

ROB317 TP2 : Découpage et indexation de vidéos

Dajing GU & Zheyi SHEN

November 2020

1 Introduction

Dans ce TP, on s'est familiarisé avec le traitement de vidéos sur OpenCV, en particulier l'estimation du champ de vitesses apparentes (flot optique), et on a réfléchi à la problématique de l'indexation automatique de vidéos. On a détecté les changements de plans dans la vidéo, trouvé pour chaque plan l'image la plus représentative et associé à chaque plan une description textuelle.

2 Histogramme de couleur

2.1 Vidéos de codage Yuv

Yuv est un espace colorimétrique en trois composantes. Y est une composante de luminance (luma), qui représente le niveau de gris, tandis que (u, v) sont deux composantes de la chrominance (chroma), qui sont utilisées pour décrire la saturation de couleur. u et v représentent respectivement le contraste Bleu/Jaune et le contraste Rouge/Cyan.

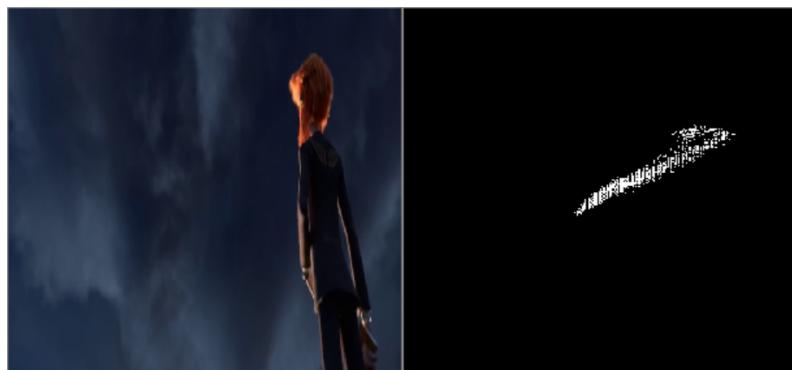


Figure 2.1: Histogramme u-v de vidéo

En comparaison avec RGB , Yuv occupe moins d'espace pour conserver l'information des couleurs, en plus, il peut satisfaire en même temps aux besoins de la télévision noir et blanc et de la télévision en couleur. Par conséquent, beaucoup de fabricants choisissent de

convertir l'espace colorimétrique *RGB* en *Yuv* afin de maintenir la compatibilité, puis de revenir au format *RGB* si on a besoin d'afficher des graphiques sur l'écran du ordinateur.

En utilisant *cv2.VideoCapture* et *cv2.cvtColor*, on a réussi à lire chaque image de la vidéo et on les a transformées en l'espace *Yuv*. Ensuite, à l'aide de la fonction *cv2.calHist*, on a tracé l'histogramme 2d correspondant à la probabilité jointe des composantes chromatiques (*u*, *v*) du codage *Yuv*, qui est présenté dans la figure 2.1.

On trouve qu'il y a des petites variations continues sur l'histogramme quand il n'y a pas de changement de plan. Mais il y a un changement soudaine et discontinue sur l'histogramme quand un changement de plan se passe. Pour avoir une connaissance plus claire, on a utilisé *cv2.compareHist* afin de calculer la corrélation entre les histogrammes des deux trames successives, qui est présentée dans la figure 2.2.

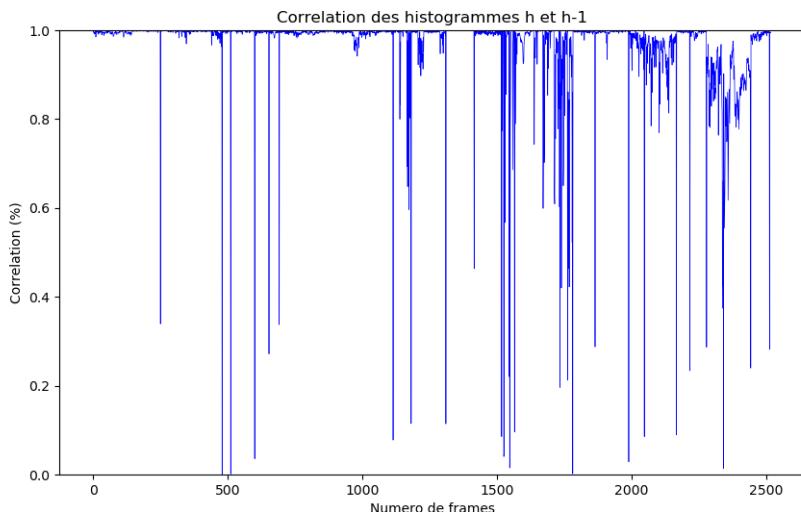


Figure 2.2: Corrélation entre les histogramme2d des deux trames successives

Les réductions soudaines de la corrélation dans la figure 2.2 peuvent indiquer des changements de plans dans la vidéo. Il est nécessaire de déterminer un seuil afin de filtrer des réductions soudaines causées par des variations dans la scène plutôt que par des changements de plans.

2.2 Vidéos monochromes

Pour une vidéo monochrome, il n'y a plus de canal *u* et de canal *v*, donc l'affichage de histogramme 2d n'est plus possible.

