ROB317 - TP1
Détection et Appariemment de Points Caractéristiques
Compte-rendu

Juliette DRUPT - Caroline PASCAL

L'ensemble de notre code est accessible sur GitHub : https ://github.com/JulietteDrupt/ROB317-TP2 ; et les vidéos que nous avons utilisées en plus de celles fournies pour le TP sont disponible à l'adresse : https ://drive.google.com/drive/folders/15Twh0yWuV48x0vsc5YEsHpJzCQxAAuwa ?usp=sharing.

1 Question 1

De manière générale, sur toute la durée d'un plan, la forme de l'histogramme 2D des composantes chromatiques (u, v) du codage Yuv varie de façon continue. En revanche, lors d'un changement de plan, en particulier lorsque la transition entre les deux plans est un *cut*, la forme de cet histogramme est modifiée de manière abrupte.

Ainsi, une méthode simple pour détecter un changement de plan, et en particulier les *cuts*, consiste à calculer pour chaque paire d'images successives, la différence pixel à pixel, ou *bin-to-bin*, des deux histogrammes 2D de ces images. Lorsque cette différence présente un maximum à l'échelle de l'extrait vidéo, c'est que nous sommes en présence d'un changement de plan.

Afin de détecter ces maxima, nous calculons un seuil représentatif de la distribution des différences entre les histogrammes 2D des images de l'extrait vidéo. Ce seuil est donné par la formule suivante :

$$seuil = \mu_{difference} + \frac{3}{2}\sigma_{difference} \tag{1}$$

Où $\mu_{difference}$ correspond à la moyenne des différences entre les histogrammes 2D des images de l'extrait vidéo et $\sigma_{difference}$ à son écart-type.

Finalement, afin de diminuer le nombre de fausses détections, dues en particulier à des modifications brèves mais intenses de la luminosité ou de la colorimétrie des images, nous choisissons de détecter les maxima de la valeur absolue de la dérivée des différences d'histogrammes 2D, plutôt que ceux des différences elles-mêmes. En effet, la dérivation va agir comme un filtre passe-haut, en amplifiant les variations brèves, que nous cherchons à détecter, et en diminuant les variations plus étendues, souvent parasites - cf. figure 1.

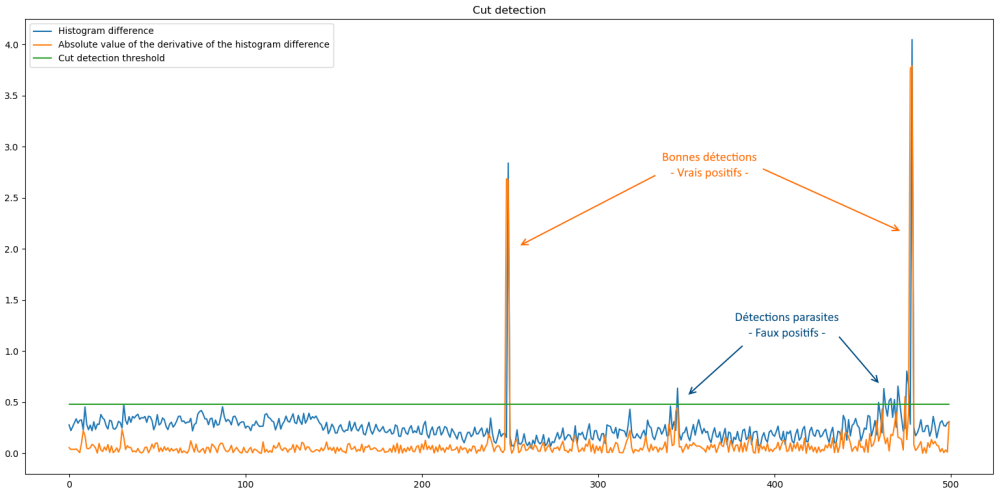


FIGURE 1 – Exemple de détection de changement de plan sur les 500 premières frames de l'extrait vidéo *Extrait1-Cosmos_ Laundromat1(340p).m4v*

Nous ferons toutefois attention à bien supprimer les maxima doublons issus du calcul de la valeur absolue de la dérivée [1]. Pour être certain de bien détecter la transition, nous choisissons de retenir le deuxième indice pour représenter le maximum.

Dans le cas d'une vidéo monochrome, c'est à dire ne pouvant pas être encodée en Yuv, il suffit de calculer l'histogramme 1D de l'intensité des pixels à la place de l'histogramme 2D calculé plus haut. Les étapes suivantes de la méthode proposée restent alors identiques.

2 Question 2

La fonction utilisée pour le calcul du flot optique dense est la fonction `calcOpticalFlowFarneback` d'OpenCV, qui prend en paramètre deux images successives et retourne le flot optique calculé entre ces deux images.

Le flot otique calculé est représenté sous la forme d'une image de la même dimension que les images étudiées, et comportant deux couches : la première représente la composante horizontale du flot, et la seconde, la composante verticale.

A l'inverse du calcul du flot optique discret, nous ne pouvons pas représenter le flot optique dense sous la forme de vecteurs : en effet, avec une telle représentation, la densité des valeurs calculées rendrait la visualisation du flot optique impossible. Ainsi, afin de visualiser correctement le flot optique dense calculé, ce dernier est converti sous la forme d'une image en couleur, codée selon la convention HSV. L'argument du flot optique *arctan* permet de coder la valeur de teinte `H`, et sa norme *sqrt* permet de définir la valeur de la valeur `V` pou chaque pixel de l'image. La valeur de la saturation est fixée à sa valeur maximale pour tous les pixels de l'image.

Cette représentation nous permet d'obtenir une visualisation très efficace du flot optique : la couleur nous permet d'identifier l'orientation du flot optique tandis que l'intensité de la couleur nous donne la norme du flot optique.

La fonction `calcOpticalFlowFarneback` prend également en argument six paramètres, qui permettent de paramétriser les différents outils utilisés lors du calcul du flot optique dense par méthode de Farnebäck.

1. La dérivée d'un "pic" est un "pic" suivi d'un autre "pic" d'amplitude opposée - Le calcul de la valeur absolue va donc mener à deux maxima successifs !

La méthode de de Farnebäck est une méthode itérative et multi-échelle d'estimation du champ de déplacement, ou flot optique, entre deux images. Cette méthode repose sur une approximation locale du voisinnage d'un pixel sous la forme d'un polynôme quadratique, appelée *polynomial expansion*. Les coefficients du polynôme sont obtenus au moyen d'une méthode des moindres carrés, pondérée par deux paramètres : la *certainty*, relative aux valeurs des pixels du voisinnage, et l'*applicability*, qui caractérise la manière dont est réalisée l'approximation. En particulier, la *certainty* module le poids accordé à chaque pixel en fonction de sa position sur l'image, ou de son incertitude de mesure, et l'*applicability* définit la taille du voisinnage sur lequel sera réalisée l'approximation, et donc par extension, l'échelle des éléments qui seront approximés.

Dès lors, il est possible d'encapsuler le voisinnage de chaque pixel des deux images successives sous la forme d'un polynôme, dont les coefficients sont propres au pixel étudié. Connaissant l'ensemble de ces polynômes, il est alors possible de déterminer une estimation du déplacement d'un pixel de la première image, en minimisant sur un certain voisinnage (i.e. localement) une expression comparant l'approximation polynomiale de ce pixel sur la première et celle du pixel situé aux même coordonnées sur la seconde image. Comme le pixel considéré sur la seconde image est situé aux même coordonnées que celles du pixel étudié sur la première image, cette estimation n'est en réalité valable que sous l'hypothèse que le champ de déplacement varie lentement entre les deux images. En répétant cette opération sur l'ensemble des pixels des deux images, nous obtenons finalement une estimation dense, c'est-à-dire, définie de manière presque continue pour chaque pixel de la première image, du champ de déplacement entre les deux images étudiées.

