Question 1

Les problèmes principaux que nous avons à résoudre dans ce travail pratique est en lien avec le fait qu'il est possible de tromper un programme de détection de coupure et de détection de fondu. Une coupure peut être caractérisée comme un changement de scène où la scène courante est complètement différente à la scène suivante et un fondu peut être caractérisé comme une coupure avec une transition entre la trame avant et après de façon graduelle.

Cependant, il est possible que ces coupures ou fondus soient fausses et qu'il ne s'agit que d'un effet du contenu de la vidéo. En écoutant le court échantillon vidéo qui nous était fourni, nous avons pu identifier 3 effets qui auraient pu gêner à la détection des coupures et fondus :

1. **La vitesse de mouvement**
   Lorsque la caméra suit le mouvement d'un objet ou d'une personne qui se déplace à grande vitesse, celle-ci capture l'arrière-plan qui change continuellement, ce qui pourrait donner l'impression au détecteur que la vidéo est dans une état de transition par fondu.

2. **Le passage d'un objet proche de la caméra**
   Lorsqu'il y a un objet ou une personne qui passe très proche de la caméra, cela donne l'impression comme s'il y a une transition qui passe d'un côté à l'autre.

3. **Le retournement de l'orientation de la caméra**
   Dans ce cas précis, la vidéo va subitement s'inverser, phénomène qui compte comme une coupure entre deux scènes. Ceci nous force à implémenter un détecteur qui a un certain niveau de reconnaissance spatiale.

Question 2

Les solutions que nous avons envisagées pour ces problèmes sont les suivantes :

1. **La vitesse de mouvement**
   Il n'existe pas réellement de solution miraculeuse à ce problème. En effet, les algorithmes de détection ne sont pas encore au point où nous puissions affirmer à 100% de certitude qu'il s'agit bel et bien d'un fondu et non un changement graduel mais rapide de la caméra. Le mieux que nous puissions faire est d'ajuster du mieux possible le seuil délimitant un fondu.

2. **Le passage d'un objet proche de la caméra**
   La différence entre cette situation et une véritable coupure est en lien avec le fait qu'une coupure change sur tout l'image au complet, tandis qu'un simple passage d'objet proche

de la caméra connaît seulement un changement partiel de l'image. Cette caractéristique pourrait être utilisé à notre avantage pour distinguer ces deux situations. En effet, sachant qu'une coupure affecte sur l'image au complet, on doit s'attendre que l'effet de changement soit beaucoup plus frappant d'une simple transition, ce qui fait qu'il devrait être possible de les distinguer simplement en choisissant un seuil raisonnable.

3. **Le retournement de l'orientation de la caméra**
   Cette situation est un peu unique. Il s'agit d'un bon contre-exemple qui montre que certaines méthodes de détection comme celle qui compare l'histogramme entre 2 trames consécutives ne sont pas en mesure de détecter les coupures espérées. En effet, lorsqu'une image est retournée, il s'agit en fait de la même image mais avec ses pixels qui ont changé de place. Puisqu'un histogramme ne fait que détecter la fréquence d'apparition de chaque couleur sans reconnaissance spatiale, un histogramme ne percevait pas de changement entre l'image originale et l'image retournée. La solution serait donc d'utiliser une autre méthode comme la détection par la comparaison d'arêtes. Cette méthode tient compte de la location des pixels du contour, donc elle pourrait facilement détecter ce genre de changement subit.

Question 3

Notre algorithme a été implémenté utilisant le langage de programmation Python. Nous nous sommes également servis de plusieurs librairie de traitement image qui nous ont permi de lire le contenu du vidéo. Entre autres, nous avons utilisé la librairie OpenCV qui nous a permit de lire chaque trame du vidéo. De plus, afin d'avoir une meilleure visualisation de nos données et d'effectuer une analyse de comparaison plus facilement, nous avons utilisé la librairie Matplotlib pour afficher des graphiques et des histogrammes de nos données.