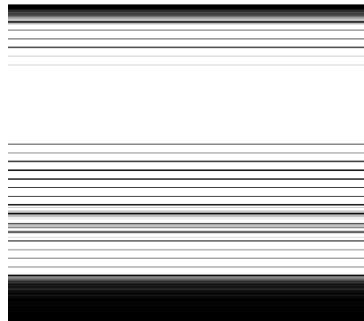


Figure 2.3: Histogramme1d de vidéo gris

Comme présenté dans la figure 2.3, on a choisi de tracer un histogramme 1d. À l'aide de `cv2.cvtColor`, on a réussi à transformer tous les trames de la vidéo en image grise. En suivant les mêmes étapes dans la sous-section précédente, la figure de la corrélation entre les histogrammes des deux trames successives est présentée dans la figure 2.4.

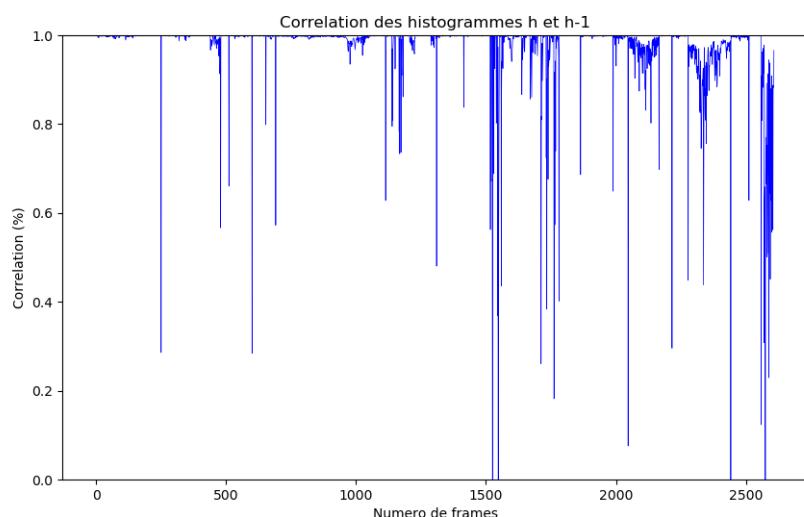


Figure 2.4: Corrélation entre les histogrammes 1d des 2 trames successives

3 Flot optique et histogramme de vitesses

Toutes les choses dans le monde sont en mouvement. Les vitesses différentes et les directions variées constituent un champ de mouvement. La projection de mouvement de l'objet sur l'image est le mouvement des pixels, dont la vitesse instantanée est le flot optique.

La méthode de flot optique est une méthode qui permet de déduire la vitesse et la direction du mouvement de l'objet en détectant les changements d'intensité des pixels de l'images au fil du temps. Les informations des mouvement peut être calculées en analysant la correspondance entre la trame précédente et la trame actuelle.

Il y a deux types de méthode de flot optique. Le flot optique clairsemé (sparse optical flow) ne concerne que des points d'intérêt de l'image, mais le flot optique dense (dense optical flow) calcule le décalage de tous les points de l'image. Par rapport au premier, le coût de calcul du second est beaucoup plus élevé. Et dans cette section, on analyse une des méthodes denses : Farnebäck.

3.1 Hypothèses basique de méthode de flot optique

Avant d'analyser plus précisément le principe, on représente les hypothèses de méthode de flot optique. En extrayant chaque trame de vidéo, on peut obtenir une séquence d'images, que l'on note $I(x, y, t)$.

En supposant que la luminance d'un point image est indépendante du temps et en notant $\mathbf{x} = [x, y]$, on a:

$$(3.1) \quad \frac{dI(\mathbf{x}, t)}{dt} = \frac{\partial I}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial t} + \frac{\partial I}{\partial t} = 0$$

En utilisant le formule de Taylor, on a :

$$I(\mathbf{x}, t) = I(\mathbf{x} + \Delta \mathbf{x}, t + \Delta t) \approx I(\mathbf{x}, t) + \frac{\partial I}{\partial \mathbf{x}} \Delta \mathbf{x} + \frac{\partial I}{\partial t} \Delta t$$

Donc,

$$\frac{\partial I}{\partial \mathbf{x}} \Delta \mathbf{x} + \frac{\partial I}{\partial t} \Delta t = 0$$

Après la dérivation temporelle, on peut déduire que

$$\frac{\partial I}{\partial \mathbf{x}} \frac{\Delta \mathbf{x}}{\Delta t} + \frac{\partial I}{\partial t} = 0$$

Car le temps entre deux trames est vraiment petit, on peut faire une approximation $\frac{\Delta \mathbf{x}}{\Delta t} \approx \frac{\partial \mathbf{x}}{\partial t} = [\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t}]$, et on a :

$$\frac{\partial I}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial I}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial I}{\partial t} = 0$$

En supposant que $\frac{\partial \mathbf{x}}{\partial t} = [\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t}] = [u, v] = \mathbf{d}$, on peux facilement déduire que :

$$(3.2) \quad \begin{aligned} I_x u + I_y v + I_t &= 0 \\ \begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} &= -I_t \end{aligned}$$

Il y a deux inconnus dans une seule équation, donc on ne peut pas la résoudre directement. Afin d'obtenir la valeur de u et de v , des méthodes de flot optique sont créées.