Afin de s'affranchir de l'hypothèse d'un champ de déplacement variant lentement, limitant l'approche précédente à de petits déplacements, deux améliorations sont mises en place. Tout d'abord, la méthode précédente est améliorée de telle sorte qu'il soit possible d'y intégrer une connaissance préliminaire du champ de déplacement. Ainsi, au lieu de considérer sur la seconde image le pixel de même coordonnées que le pixel étudié sur la première image, c'est le pixel situé aux coordonnées translatées selon le champ de déplacement intuité qui sera considéré. Dans les deux cas, le voisinnage de pixels dans lequel la minimisation est réalisée reste géométriquement le même.

Dans un second temps, une approche itérative et multi-échelle, explicitée sur le diagramme 2, est proposée.

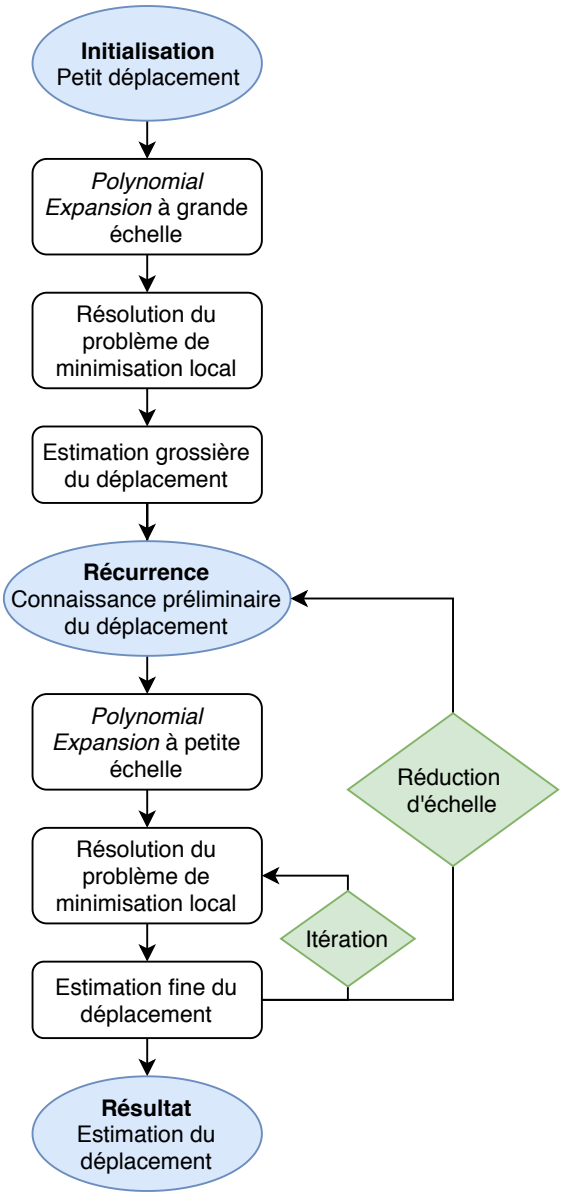


FIGURE 2 – Représentation schématique du fonctionnement de la méthode de Farnebäck

L'aspect multi-échelle de cette approche est implémenté au moyen d'une pyramide multi-échelle de l'image, dont nous avons détaillé le formalisme dans le compte rendu précédent - le fonctionnement du détecteur ORB repose également sur l'utilisation d'une pyramide multi-échelle.

L'utilisation couplée d'une méthode itérative et multi-échelle pour réaliser le calcul du champ de déplacement permet à la fois d'améliorer la précision du résultat, ainsi que sa robustesse. En particulier, cette méthode permet de quasiment éliminer l'hypothèse de petits déplacements, qui ne subsiste finalement que dans la phase d'initialisation. En conséquence, il est désormais possible de détecter et d'estimer de grands déplacements de pixels, et in fine d'estimer de grandes vitesses de déplacement.

Comme mentionné précédemment, six paramètres pris en argument de la fonction `calcOpticalFlowFarneback` permettent de moduler l'application de la méthode de Franebäck :

— `levels` définit le nombre de niveaux de la pyramide multi-échelle utilisée ;

— `pyr_scale` caractérise le facteur d'échelle de la pyramide multi-échelle, et définit la manière dont la résolution de l'image est réduite (resp. augmentée) entre chaque niveaux de la pyramide ;

— `winsize` correspond à la taille de la fenêtre carrée définissant le voisinnage sur lequel est résolu le problème de minimisation permettant d'estimer le champ de déplacement entre les deux images. La taille de cette fenêtre impacte directement l'échelle des déplacements calculés : plus cette fenêtre sera grande, plus l'amplitude des déplacements estimés pourra être importante, et plus l'estimation sera robuste au bruit de l'image (effet "lissant"). Toutefois, l'utilisation d'une fenêtre de taille trop importante peut entraîner une mauvaise détection des faibles déplacements, et mener à une estimation peu précise, voire incorrecte, du flot optique ;

— `iterations` correspond au nombre d'itérations à réaliser lors de l'estimation du champ de déplacement à chaque niveau d'échelle (i.e. niveau de la pyramide multi-échelle). En pratique, un nombre important d'itérations va permettre d'obtenir une estimation plus précise du déplacement de chaque pixel, mais implique généralement une augmentation importante du temps de calcul ;

— `poly_n` renseigne la taille de la fenêtre carrée sur laquelle est réalisée l'approximation par *polynomial expansion* du voisinnage de chaque pixel des deux images. En d'autres termes, cette fenêtre correspond (en partie) à l'*applicability* évoquée plus haut. De la même manière que pour `winsize`, la taille de la fenêtre utilisée pour réaliser l'approximation polynomiale va directement impacter l'échelle des structures approximées. Par exemple, une fenêtre de taille importante va permettre d'obtenir une approximation "lissée", et donc plus robuste au bruit de l'image. En conséquence, le champ de déplacement estimé sera également plus robuste au bruit de l'image, mais perdra certainement en précision ;

— `poly_sigma` correspond à l'écart-type du noyau gaussien dérivé utilisé pour le calcul des cœfficients des polynômes de la *polynomial expansion*. La valeur de cet écart-type est directement reliée à l'échelle des éléments dont l'approximation par *polynomial expansion* doit rendre compte. Ainsi, une valeur d'écart-type `poly_sigma` incohérente par rapport à une fenêtre de taille `poly_n`, pourra entraîner une perte de robustesse dans le cas d'un écart-type trop faible, ou une perte de précision dans le cas inverse ;

Comme nous l'avons détaillé dans le compte rendu précédent, l'effet des deux premiers paramètres sur le calcul du flot optique dense est couplé. Un facteur d'échelle faible (i.e. proche de 1), nécessite l'ajout de niveaux de pyramide supplémentaires afin de parcourir une certaine gamme d'échelles sur l'image, et sera donc plus coûteux en temps de calcul. A l'inverse, un facteur d'échelle plus grand permettra de parcourir la même gamme d'échelle en moins de niveaux de pyramide, mais entraînera une dégradation importante dans la précision du champ de déplacements calculé. Aussi, il est indispensable de trouver un bon compromis entre temps de calcul et précision du calcul, de même qu'entre le facteur d'échelle et le nombre de niveaux de la pyramide multi-échelle.

3 Question 3

On complète les programmes fournis afin de calculer et afficher l'histogramme 2D correspondant à la probabilité jointe des composantes (V_x, V_y) du flot optique pour chaque image, avec les flots *Dense* et *Sparse*.