Notre algorithme est constitué de trois parties. En effet, nous avons eu besoin de recourir à plusieurs méthodes de détection différentes afin de complémenter les résultats, car une seule des méthodes ne suffit pas pour tous les trouver. La première étape est de lire et décomposer le vidéo en plusieurs trames. Cela est accompli en utilisant la fonction VideoCapture() et read() de la librairie OpenCV. Ensuite, nous traversons le vidéo au complet en comparant deux trames consécutives à l'aide des méthodes d'histogramme, des arêtes et de différence relatives. Pour finir, nous tentons d'assembler ces résultats en une seule réponse de détection.

Méthode par comparaison d'histogramme

Pour la méthode d'histogramme, nous avons d'abord mesurer l'histogramme des deux images à l'aide de la fonction calcHist de OpenCV. Noter que nous avons décidé de choisir ici une valeur de 64 plutôt que 256 comme les notes de cours pour le paramètre HistSize qui représente le nombre de niveaux pour calculer l'histogramme, ce qui permet de devenir moins sensible aux bruits. Nous avons essayé avec plusieurs valeurs et celle-ci semble donner de

meilleurs résultats. Nous avons ensuite calculé la différence de deux histogrammes avec la formule suivante :

$$\sqrt{\sum_{j=1}^{K}(h_i[j] - h_M[j])2}$$

Cependant, nous avons effectué une modification par rapport à l'algorithme vu au cours. En effet, celui du cours nous donnait des résultats avec beaucoup de faux positifs au niveau des fondus avec des intervalles très petites en plus. Par conséquent, nous avons fait en sorte que les coupures et fondus ne peuvent être détectés seulement si la différence dépasse un premier seuil. Pour distinguer un fondu, nous avons considéré une coupure comme un fondu qui s'étale sur une seule trame, donc si la prochaine trame est en-dessous d'un deuxième seuil, il s'agit d'une coupure. Si la différence demeure au-dessus du deuxième seuil pendant plus d'au moins une trame, il s'agit d'un fondu. Le premier seuil est supérieur au deuxième seuil.

Méthode par comparaison des arêtes

Pour la méthode de détection par les arêtes, nous avons choisi de les faire sur des images de ton gris. En effet, contrairement à l'histogramme qui est sensible aux couleurs, cette méthode a juste besoin de connaître les contours, alors nous avons choisi de manipuler avec des images en tons de gris pour plus de facilité.

Tout d'abord, nous avons ajouté un contour d'une largeur de 1 pixel autour des images afin de pouvoir convoluer sur les contours de l'image. Nous l'avons agrandi d'une largeur de 1 pixel car nous avons choisi de convoluer avec une fenêtre de 3x3.

Ensuite, nous avons créé un filtre de Sobel en x et y et nous avons convolué les deux images en utilisant la fonction Sobel() de la librairie de OpenCV.

Par la suite, nous avons créé des images avec les contours dilatés à partir de ce que nous avons obtenu avec la convolution de Sobel en utilisant la fonction dilate() de OpenCV. Cela va nous permettre de comparer ces images dilatés avec les images de contour avant et après afin de pouvoir calculer les valeurs Pin et Pout qui correspond au nombre de pixel de contour qui s'est déplacé vers l'intérieur et vers l'extérieur de la zone de contour dilaté.

Enfin, nous comparons la valeur maximale entre Pin et Pout avec un seuil, ce qui permet de définir ce qui est une coupure. Nous avons choisi de ne pas considérer la partie de fondu, car cette méthode ne sert que pour complémenter les coupures manquantes que la méthode d'histogramme n'a pas réussi à détecter, soit les coupures par retournement de l'image.

Méthode par différence relatives

Pour la méthode de détection par différence relatives, nous commençons d'abord par transformer les trames dans l'espace de couleur YUV. Par la suite, nous évaluons la différence des valeurs de pixel entre deux trames consécutives en utilisant la même équation que la

méthode de l'histogramme, cette fois appliquée sur les pixels de façon individuelle et sommés sur toute l'image:

$$\sqrt{\sum_{j=1}^{K}(I[j] - M[j])2}$$

Ensuite, pour chaque canal, nous divisons cette valeur de différence avec la valeur calculée pour la paire de trames précédente, puis nous calculons la moyenne de ces différences. Ceci est réalisé dans le but d'atténuer les différences causées par les mouvements rapides mais prolongés, permettant ainsi de détecter une coupure si une trame cause un certain nombre de fois plus de différences que la trame précédente.

