures peuvent être utilisées pour calculer une carte de différence entre deux images consécutives et détecter des régions où la différence est supérieure à un seuil donné.

Il est également possible d'utiliser des techniques de traitement d'images telles que la segmentation ou la détection de contours pour extraire des informations à partir de vidéos monochromes et détecter des changements de plan. Nous pouvons donc utiliser ces méthodes pour détecter les changements de plan dans les vidéos monochromes.

3 Flot optique et histogramme de vitesses

Q2 Analyser le comportement et les arguments de la fonction de calcul du ot optique dense dans le script Dense-Optical-Flow.py, qui utilise la méthode de Farnebäck [1]. Expliquer brièvement le principe de cet algorithme, en particulier expliquer comment il peut calculer un ot dense, et comment il peut estimer de grandes vitesses.

Le script Dense-Optical-Flow.py utilise l'algorithme de Farnebäck pour calculer le flot optique dense entre les images successives d'une vidéo. L'algorithme de Farnebäck est une méthode basée sur la correspondance locale en utilisant l'approximation polynomiale des voisins des pixels.

Principe de l'algorithme de Farnebäck:

1. Construction d'une pyramide d'images pour chaque image, ce qui permet de gérer les grandes vitesses en travaillant à différentes échelles.

2. Pour chaque niveau de la pyramide, approximation des voisins du pixel par un polynôme quadratique en utilisant une fenêtre de taille spécifiée (poly_n) et un lissage gaussien (poly_sigma).

3. Estimation du déplacement local en comparant les polynômes quadratiques approximés des deux images.

4. Propagation du déplacement estimé aux niveaux supérieurs de la pyramide pour affiner l'estimation.

Dans le script, la fonction cv2.calcOpticalfFlowFarlback() est utilisée pour calculer le flot optique dense. Les paramètres importants incluent pyr_scale (taux de réduction pyramidal), levels (nombre de niveaux de la pyramide), winsize (taille de fenêtre de lissage), iterations (nombre d'itérations par niveau), poly_n (taille du voisinage pour approximation polynomiale) et poly_sigma (écart-type de la gaussienne pour le calcul des dérivées).

Le flot optique calculé est converti en coordonnées polaires (magnitude et angle) pour être visualisé en couleurs HSV. La teinte représente l'angle et la valeur représente la magnitude du flot optique.

L'algorithme de Farnebäck calcule un flot optique dense en estimant le déplacement local pour chaque pixel de l'image en utilisant l'approximation polynomiale des voisins des pixels. Il est capable de calculer un flot dense car il estime le déplacement pour chaque pixel, plutôt que seulement pour des points d'intérêt spécifiques.

Pour estimer de grandes vitesses, l'algorithme utilise une approche multi-échelle en construisant une pyramide d'images pour chaque image. En travaillant à différentes échelles, il est capable de capturer des déplacements plus importants entre les images. Les déplacements estimés aux niveaux inférieurs de la pyramide sont ensuite propagés aux niveaux supérieurs pour affiner l'estimation du flot optique.

Q3 Calculer et a-cher pour chaque image de la vidéo, sous la forme d'une image, l'histogramme 2d correspondant à la probabilité jointe des composantes (Vx, Vy) du ot optique. Observer les variations de l'histogramme et expliquer comment il peut être exploité pour identier le type de plan.

4 Découpage et Indexation