Afin d'observer au mieux les variations de ces histogrammes en fonction du type de plan, nous avons utilisé des vidéos montrant chacune un type de plan parmi les suivants :

— Plan fixe ;

— Panoramique horizontal (PAN) ;

— Panoramique vertical (TILT) ;

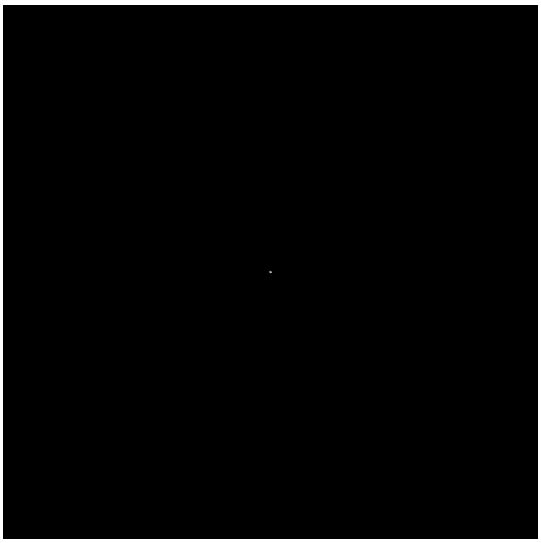
— Rotation ;

— Travelling horizontal ;

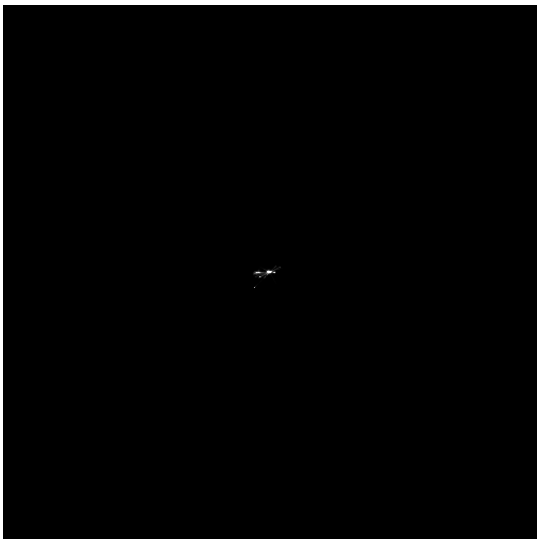
— Travelling avant ;

— Zoom avant.

Les figures 3, 4, 5, 6, 7, 8 et 9 montrent des images représentatives des histogrammes caractéristiques observés.

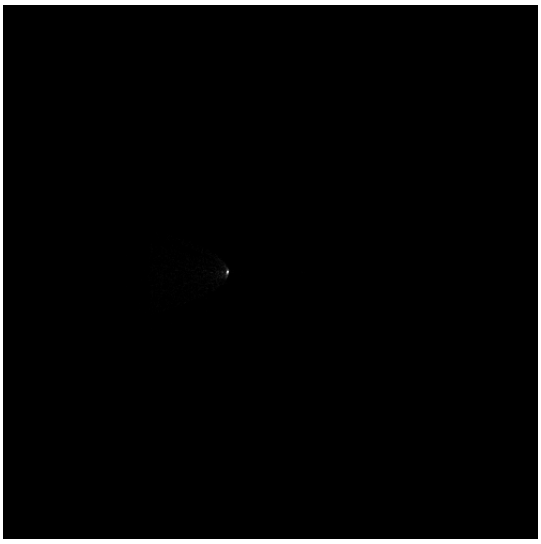


(a) *Sparse* : point blanc central.

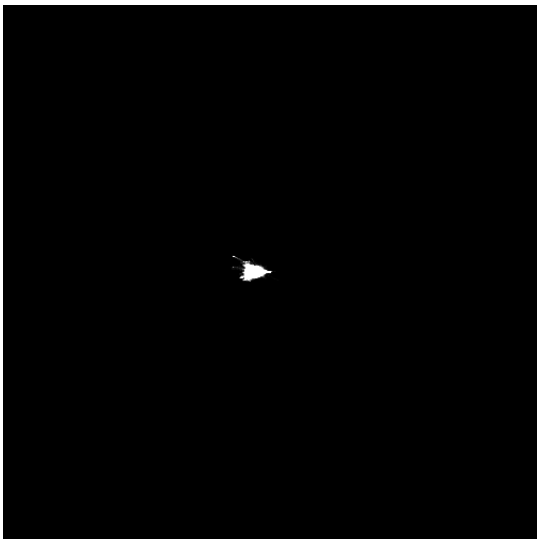


(b) *Dense* : tâche blanche centrale.

FIGURE 3 – Allure de l'histogramme 2D pour un plan fixe.

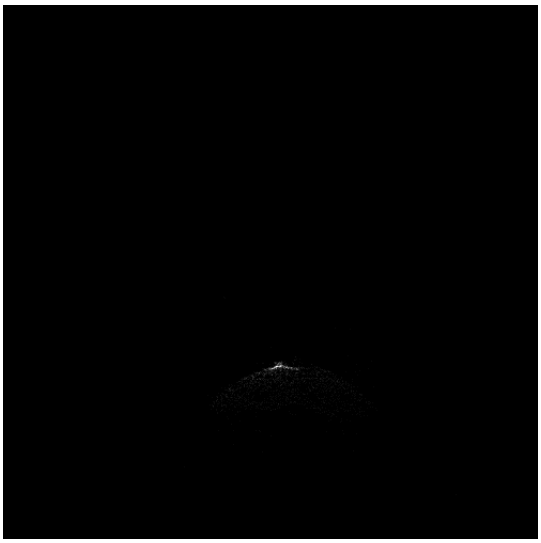


(a) *Sparse* : point blanc centré à gauche.

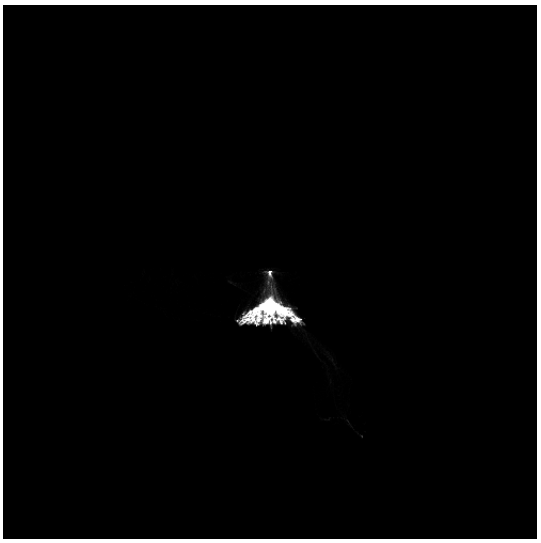


(b) *Dense* : cône blanc sur l'axe horizontal, du côté gauche, dont le sommet est vers le centre.

FIGURE 4 – Allure de l'histogramme 2D pour un PAN (vers la droite).

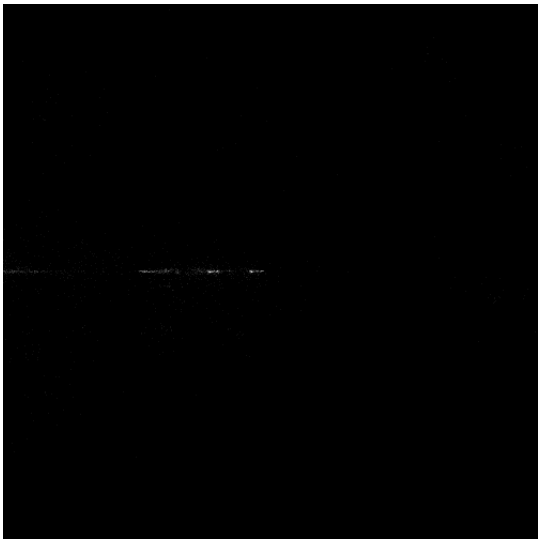


(a) *Sparse* : point blanc centré bas.