Question 4

Les implémentations seront évaluées selon leurs résultats le fichier vidéo AVI fourni, et tenteront la détection de la "vérité terrain" des coupures et fondus présents dans la vidéo. Les six repères à détecter dans la séquence sont les suivantes, telle que spécifiée dans le document du laboratoire:

1. Une coupure à 4.550s (image 137), correspondant à une inversion de l'image.
2. Une coupure à 10.260s (image 308), une véritable coupure entre deux plans.
3. Une coupure à 19.630s (image 589), un deuxième changement de plans.
4. Un premier fondu à environ 31s (images 930 - 940).
5. Une coupure à 35.100s (image 1053), une ré-inversion vers l'image originale.
6. Un deuxième fondu à environ 49s (images 1466 - 1474).

Pour évaluer nos implémentations, nous essayons de trouver un seuil qui permet de détecter un maximum des coupures et fondus tout en gardant au minimum le nombre de faux positifs.
Voici les trames clés détectées par les différentes méthodes implémentées:

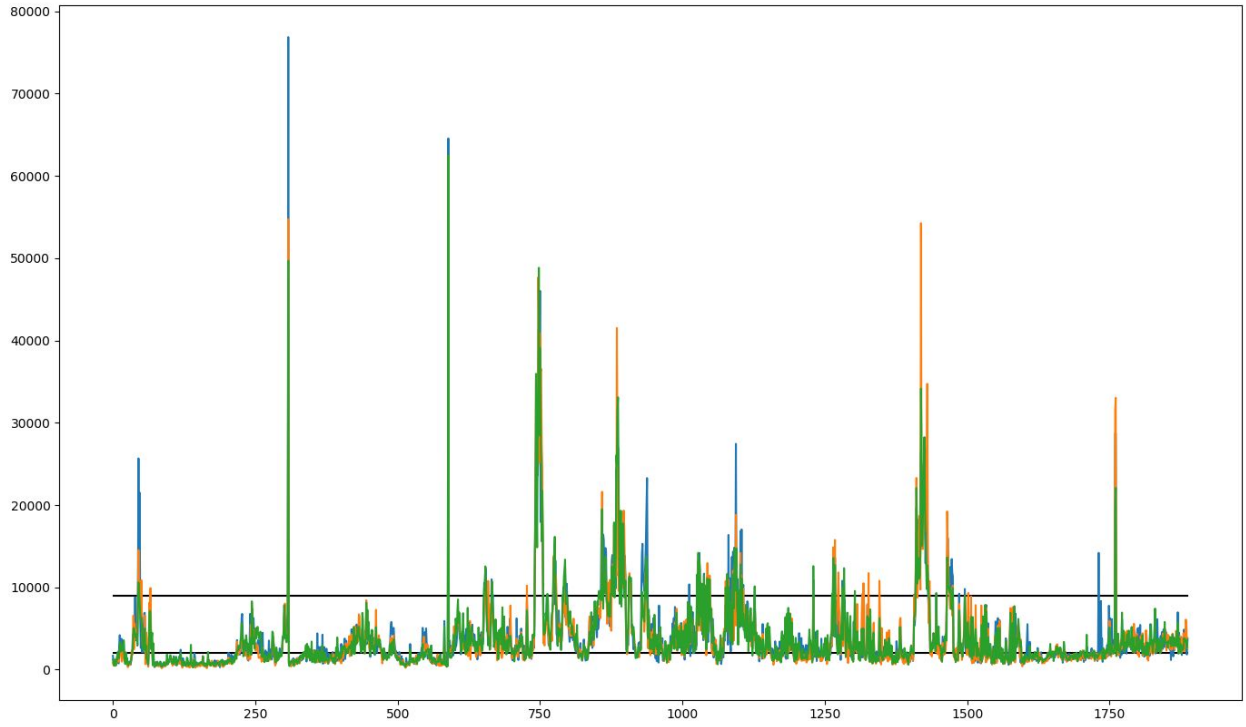
```
PS X:\Bureau\new8770> python main.py
============================ HISTOGRAM ==============================
Fondu entre les trames 45 et 52
Coupure dans la trame 308
Coupure dans la trame 589
Fondu entre les trames 742 et 768
Fondu entre les trames 858 et 903
Fondu entre les trames 938 et 943
Fondu entre les trames 1094 et 1098
Fondu entre les trames 1411 et 1437
Fondu entre les trames 1465 et 1476
Fondu entre les trames 1760 et 1762
============================ OUTLINE ================================
Coupure dans la trame 137
Coupure dans la trame 308
Coupure dans la trame 1053
============================ DIFFERENTIAL ===========================
Coupure dans la trame 63
Coupure dans la trame 137
Coupure dans la trame 308
Coupure dans la trame 589
Coupure dans la trame 1053
```