3.2 Principe de Farnebäck

3.2.1 La théorie de la méthode

Farnebäck est une méthode basée sur le gradient. Le gradient d'image et le flot optique local sont supposés constant dans ce méthode.

En supposant que les images sont grises, le niveau de gris du pixel d'image peut être considéré comme une fonction de variables bidimensionnelle $f(x, y)$. En créant une coordonnée locale (pas sur l'image entière), on peut obtenir que :

$$(3.3) \quad \begin{aligned} f(x, y) &\approx r_1 + r_2x + r_3y + r_4x^2 + r_5y^2 + r_6xy \\ &= (x \ y)^T \begin{pmatrix} r_4 & r_6/2 \\ r_6/2 & r_5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} r_2 \\ r_3 \end{pmatrix}^T \begin{pmatrix} x \\ y \end{pmatrix} + r_1 \\ &= \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x} + c, \end{aligned}$$

avec \mathbf{x} un vecteur colonne à deux dimensions, \mathbf{A} une matrice symétrique 2×2 et \mathbf{b} une matrice 2×1 .

Si un pixel déplace \mathbf{d} , l'équation 3.3 devient :

$$\begin{aligned} f_2(\mathbf{x}) &= f_1(\mathbf{x} - \mathbf{d}) = (\mathbf{x} - \mathbf{d})^T \mathbf{A}_1 (\mathbf{x} - \mathbf{d}) + \mathbf{b}_1^T (\mathbf{x} - \mathbf{d}) + c_1 \\ &= \mathbf{x}^T \mathbf{A}_1 \mathbf{x} + (\mathbf{b}_1 - 2\mathbf{A}_1 \mathbf{d})^T \mathbf{x} + \mathbf{d}^T \mathbf{A}_1 \mathbf{d} - \mathbf{b}_1^T \mathbf{d} + c_1 \\ &= \mathbf{x}^T \mathbf{A}_2 \mathbf{x} + \mathbf{b}_2^T \mathbf{x} + c_2 \end{aligned}$$

avec

$$\begin{aligned} \mathbf{A}_2 &= \mathbf{A}_1 \\ \mathbf{b}_2 &= \mathbf{b}_1 - 2\mathbf{A}_1 \mathbf{d} \\ c_2 &= \mathbf{d}^T \mathbf{A}_1 \mathbf{d} - \mathbf{b}_1^T \mathbf{d} + c_1 \end{aligned}$$

Si \mathbf{A}_1 est inversible, on peut calculer facilement le flot optique:

$$(3.4) \quad \mathbf{d} = -\frac{1}{2} \mathbf{A}_1^{-1} (\mathbf{b}_2 - \mathbf{b}_1)$$

3.2.2 Approximation plus pratique

Selon la théorie, il faut que $\mathbf{A}_1 = \mathbf{A}_2$. Mais c'est pas toujours vrai dans le cas réel, donc on a fait des approximations. En supposant que $\mathbf{A}(\mathbf{x}) = \frac{1}{2}(\mathbf{A}_1(\mathbf{x}) + \mathbf{A}_2(\mathbf{x}))$ et $\Delta\mathbf{b}(\mathbf{x}) = -\frac{1}{2}(\mathbf{b}_2(\mathbf{x}) - \mathbf{b}_1(\mathbf{x}))$, on peut déduire que:

$$\mathbf{A}(\mathbf{x})\mathbf{d}(\mathbf{x}) = \Delta\mathbf{b}(\mathbf{x})$$

Donc on a finalement:

$$(3.5) \quad \mathbf{d} = (\mathbf{A}^T \mathbf{A})^{-1} (\mathbf{A}^T \Delta\mathbf{b})$$

En fait il y a trop de bruit dans le résultat de l'équation 3.5. Afin de l'éliminer, on a utilisé la pondération du voisinage des pixels d'intérêt, dont la formule est présentée ci-dessous:

$$(3.6) \quad \mathbf{d}(\mathbf{x}) = \left(\sum w \mathbf{A}^T \mathbf{A} \right)^{-1} \sum w \mathbf{A}^T \Delta\mathbf{b}$$

où $w(\Delta\mathbf{x})$ est une fonction de poids pour les points voisinsages.

3.3 Fonction de *calOpticalFlowFarneback*

Dans le script *Dense-Optical-Flow.py*, on utilise la fonction *cv2.calcOpticalFlowFarneback*. Il calcule le flot optique dense de la vidéo à l'aide de l'algorithme de Gunnar Farnebäck. Les paramètres principaux de cette fonction sont décrits ci-dessous :