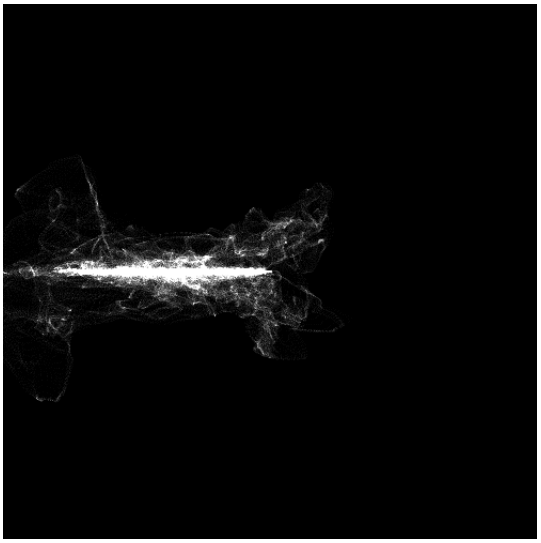


(b) *Dense* : cône blanc sur l'axe vertical, dans la moitié inférieure de l'image, dont le sommet est proche du centre (un peu au dessous)

FIGURE 5 – Allure de l'histogramme 2D pour un TILT.

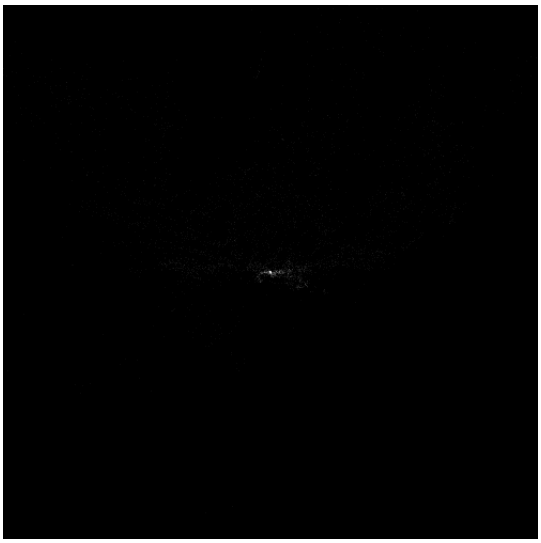


(a) *Sparse* : ligne blanche horizontale allant du centre vers la gauche.

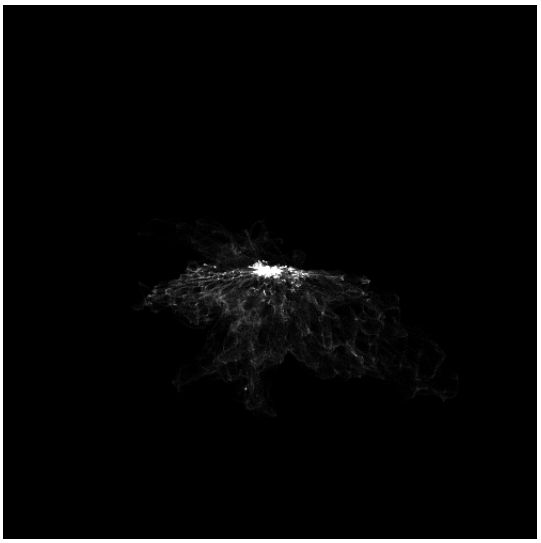


(b) *Dense* : ligne blanche horizontale allant du centre vers la gauche, mais plus épaisse que pour *Sparse* et avec une sorte de halo.

FIGURE 6 – Allure de l'histogramme 2D pour un travelling avant (vers la droite).

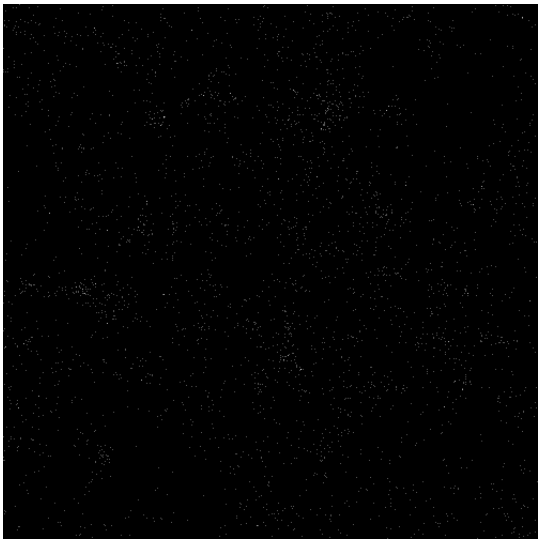


(a) *Sparse* : tâche blanche centrale.

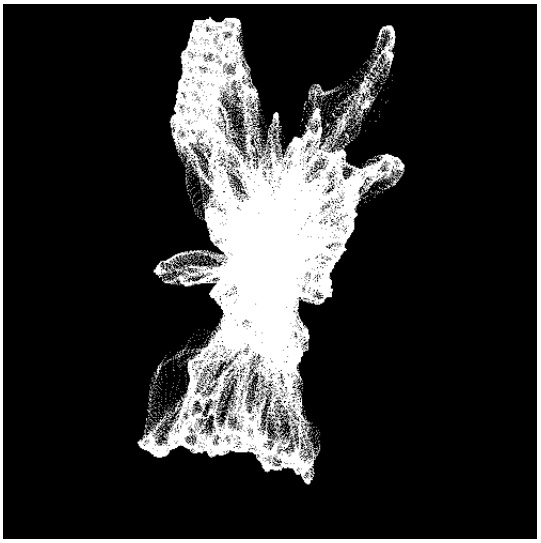


(b) *Dense* : tâche blanche centrale avec un halo vers le bas.

FIGURE 7 – Allure de l'histogramme 2D pour un travelling avant.



(a) *Sparse* : points blancs répartis dans toute l'image de manière apparemment homogène.



(b) *Dense* : grande forme blanche mouvante d'une image à l'autre.

FIGURE 8 – Allure de l'histogramme 2D pour une rotation.

(a) *Sparse* : tache blanche centrale. (b) *Dense* : tâche blanche centrale.

FIGURE 9 – Allure de l'histogramme 2D pour un zoom avant.

On pourrait donc envisager d'identifier la nature d'un plan parmi ces types de plans-ci en comparant l'histogramme 2D de ce plan aux histogrammes caractéristiques présentés ici. On constate toutefois qu'il est difficile de faire la différence entre certains plans à partir de l'histogramme 2D lié au flot *Sparse* : le zoom avant, le travelling avant et le plan fixes sont ainsi quasi impossibles à discerner de cette manière par l'oeil humain. Les différences entre histogrammes sont en revanche plus évidentes pour le flot *Dense* ; c'est donc plutôt celui-ci qu'on utilisera pour l'identification du type de plan.

4 Question 4

La méthode présentée dans la question 1 nous fournit un premier outil - très simple - permettant de détecter les transitions entre deux plans, et donc, à fortiori, de découper en plans n'importe quel extrait vidéo.

Toutefois, après avoir confronté cette méthode aux extraits vidéos proposés dans ce TP, nous avons rapidement mis en évidence trois défauts :

— Une grande sensibilité aux **variations de luminosité, ou de coloration**, particulièrement observable sur les extraits *Extrait3-Vertigo-Dream_ Scene(320p).m4v* et *Extrait1-Cosmos_ Laundromat1(340p).m4v* (figure 10), dans lesquels nous retrouvons à la fois des variations brèves de luminosité (comme des éclairs, par exemple), et de coloration ;

— Une grande sensibilité aux **saccades dans les vidéos peu ou mal stabilisées**, comme l'extrait *Extrait4-Entracte-Poursuite_ Corbillard(358p).m4v* (figure 11) ;

— Une faible capacité à détecter les transitions continues entre deux plans, comme les *dissolve* ou les *fade-in* de l'extrait *Extrait3-Vertigo-Dream_ Scene(320p).m4v* (figure 12).