Histogramme

En définissant le seuil de coupure à 9000 et le seuil de fondu à 2000, on obtient la meilleure détection de coupures et fondus avec un minimum de faux positifs: La méthode peut détecter les coupures aux trames 308 et 589, ainsi que les deux fondus de 930-940 et 1466-1476, avec un léger décalage sur le premier.

La méthode détecte aussi plusieurs transitions parasites, et certaines d'entre elles sont facilement identifiables: les faux positifs de 742-768, 858-903 et 1411-1437 correspondent aux passages du personnel devant la caméra, modifiant momentanément et significativement les histogrammes des trames concernées. Les transitions détectées en 45-52 et 1760-1762 sont causées par l'apparition de coureurs dans l'arrière plan, influençant de manière similaire les histogrammes de couleur.

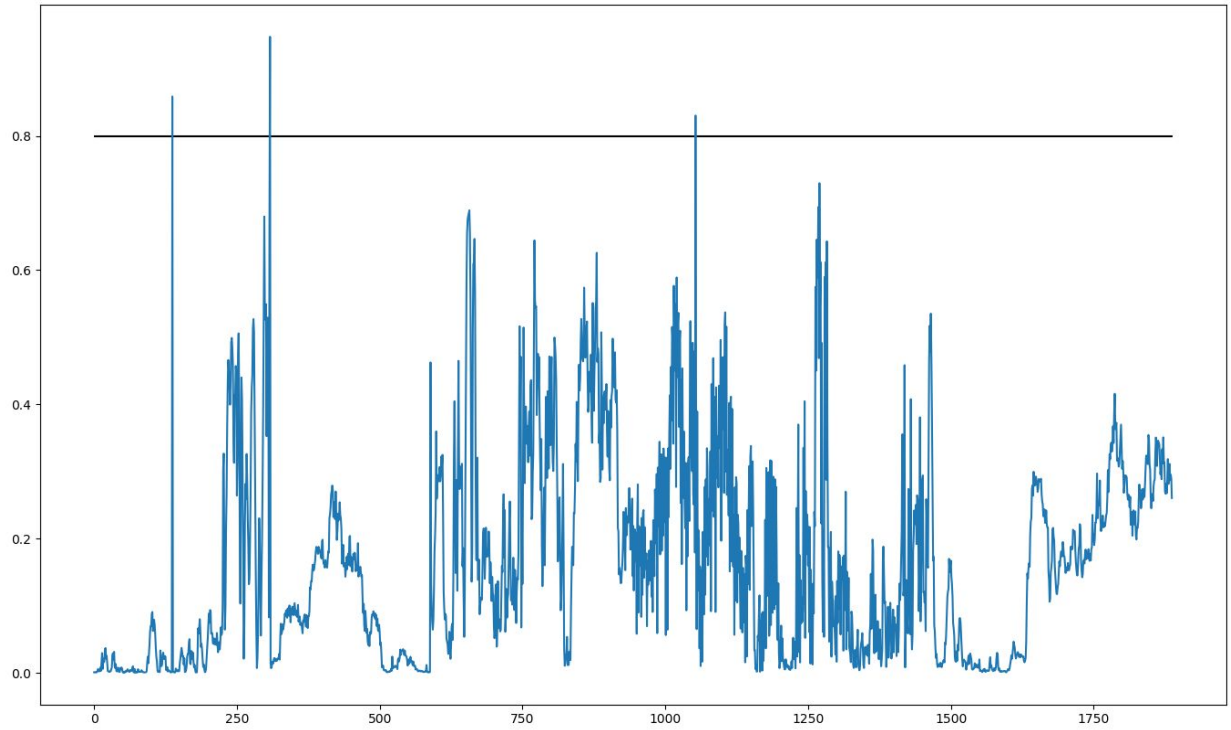
Il est aussi évident que la méthode de l'histogramme est incapable de détecter les coupures par inversion (137, 1053) insérées dans la vidéo: les histogrammes de couleurs sont calculés sans pour deux images ayant des couleurs identiques mais simplement déplacées.