- **prev** : Première image d'entrée monocanal 8 bits.
- **next** : Deuxième image d'entrée de même taille et du même type que la précédente.
- **pyr_scale** : Un paramètre spécifiant l'échelle de l'image (< 1) pour construire des pyramides pour chaque image. Ce paramètre est généralement 0.5, ce qui signifie une pyramide classique, où chaque couche suivante est deux fois plus petite que la précédente.
- **levels** : Nombre de niveaux de pyramide incluant l'image initiale. **levels=1** signifie qu'aucune couche supplémentaire n'est créée et que seules les images originales sont utilisées.
- **winsize** : La taille de la fenêtre de valeur moyenne. Plus la taille de la fenêtre est grande, plus l'algorithme est robuste pour le bruit, et il peut donner plus de chances de détection de mouvement rapide, mais il produisent un champ de mouvement plus flou.
- **iterations** : Nombre d'itérations effectuées par l'algorithme à chaque niveau de la pyramide.
- **ploy_n** : Nombre de pixels adjacents utilisé pour trouver l'expansion polynomiale pour chaque pixel. Des valeurs plus élevées signifient que l'image sera approximée avec des surfaces plus lisses, donnant un algorithme plus robuste et un champ de mouvement plus flou, généralement **poly_n = 5 ou 7**.
- **ploy_sigma** : L'écart type de la Gaussienne. Pour **poly_n = 5**, vous pouvez définir **poly_sigma = 1.1**, pour **poly_n = 7**, une bonne valeur serait **poly_sigma = 1.5**.
- **flags** : Indicateurs d'opération qui peuvent être une combinaison des éléments suivants :
 - **OPTFLOW_USE_INITIAL_FLOW** : utilise le flot d'entrée comme une approximation de flot initial.
 - **OPTFLOW_FARNEBACK_GAUSSIAN** : utilise le filtre Gaussien *winsize × winsize* au lieu d'un filtre boîte de même taille pour l'estimation de flot optique.

Le résultat de cette fonction **flow** est sous la forme d'image de même taille avec **prev** et **next**. Il montre un flot optique pour chaque pixel de **prev** en utilisant l'algorithme de Farnebäck tel que :

$$\mathbf{prev}(y, x) \sim \mathbf{next}(y + \mathbf{flow}(y, x)[1], x + \mathbf{flow}(y, x)[0])$$

3.4 L'histogramme 2D des composants (V_x, V_y)

Comme présenté dans la sous-section précédente, la fonction `cv2.calOpticalFlowFarneback` nous rend les composantes horizontales V_x et les composantes verticales V_y du mouvement optique. En utilisant la même méthode dans la section 2, on a réussi à 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 (V_x, V_y) du flot optique.

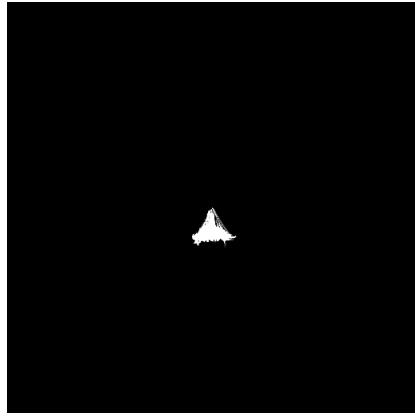


Figure 3.1: Histogramme 2D (V_x, V_y) de vidéo

Un exemple de l'histogramme 2d est présenté dans la figure 3.1. L'histogramme ici est similaire à une explosion étendant vers toutes les directions, ce qui signifie que le mouvement de l'objet dans cette vidéo est vers toutes les directions.

Ensuite on va analyser la forme de l'histogramme2d (V_x, V_y) pour des vidéos subissant des mouvement différents:

- Plan fixe
- Panoramique horizontal (Rotation OY: Pan)
- Panoramique vertical (Rotation OX: Tilt)
- Rotation OZ (Roll)
- Travelling horizontal (travelling OX)
- Travelling avant (travelling OZ)
- Zoom avant

3.4.1 Analyse des formes différentes de l'histogramme

Comme présenté dans la figure 2(a), l'histogramme de plan fixe est un point au centrale car il n'y a aucun mouvement horizontal ou vertical.