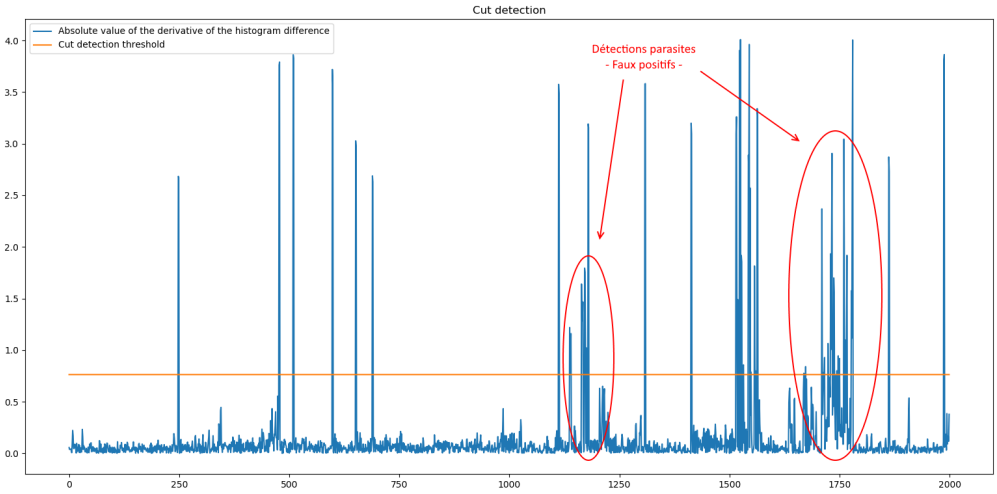


FIGURE 10 – Illustration de la sensibilité à la luminosité - Les régions indiquées en rouge correspondent à des détections parasites dues à des variations de luminosité

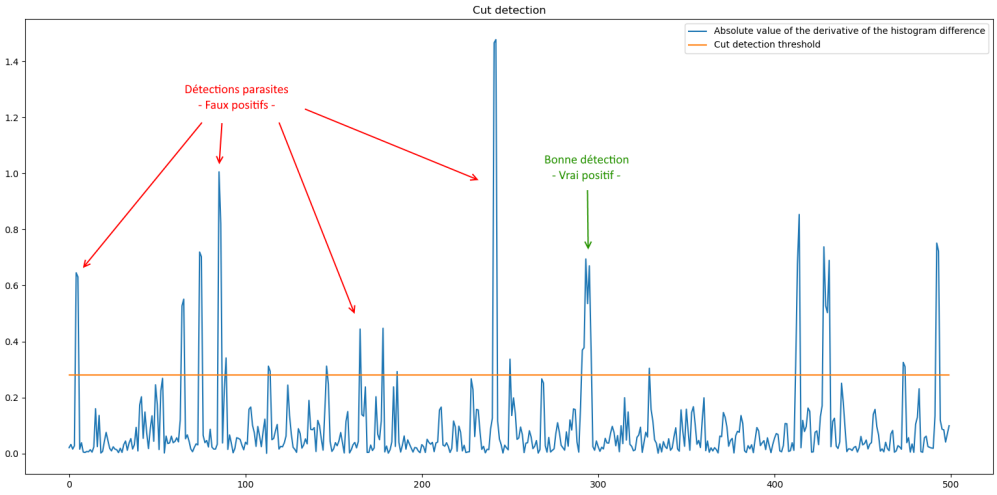


FIGURE 11 – Illustration de la sensibilité aux saccades - De nombreux maximas, détectés comme des transitions, correspondent en réalité à des saccades dans l'extrait vidéo

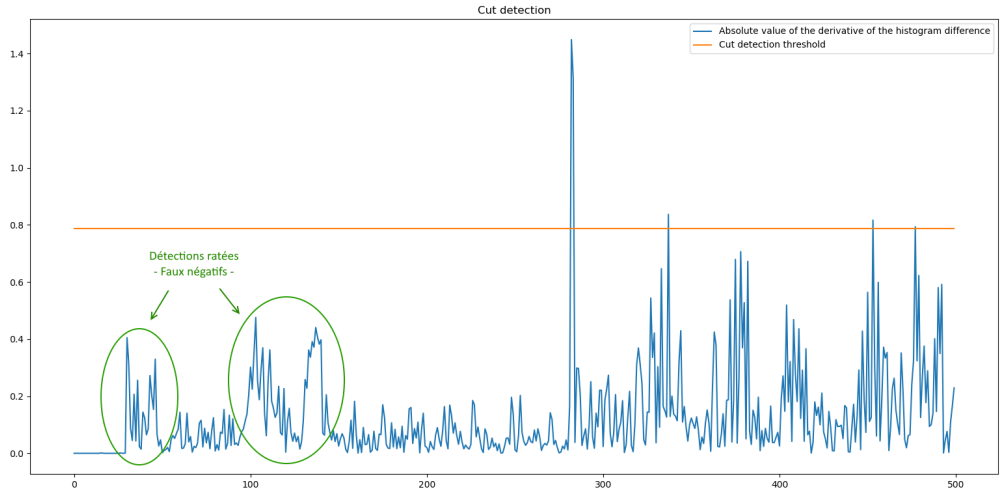


FIGURE 12 – Illustration de la non-détection des transitions continues entre deux plans - Les régions indiquées en vert correspondent à des transitions continues non-détectées

Afin de résoudre les problèmes de détection, et de non-détection, causés par les défauts de sensibilité de la méthode initialement proposée, nous avons trouvé deux pistes d'amélioration.

4.1 Élimination des faux positifs : prise en compte de la dissimilitude des images lors d'un changement de plan

La trop grande sensibilité de la méthode initiale envers la luminosité, la coloration, mais aussi les saccades, est en grande partie due à la perte d'informations spatiales lors du calcul de l'histogramme de l'image. En effet, l'histogramme, 1D ou 2D, ne rend compte que de la répartition statistique des couleurs des pixels, ou de leur intensité, sur l'image, sans nous donner d'informations sur la position effective de chaque pixel. Ainsi, lors d'une modification de la luminosité ou de la coloration d'un plan, les valeurs portées par chaque pixels peuvent varier de façon très importante, alors que l'objet, ou les formes représentées restent identiques. Il en résulte alors les détections parasites de changement de plan que nous avons mis en évidence dans la partie précédente. De la même manière pour les plan saccadés, la répartition des pixels peut être amenée a varier fortement, alors que la scène reste bien la même.

Cette observation met en exergue la nécessité d'introduire un outil de mesure de similitude entre deux images, qui ne dépende pas de la répartition statistique des valeurs portées par les pixels, mais plutôt des objets, ou des formes présentes sur ces images.

Dans notre implémentation, nous avons décidé d'utiliser le descripteur-detecteur ORB pour réaliser cette mesure de similitude. Pour chaque couple d'image à comparer, nous calculons les correspondances entre les points d'intérêt détectés sur les deux images : si le nombre de correspondances correctes après le test du ratio est supérieur à un certain seuil, nous en déduisons que les images sont similaires.

Une fois en possession d'un tel outil, il est alors possible de réaliser une vérification de la pertinence des maxima détectés. Pour chaque nouveau maximum, la similitude entre l'image située deux *frames* avant ce maximum et celle située deux *frames* après est calculée : si les images sont similaires, il n'y a pas eu de changement de plan, et nous avons à faire à un faux positif !

Cette méthode a montré des résultats prometteurs sur les extraits vidéos dans lesquels la luminosité et la coloration varient beaucoup (figure 13), et de très bons résultats pour les extraits vidéos saccadés (figure 14).