Contours

Avec un seuil de détection de coupures de 80%, la méthode de détection par contours peut retrouver sans faux positifs trois des quatre coupures de la vidéo d'origine, complémentant les résultats du détecteur par histogramme.

Cette méthode permet ainsi de mieux détecter les coupures d'inversions (137, 1053) qui étaient complètement invisibles avec la méthode par l'histogramme. Par contre, la détection de fondus avec cette méthode aurait causé autant de problème de faux positifs que la méthode par histogramme.

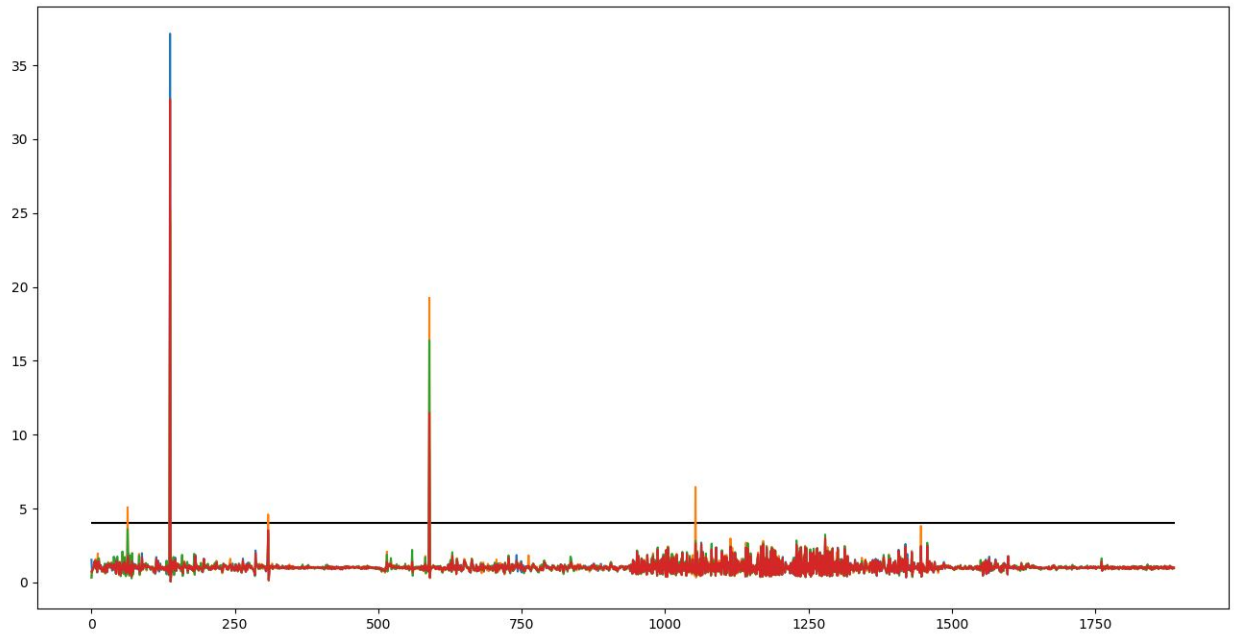


Différence Relatives

En utilisant un seuil de 400%, les coupures de la vidéo (137, 308, 589, 1053) sont facilement détectables même au travers du bruit du mouvement de l'athlète. Malheureusement, les transitions par fondus (935, 1470) deviennent indétectables.