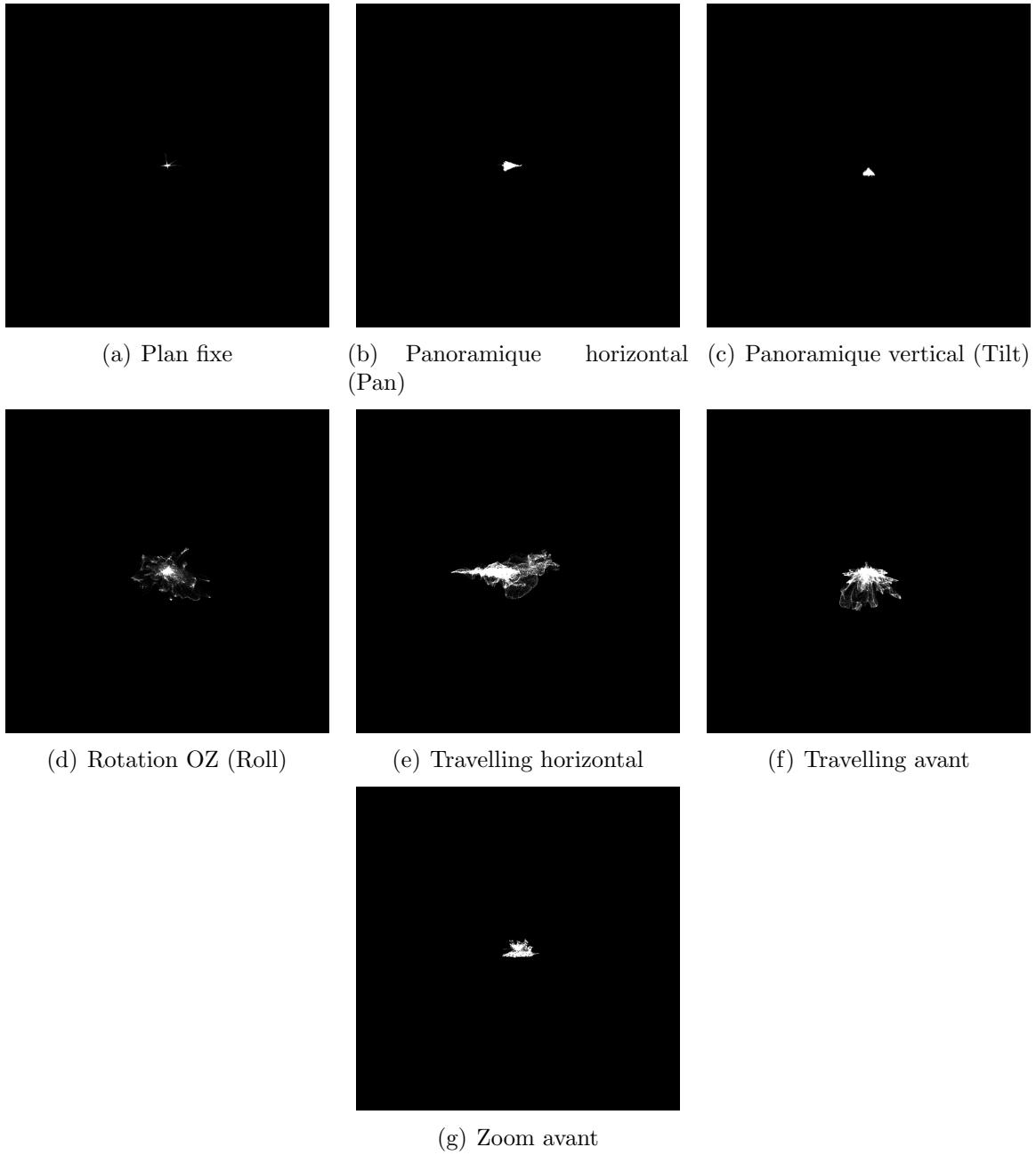


Figure 3.2: Histogramme 2d $V_x - V_y$ des plans de types différents

L'histogramme de panoramique horizontal, présenté dans la figure 2(b), est un cône d'axe horizontal (ici vers la droite). Le cône est au centre de l'histogramme. C'est raisonnable car dans le mouvement de panoramique horizontal (vers la droite), la moitié du pixel en haut se déplace vers la gauche et le haut, la moitié du pixel à droite se déplace vers la gauche et le bas, cela nous donne une forme de cône vers la droite.

L'histogramme de panoramique vertical, présenté dans la figure 2(c), est un cône d'axe vertical (vers le haut ici). Le cône est au centre de l'histogramme. C'est raisonnable car dans le mouvement de panoramique vertical (vers le haut), la moitié du pixel à gauche se déplace vers la gauche et le bas, la moitié du pixel à droite se déplace vers la droite et le

bas, cela nous donne une forme de cône vers le haut.

L'histogramme de Roll, présenté dans la figure 2(d) est un gros nuage de points autour du centre de l'histogramme. Pour un pixel dans l'image, plus loin il est du centre, plus grande sa vitesse de retournement. Puisque le déplacement se fait vers toutes les directions, l'histogramme est aussi quasi-circulaire.

Le mouvement Travelling horizontal de la caméra nous rend un histogramme présenté dans la figure 2(e), avec une ligne horizontale comme l'axe des points de nuages. La cohérence entre le mouvement de caméra et le mouvement du pixel a été vérifiée dans cet histogramme.

Le mouvement Travelling avant de la caméra nous rend un histogramme similaire à celui de Roll, qui est présenté dans la figure 2(f). C'est un nuage car tous les pixels se déplacent vers l'extérieur, par exemple, les pixels à gauche se déplacent vers la gauche, les pixels en haut se déplacent vers le haut, etc. Ici, comme la couleur du ciel est plus homogène, il est difficile de reconnaître son mouvement, de sorte que dans l'histogramme, la partie située en-dessous du centre a beaucoup de points blanc que la partie située au-dessus du centre.

Comme le cas de Travelling avant, l'histogramme de Zoom avant est aussi un histogramme de nuages de points, qui est présenté dans la figure 2(g). L'explication de la création de cet histogramme est pareille, mais la rayon de partie blanche est plus courte qu'avant. C'est parce que la vitesse de Zoom avant dans la vidéo est plus petite.

En fonction de différents types d'histogrammes mentionnés ci-dessus, on peut facilement identifier les plans de types différents.

4 Découpage et Indexation

La découpage et indexation des vidéos est un travail vraiment important dans le domaine de traitement de vidéo. Dans cette section, on va présenter le processus de découper une vidéo en plans (section 4.1), trouver l'image clef de ce plan (section 4.2) et finalement identifier le type du plan (section 4.3).