Toutefois, la mesure de similitude peut d'avérer inefficace lorsque les images sont peu texturées, et que très peu de points d'intérêts sont détectés, comme sur les images représentant le ciel dans les extraits *Extrait5-Matrix-Helicopter_Scene(280p).m4v* et *Extrait1-Cosmos_Laundromat1(340p).m4v*. De plus,

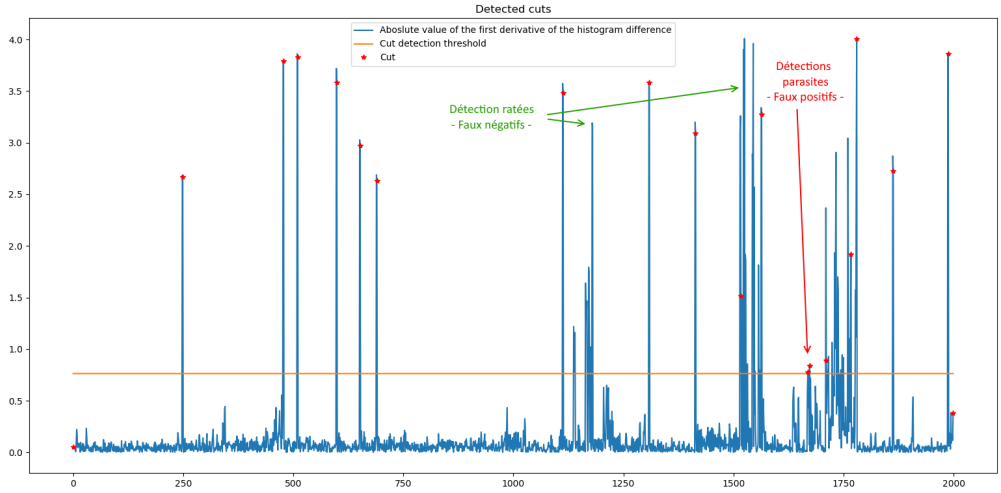


FIGURE 13 – Détection des *cuts* sur l'extrait *Extrait1-Cosmos_ Laundromat1(340p).m4v*

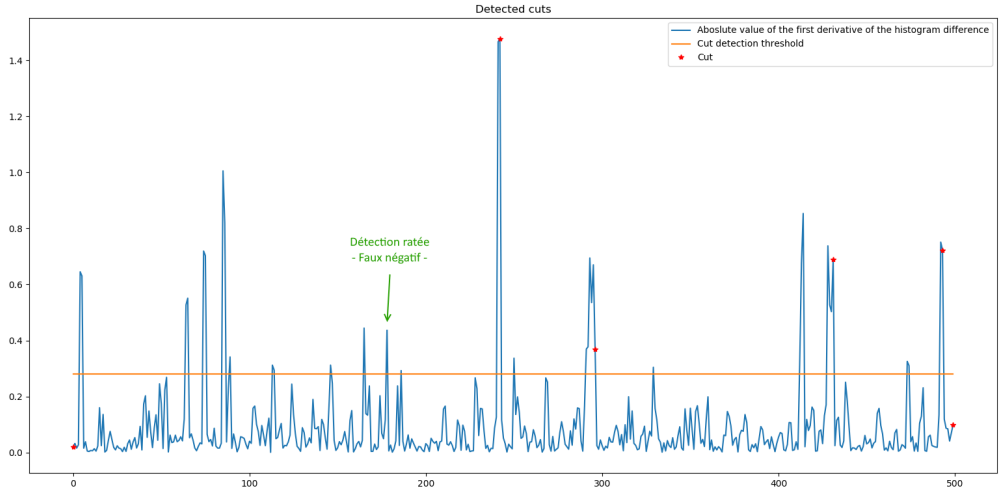


FIGURE 14 – Détection des *cuts* sur l'extrait *Extrait4-Entracte-Poursuite_ Corbillard(358p).m4v*

la détection des points d'intérêts s'avère également très sensible à la luminosité, comme l'illustrent les faux négatifs et faux positifs de la figure 13, qui présente les résultats obtenus pour l'extrait *Extrait1-Cosmos_ Laundromat1(340p).m4v*.

De même, notre méthode de vérification ne montre pas de bons résultats pour les plans en mouvement rapides, dans lesquels les prises de vue peuvent grandement varier sur une faible échelle de temps - l'extrait *Extrait5-Matrix-Helicopter_ Scene(280p).m4v* en est un très bon exemple. Dans ce genre de situation, une solution pourrait consister à réaliser un suivi "image par image" des points d'intérêt tout au long de l'extrait vidéo. Le choix des paires d'images à comparer pourrait alors s'effectuer en se servant de l'estimation du flot optique, qui nous permettrait de calculer la durée au bout de laquelle chaque point d'intérêt va sortir du champ de la caméra, et donc sur quel intervalle d'images la mesure de similitude est pertinente. Toutefois, l'apport en terme de qualité des résultats d'une telle solution reste, à notre avis, très faible par rapport à son coût en terme de temps de calcul.

Finalement, à l'inverse de la méthode précédente, celle-ci ne détecte pas tout le temps l'intégralité des transitions de type *cut* (faux négatifs), en particulier sur les extraits vidéos de basse qualité comme *Extrait4-Entracte-Poursuite_ Corbillard(358p).m4v* et *Extrait2-ManWithAMovieCamera(216p).m4v*, sur lesquelles la mesure de similitude est souvent approximative. En effet, sur ces extraits les *cuts* ont tendance à être bruités par un "voile blanc", qui parasite la mesure de similitude entre les images que nous supposons situées "avant" et "après" la transition. Dans ces conditions, il arrive qu'un maximum soit rejeté alors qu'il correspondait bien à un *cut*. Pour résoudre ce problème, il faudrait soit augmenter le seuil sur les correspondances utilisé lors du calcul de la similitude, soit augmenter la distance entre les deux images comparées afin d'évaluer la pertinence d'un maximum donné.

4.2 Élimination des faux négatifs : détection des transitions continues

La question de l'absence de détection des transitions continues, en particulier au début de l'extrait *Extrait3-Vertigo-Dream_ Scene(320p).m4v*, est un peu plus problématique. Ces transitions continues, appelées *dissolve* ou *fade-in/fade-out*, sont principalement caractérisées par une évolution linéaire de l'intensité de chaque pixel, comme illustré sur la figure 15. Ce genre de transition génère également un maximum local dans les différences entre les histogrammes 2D des images, mais bien plus faible que les maxima globaux des *cuts*.

Afin de détecter ces transitions, nous avons décidé de procéder en deux étapes.

Tout d'abord, nous appliquons la méthode décrite dans la partie précédente afin de détecter les transitions de type *cut*. Une fois cette première vague de détection réalisée, nous sommes alors capables d'effectuer un premier découpage en plans *cut* de l'extrait vidéo.

La deuxième étape de notre méthode de détection des transitions continues consiste alors à réaliser une seconde vague de détection de maxima sur chacun des plans alors définis. Lors de cette seconde passe, nous utilisons à nouveau le seuil que nous avons défini dans l'équation 1, calculé cette fois à l'échelle du plan, et en choisissant un facteur multiplicatif égal à 1 au lieu de $\frac{3}{2}$. Ce changement d'échelle et de facteur multiplicatif vont nous permettre de détecter cette fois des maxima locaux, qui peuvent signifier la présence de transitions continues - cf. figure 12.

Une fois ces maxima détectés, il ne reste plus qu'à déterminer s'ils correspondent bien à des transitions continues. Pour ce faire, nous calculons la dérivée de la moyenne de l'intensité des pixels des images du plan, au moyen d'un filtre de Stavisky-Golay de fenêtre égale à 5, et utilisant une interpolation polynomiale de degré 2. Un maxima correspondra alors à une transition continue, s'il existe autour de lui un voisinnage d'images assez grand sur lequel la fonction dérivée précédemment calculée est de signe constant. En effet, comme nous l'avons indiqué plus haut, une transition continue est caractérisée par une évolution linéaire de l'intensité de chaque pixel entre les différentes images de la transition. Si un tel voisinnage existe, il correspond en réalité à la fenêtre sur laquelle se déroule la transition continue.

Une fois les transitions continues identifiées, nous réalisons une dernière vérification de dissimilitude entre la première et la dernière image de la transition, afin de s'assurer qu'il s'agit bien d'un changement de plan, et pas d'un simple mouvement de caméra. A l'instar de la détection des *cuts*, cette vérification est réalisée avec l'outil de mesure de similitude que nous avons présenté dans la section précédente.