On peut remarquer l'absence totale de pics aux alentours des trames 750, 900 et 1400: le passage du personnel devant la caméra est ignorée car la différence sur chaque trame est trop faible, n'affectant qu'une partie de l'image à la fois.

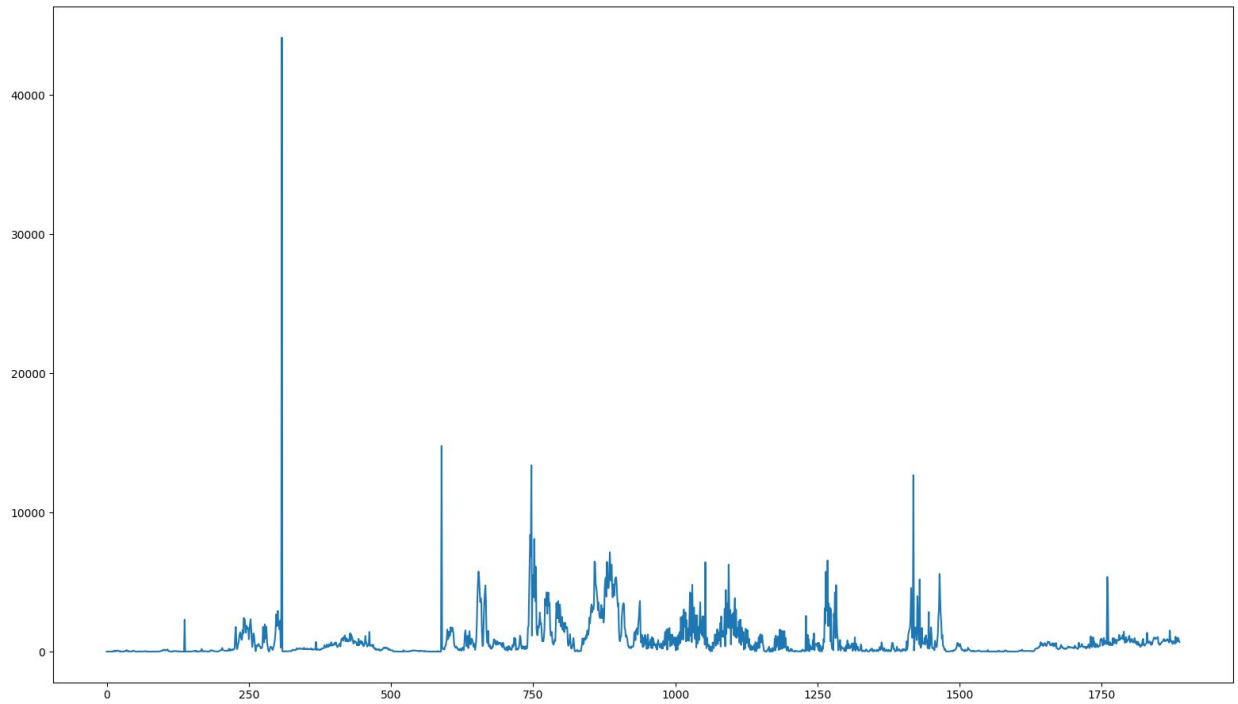
On note aussi un pic apparaissant à la trame 63, qui correspond à l'ajout un bloc d'information surimposé sur l'image d'origine, qui cause une différence significative dans une quantité imposante de pixels.