4.1 Une technique de découpage en plans

D'après la section 2, on sait que pour une vidéo en couleur, on peut détecter le changement de plan en comparant les histogrammes u-v des deux trames successives. Si la corrélation réduit soudainement, il est très probable qu'il y a un changement de plan. On peut utiliser ce principe pour découper le vidéo en plans.

La figure 4.1 représente le résultat de découpage de la vidéo *Extrait5-Matrix-Helicopter-Scene(280p).m4v*. On choisit un seuil de corrélation, si la corrélation est inférieur à ce seuil, la trame est considérée comme un changement de plan ainsi que un des points de découpage. Pour l'extrait 5, tous les changements de plan sont bien détectés avec le seuil égale à 0.6.

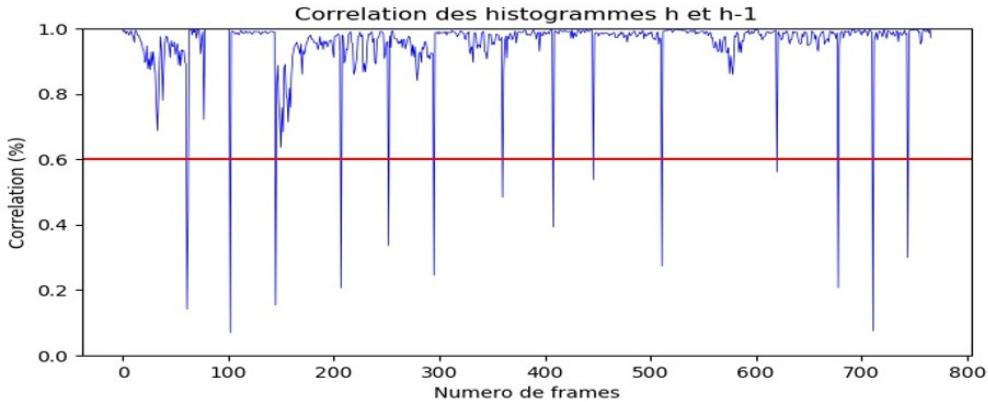


Figure 4.1: Résultats de découpage de l'extrait 5

La figure 4.2 représente le résultat de découpage de la vidéo *Extrait1-Cosmos-Laundromat1 (340p).m4v*. On choisit un seuil de corrélation égale 0.5.

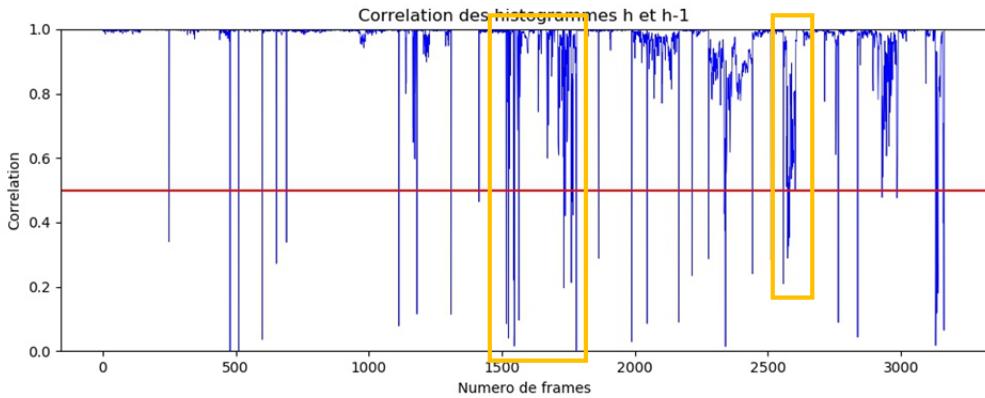


Figure 4.2: Résultats de découpage de l'extrait 1

La plupart de changement de plan peut être détectée, mais il y a encore des détections fausse. Par exemple, les fautes indiquées par les deux rectangles jaune dans la figure 4.2. Dans ces deux cas, il y a seulement 1 ou 2 changement de plan mais plusieurs trames sont notées comme le point de découpage car leurs corrélations réduisent de manière abrupte. Dans la première rectanglles, les détections fausses sont causées par la "foudre" dans le vidéo où l'image devient soudainement plus claire, puis s'assombrit immédiatement. Dans le deuxième cas, L'arrière-plan coloré de l'image tourne rapidement, de sorte que l'histogramme u-v change également rapidement.

Alors, on voit que cette méthode est très sensible aux changements de couleur. Pour une vidéo comme *Extrait3-Vertigo-Dream.Scene(320p).m4v*, il ne peut pas être découpé en utilisant seulement l'histogramme u-v. Pour résoudre ce problème, Il faut ajouter une méthode qui permet de caractériser plutôt les formes et les contours des objets sur l'image et est moins affectée par les variations de couleur. Par exemple, on peut essayer des descripteurs comme ORB, KAZE, etc. Et puis, calculer la similarité des points caractéristiques entre deux images successives et noter où la similarité diminue de manière abrupte.