Cette méthode a abouti a des résultats très satisfaisants pour l'extrait *Extrait3-Vertigo-Dream_ Scene(320p).m4v*, illustrés sur la figure 15.

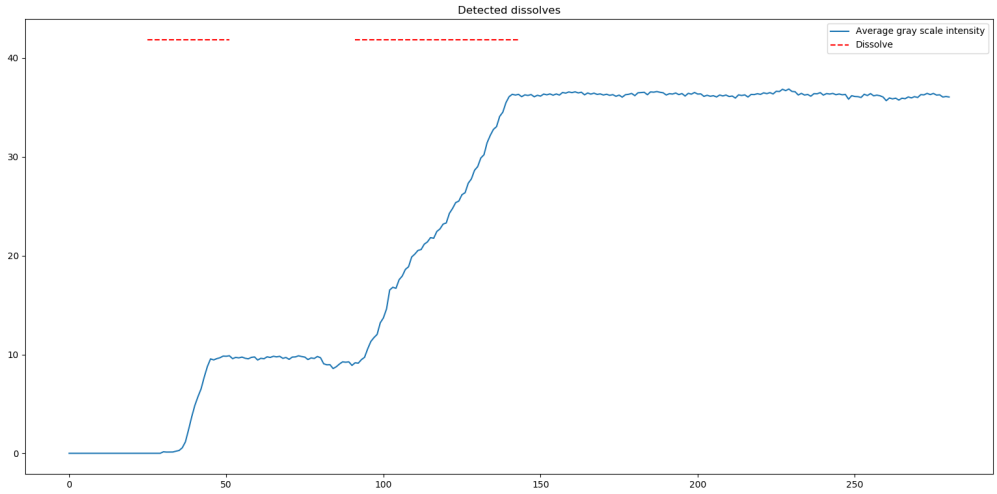


FIGURE 15 – Détection des transitions continues sur l'extrait *Extrait3-Vertigo-Dream_ Scene(320p).m4v*

L'un des principaux inconvénients, dans l'état actuel, de cette méthode, est la séparation stricte de la détection des *cuts* et des transitions continues. En conséquence, un *cut* non détecté lors de la première passe de détection ne sera pas détecté comme *cut*, et encore moins comme transition continue, lors de la seconde passe de détection. Pour résoudre ce problème, il faudrait réaliser une troisième passe de détection sur les maxima qui n'ont pas été retenus en tant que transition continue lors de la seconde passe, et tester leur dissimilitude en tant que *cut*.

Comme pour la méthode précédente, celle-ci risque également de mener à la détection de faux positifs, en particulier lors de plans avec un mouvement continu d'un environnement à un autre (comme un travelling, ou un panorama). Nous observons effectivement une détection erronée d'une transition continue au début du premier travelling horizontal de l'extrait *Extrait4-Entracte-Poursuite_ Corbillard(358p).m4v*[2].

Ainsi, les deux méthodes de détection présentées nous permettent de détecter efficacement - modulo quelques petits ratés - les *cuts* et les transitions continues entres les différents plans d'un extrait vidéo. Une fois les *cuts* et les limites des transitions continues déterminées, il est alors possible de réaliser un découpage en plan de l'extrait vidéo étudié.

5 Question 5

Une fois l'extrait vidéo découpé en plans, nous avons testé deux méthodes de calcul d'image-clé représentative.

La première consiste à calculer l'**entropie**, au sens de Shannon, de chacune des images du plan, et de retenir comme image-clé l'image possédant l'entropie la plus importante. Ainsi, l'image-clé retenue pour chaque plan sera celle contenant le plus d'information à l'échelle de ce plan. Un exemple d'extraction d'image-clé utilisant cette méthode est représenté sur la figure 16.

Cependant, la notion d'information n'est pas forcément liée à la notion de représentativité d'un plan, au sens de la perception qu'un humain peut en avoir. Aussi, la deuxième méthode que nous avons testé consiste à calculer l'image du plan la plus proche de toutes les autres, au sens de la distance euclidienne. L'image-clé retenue pour chaque plan est donc, d'un point de vue pixelique, l'image la plus proche de toutes les images du plan étudié, et donc dans un sens, la plus représentative.

La plus grande différence entre ces deux méthodes de calcul d'image-clé réside dans leur complexité

2. Ce problème a finalement été résolu en augmentant le seuil définissant la taille minimale d'un voisinnage définissant une transition continue

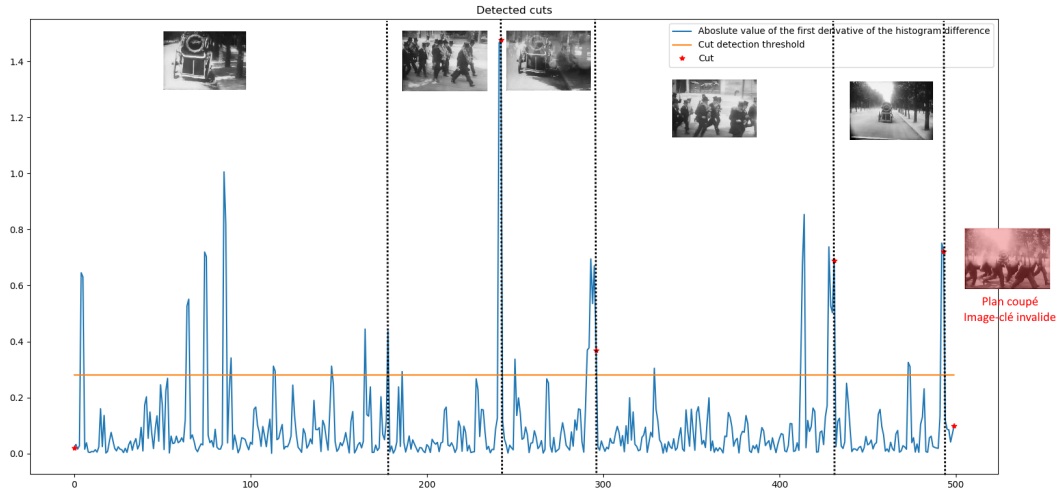


FIGURE 16 – Exemple d'extraction d'images-clés sur l'extrait vidéo *Extrait4-Entracte-Poursuite_ Corbillard(358p).m4v*

de calcul. En effet, si l'entropie d'une image est très simple à calculer, le calcul de l'image la plus proche nécessite l'établissement d'une matrice de distances, et peut donc rapidement devenir très long. Nous avons donc retenu la première méthode en tant que choix par défaut, mais une implémentation fonctionnelle de la seconde méthode est également disponible.

Du point de vue des résultats, ces deux méthodes nous renvoient bien une image contenue dans le plan, et le plus souvent plutôt bien représentative de ce dernier. Toutefois, il arrive que l'image retenue soit assez proche du début ou de la fin du plan, comme c'est le cas par exemple pour l'image-clé retenue pour le troisième plan identifié sur la figure 16. Il nous faut donc sérieusement considérer le cas dans lequel l'image choisie est une image de transition, et donc peu représentative. Pour palier cette éventualité, il serait judicieux d'ajouter une pondération en fonction de la distance aux extrémités du plan dans le calcul de l'image-clé représentative, sous la forme d'une gaussienne centrée sur le milieu du plan par exemple.

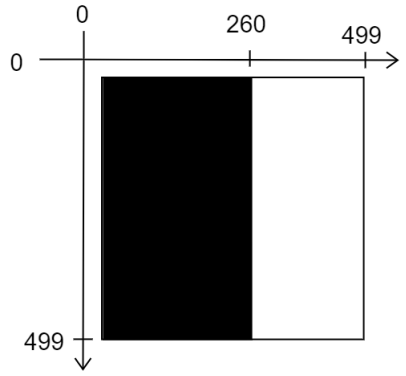
6 Question 6

Comme on l'a vu lors de la question 3, on va utiliser le flot *Dense* plutôt que le flot *Sparse* pour élaborer notre algorithme d'identification de type de plan.