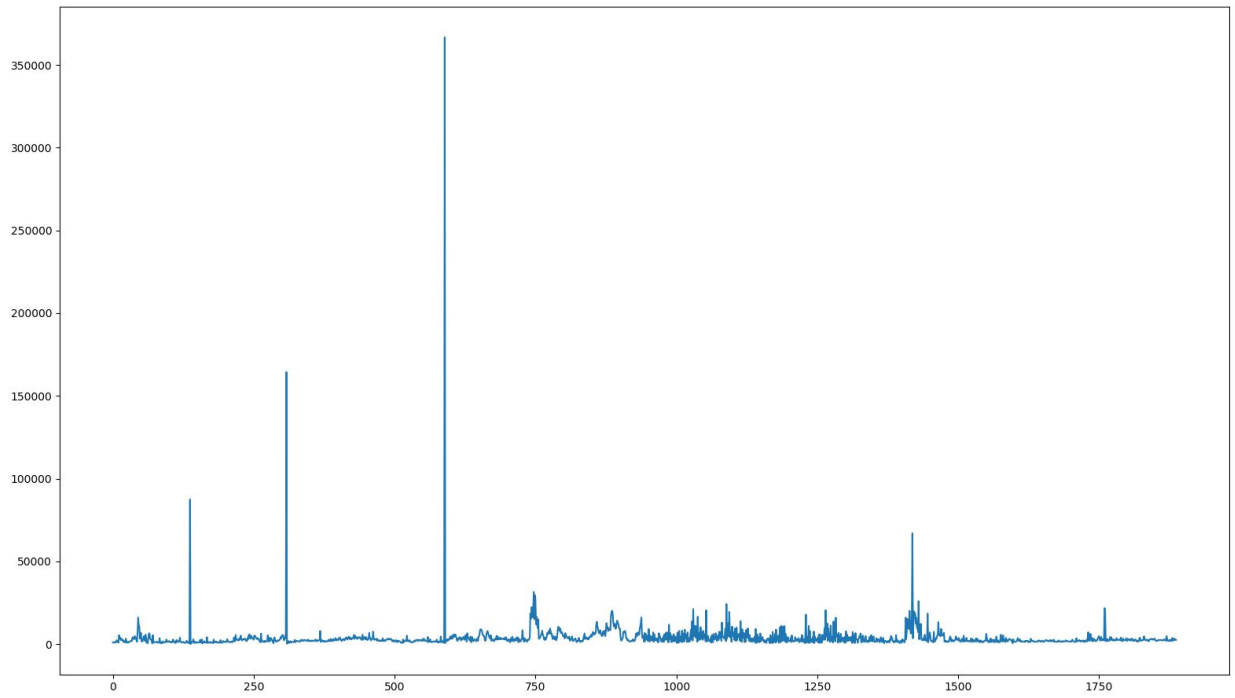
Combinaisons

Afin de tenter de réduire la difficulté d'isoler les coupures et transitions, nous avons appliqué plusieurs méthodes de détection simultanément à la vidéo et multiplié les valeurs de retour. Ceci pourrait, avec le seuil approprié, facilement retourner toutes les coupures sans faux positifs, mais ces méthodes finissent par faire disparaître complètement les transitions en fondus.