Ici, on a essayé une autre méthode, calculer la corrélation de l'histogramme de gradient orienté (HOG). Le descripteur HOG peut décrire l'apparence et la forme locale d'un objet dans une image par la distribution de l'intensité du gradient ou la direction des contours. Donc, c'est un histogramme moins affecté par la couleur. Comme la méthode précédente, on calcule la corrélation de HOG des deux trames successives. Le résultats est représenté dans la figure 4.3.

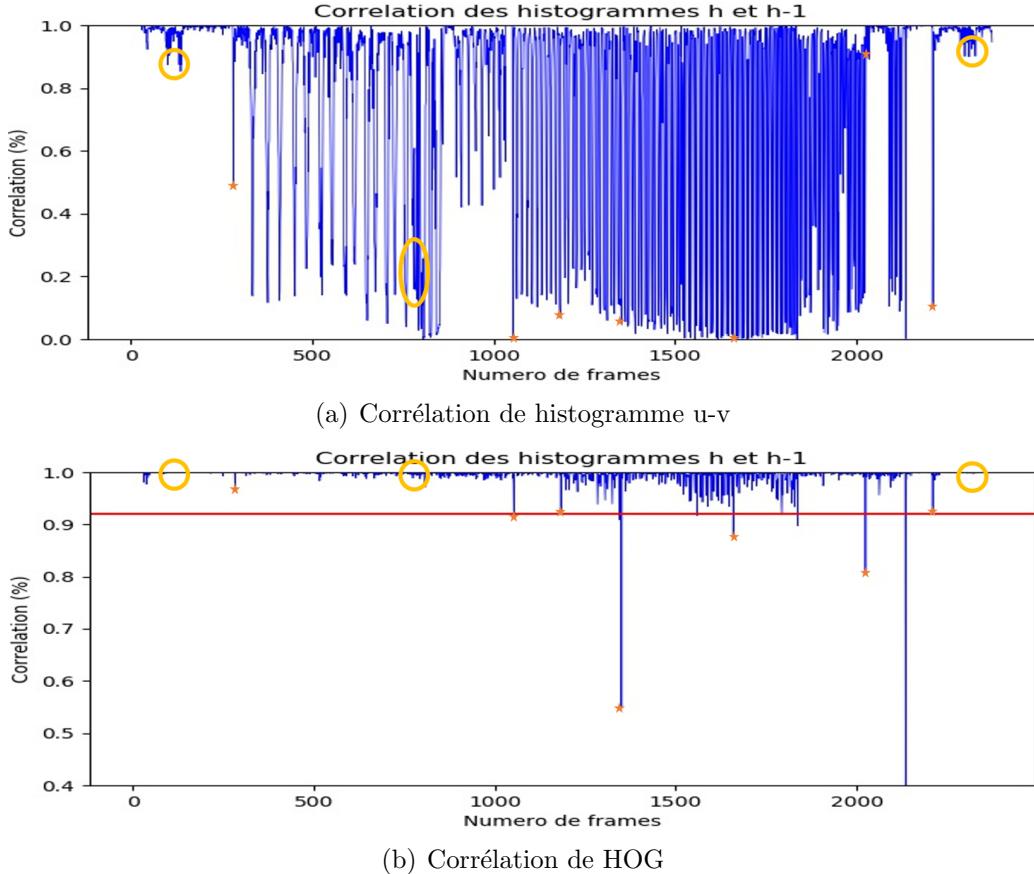


Figure 4.3: Comparaison de deux méthodes de découpages pour l'extrait 3

Les points de changement de plan sont marqués sur la figure par les étoiles oranges ou les contours Jaunes. D'après la figure 4.3, Par rapport à la méthode de histogramme u-v, la méthode de HOG peut filtrer en certain mesure les erreurs de détection causées par le changement de couleur de la scène, mais en même temps, il a également raté quelques points de découpage ou ajouté des points fausse. Les cercles jaunes signifient que il y a un changement de plan mais la transition est très lentement. L'histogramme u-v et HOG ne peuvent pas détecter ce type de changement.

Pour les vidéos monochromes, on utilise l'histogramme de niveau de gris au lieu de l'histogramme u-v. Comme présenté dans la figure 2.3, pour l'extrait 1, la corrélation de histogramme 1d peut aussi indiquer le changement de plan.

Cependant, le résultat de découpage est largement affecté par la qualité vidéo. Par exemple, il est difficile de découper en plans les vidéos *Extrait2-Man With A Movie Camera.m4v* et *Extrait4-Entracte-Poursuite_Corbillard(358p).m4v*. Ces deux vidéos ont beaucoup de

bruit et ils "scintillent" souvent, c'est-à-dire, le vidéo est parfois clair parfois sombre.

La figure 4.4 représente les résultats des deux méthodes pour l'extrait 2, les changement réel de plan sont marqués par les étoile oranges.