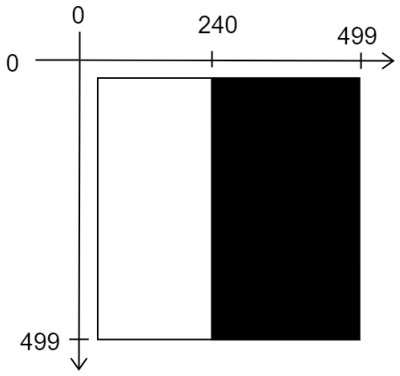
Le principe est le suivant : on réalise une moyenne temporelle de l'histogramme sur toute la durée du plan. On applique ensuite un seuil afin de mettre à la valeur maximale tous les pixels au-dessus de ce seuil et à la valeur minimale tous les autres. Cette deuxième étape permet d'augmenter le poids des pixels non-centraux, qui sont justement ceux qui dessinent la forme caractéristique du plan mais qui ont une valeur moyenne plus faible que les pixels centraux. Par la même étape, on élimine une partie important des aberrations des histogrammes 2D de la vidéo. Nous avons testé plusieurs valeurs de seuil possible, dépendant ou non du nombre de points blancs. Nous avons constaté que dans tous les cas sauf le travelling avant, la valeur de seuil 50 donnait de bons résultats. En revanche, comme on l'a vu sur la figure 7, le motif du travelling avant repose sur une sorte de halo de points avec des valeurs assez faibles, qui disparaissent donc avec un seuillage à 50, donnant simplement une tâche centrale. Nous avons donc décidé de traiter le cas particulier du travelling avant à part.

Une troisième étape consiste à déterminer si le motif de l'histogramme ainsi obtenu est centré ou s'il se trouve dans l'une des moitiés haute, basse, droite ou gauche de l'image. Pour cela, nous avons utilisé des masques, présentés dans la figure 17. La somme des pixels après multiplication par ces masques est comparée à la somme des pixels de l'histogramme complet. Nous avons considéré que si le rapport des deux

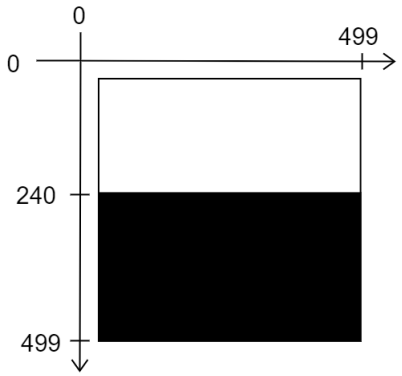
était supérieur à 70% pour l'un des masques, alors le motif caractéristique était contenu dans la partie non masquée du masque correspondant. Ces masques cachent également les pixels les plus proches du centre afin de limiter les erreurs dans le cas d'une tâche centrale un peu décentrée. Si aucun masque ne préserve plus de 70% des pixels blancs, on considère qu'on est face à un motif de tâche centrale, qui peut être soit un plan fixe, soit un zoom, soit une rotation, soit un travelling avant - puisqu'avec notre choix de seuil il correspond aussi à une tâche centrale.



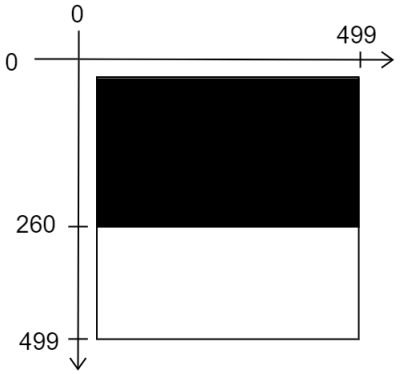
(a) Masque de la partie gauche : si ce masque laisse passer 70% des pixels de l'histogramme, alors on a affaire à un déplacement horizontal vers la gauche, qui peut donc être soit un travelling horizontal soit un PAN.



(b) Masque de la partie droite : si ce masque laisse passer 70% des pixels de l'histogramme, alors on a affaire à un déplacement horizontal vers la droite, qui peut donc être soit un travelling horizontal soit un PAN.



(c) Masque de la partie inférieure : si ce masque laisse passer 70% des pixels de l'histogramme, alors on devrait avoir affaire à un mouvement vers le bas, qui est un TILT.



(d) Masque de la partie inférieure : si ce masque laisse passer 70% des pixels de l'histogramme, alors on a affaire à un mouvement vers le haut, qui est un TILT.

FIGURE 17 – Masques utilisés.

Si les masques 17c ou 17d laissent passer 70% des pixels blancs, alors l'identification du plan est terminée. En revanche, s'il s'agit des masques 17a ou 17b, il faut déterminer s'il s'agit d'un PAN ou d'un travelling. Pour ce faire, on va chercher à estimer si le motif est un cône dont le sommet est au centre, ou non. On commence par réaliser une érosion avec un noyau $3x3$ afin d'éliminer les éventuels pixels isoler. On détermine ensuite la longueur du motif, en cherchant la ligne de l'histogramme avec le plus de pixels blancs. On détermine ensuite la largeur maximale de la forme dans sa moitié droite et dans sa moitié gauche. Si la largeur maximale côté centre est significativement plus petite que celle côté bord (i.e. plus petite que 80% de cette longueur dans notre cas), alors on a affaire à une sorte de cône, caractéristique du PAN. Dans le cas contraire, il s'agit d'un travelling.

Enfin, si aucun des masques ne laisse passer 70% des pixels blancs, alors on a affaire à une tâche centrale,

qu'on va essayer de caractériser plus précisément. On commence par vérifier si cette tâche correspond à une rotation : ce test est en effet assez simple, puisque dans le cas d'une rotation la tâche blanche moyenne est très large et recouvre - selon nos observations - plus de 10% de l'image. Dans le cas contraire, on vérifie si l'histogramme peut correspondre à un travelling avant. Ce test est relativement simple mais implique de retraiter l'histogramme moyen en lui appliquant un seuil différent. Nous avons utilisé un seuillage médian, qui permet bien de faire apparaître une large tâche blanche dans le cas de ce plan. Il suffit ensuite de vérifier si on obtient bien une telle tâche, i.e. recouvrant - là aussi selon nos observations - plus de 20% de l'image. Dans le cas contraire, on a affaire à un plan fixe ou un zoom, dont la différence à l'oeil nu n'est pas évidente sur les histogrammes de nos vidéos de test. La différence la plus visible est que la tâche du zoom est un peu plus large que celle du plan fixe. Cela dit, ce manque de différences peut s'expliquer par le fait que notre vidéo de zoom soit très lente, ce qui explique que les normes des vecteurs du flot optique soient très petites et donc qu'on n'observe qu'une petite tâche centrale sur l'histogramme. Avec nos vidéos de test, le plan fixe donnait une concentration de pixels blancs d'environ 0.0002, contre 0.0004 pour le zoom : nous avons donc considéré qu'au delà d'une concentration de pixels blancs de 0.0003 on avait affaire à un zoom, et en-deçà à un plan fixe. Toutefois, il est peu probable que cette estimation fonctionne de manière générale.

Cette méthode d'identification de plan nous a permis d'identifier automatique nos vidéos test montrant les différents plans avec peu de mouvement dans l'image autre que le déplacement de la caméra. Toutefois, il n'est pas sûre que cette méthode fonctionne bien pour des plans dans lesquels on aurait également des objets en mouvement. Il est possible que la reconnaissance de déplacement horizontale, de TILT et de rotation fonctionne encore, puisqu'elle implique une analyse finalement assez grossière de l'histogramme, qui devrait rester peu sensible au bruit. En revanche, différencier un PAN d'un travelling horizontal ou, plus complexe, différencier les plans fixes, travellings avant et zooms a de plus forte chance de donner de mauvais résultats. C'est exactement ce qu'on observe sur les premiers plans de *Extrait1-Cosmos_ Laundromat1(340p).m4v* : il y a confusion entre plans fixes, zooms et travellings avant.