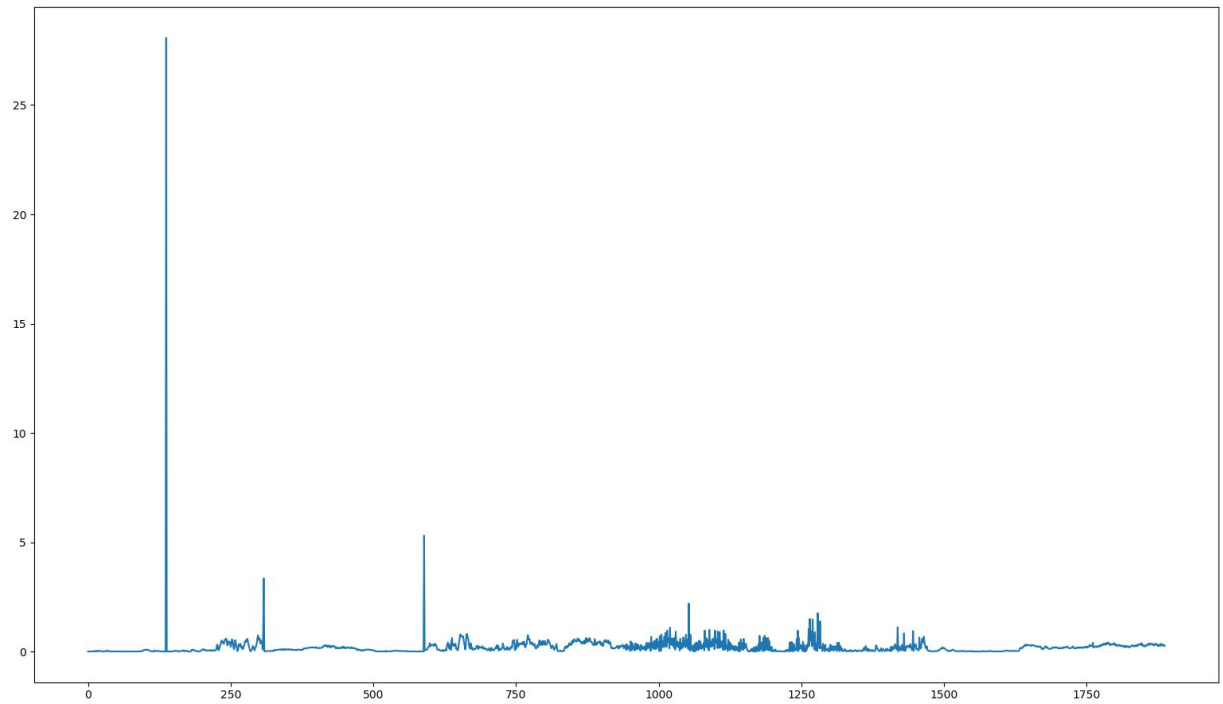
Histogramme-Contours:



Contours-Différence Relative:



Différence Relative-Histogramme:



Question 5

En combinant les résultats des méthodes histogramme, contour, et différence relative, il devient possible de détecter toutes les coupures avec aise.
Indépendamment, chaque méthode présente ses propres limitations:

- La méthode par histogramme est incapable de détecter la différence entre l'image d'origine et l'image retournée puisque les histogrammes de couleurs restent les mêmes sur les deux trames, tel une vidéo normale. Elle peut détecter les transitions par fondus entre deux plans distincts, mais malheureusement la quantité de mouvement de caméras du vidéo d'origine causent beaucoup de faux positifs, détectant comme des transitions le passage d'un individu devant la caméra.
- La méthode par contour est efficace pour trouver des coupures claires, incluant les retournements d'image, mais l'analyse d'images présentant trop de détails empêche la comparaison précise d'arêtes avec l'image qui suit: le nombre de détails imposants donnent trop de zones de contours après dilatation, et les contours détectés de la trame suivante ne sont pas détectés comme des contours "sortants". La méthode présente aussi beaucoup trop de bruit pour clairement identifier les fondus.
- La méthode par différence relative permets de minimiser le bruit provenant de scènes mouvementées de la vidéo, mais devient inefficace lorsqu'il s'agit de détecter les transitions graduelles entre les scènes.

Les trois méthodes souffrent toutefois du même problème: elles n'évaluent les trames du vidéo que par paires, empêchant de distinguer des changements dûs aux mouvements et ceux dûs au fondus. Les séquences mouvementées de la vidéo nous empêchent de bien implémenter le Critère #2 de la décomposition en prises de vue des notes du cours par méthode d'histogramme car le seuil ts qui permettrait de détecter les fondus recherchés cause un grand nombre de faux positifs lors de la détection. Si, par exemple, nous avions implémenté des comparaisons sur un nombre supérieur d'image, il aurait probablement été possible de déterminer des différences anormales sur plusieurs images et mieux distinguer les mouvements des transitions en fondus.