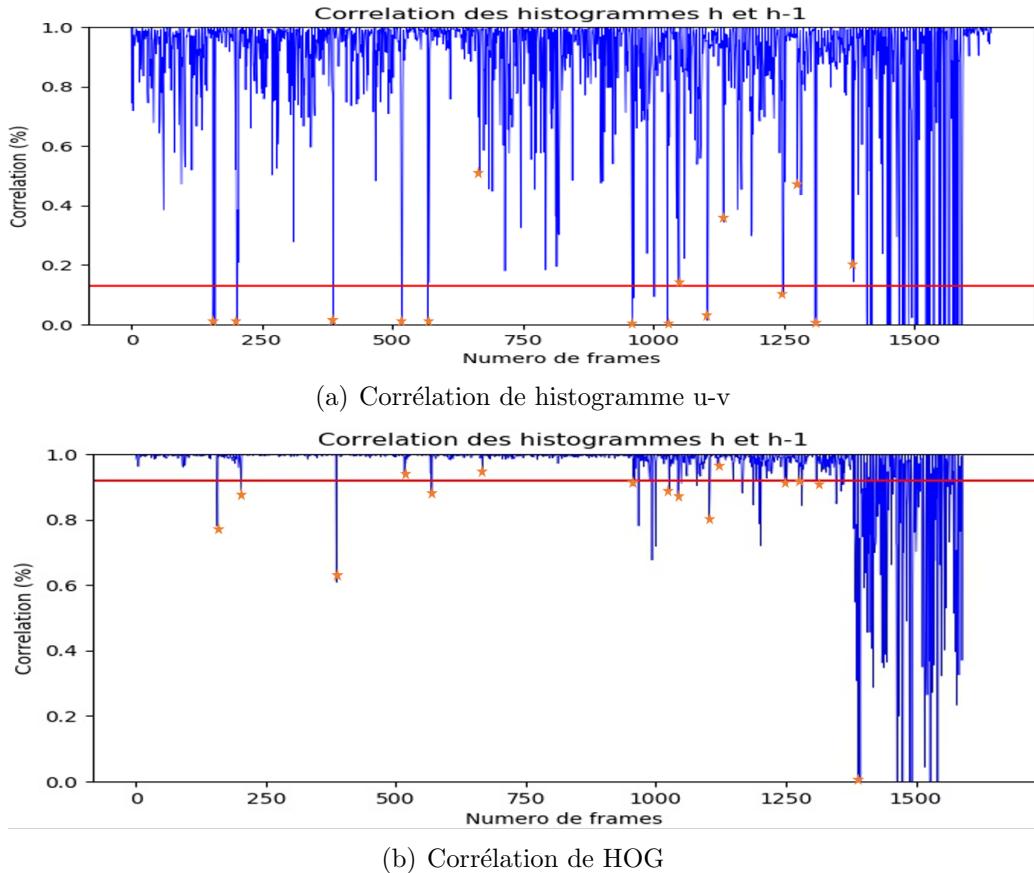


Figure 4.4: Comparaison de deux méthodes de découpages pour l'extrait 2

A cause de la mauvaise qualité de la vidéo, le seuil de la première méthode doit être abaissé pour filtrer le bruit. Dans le dernier plan de la vidéo, aucune des deux méthodes ne donne de bons résultats. C'est parce qu'il y a souvent les trames totalement noires qui influencent les valeurs Yuv et les gradients de l'image.

Pour l'extrait 4, la qualité est pire. La corrélation de l'histogramme de niveau de gris change toujours de manière abrupte et la corrélation de HOG détecte beaucoup de trames qui n'est pas un changement de plan. La raison est que l'extrait 4 n'est pas "lisse", c'est-à-dire chaque action et chaque transition de plan a un retard important. L'algorithme ne peut pas distinguer c'est quel type de retard, cela cause la difficulté pour le découpage.

Pour conclure, calculer la corrélation de l'histogramme u-v ou en niveau de gris est une bonne technique pour découper une vidéo en plans, mais c'est sensible au changement de couleur. L'ajoute de la calculation de la corrélation de HOG aide à filtrer des perturbations de teinte de plan. Cependant, tous les deux méthodes ne peuvent pas obtenir un résultat raisonnable pour une vidéo de mauvaise qualité. De plus, il est aussi difficile de détecter une transition lente.

4.2 Une mesure d'extraction d'image-clef

Dans la section précédente, on a réussi à découper la vidéo en plans, pour avoir une connaissance plus claire des plans découpés, on a besoin d'extraire une image-clef de chaque plan.

Avant de présenter notre méthode d'extraire l'image-clef, on fixe tout d'abord la définition de l'image-clef. À notre avis, l'image-clef est une trame de vidéo qui est capable de bien présenter les caractéristiques du plan, par exemple, le type du plan, la variation des pixels, etc. L'analyse de l'image-clef peut atténuer la difficulté d'étudier les plans de la vidéo.

On a pensé à 3 méthodes pour extraire l'image clef: *la méthode de points d'intérêt* et *la méthode de moyenne*.

- *La méthode de milieu*: Dans cette méthode, on va prendre l'image du milieu $I_{(t_{min}+t_{max})/2}$, mais ce n'est pas forcément l'image la plus représentative.
- *La méthode de points d'intérêt*: Dans cette méthode, on va calculer le nombre des points d'intérêt pour chaque trame de chaque plan. La trame dont le nombre des points d'intérêt est le plus grand vont être choisie comme l'image-clef de ce plan.
- *La méthode de moyenne*: Dans cette méthode, on va faire une comparaison entre toutes les trames dans un plan avec la trame moyenne, dont $I(x, y) = \frac{1}{N} * \sum_{plan} I(x, y)$. La trame qui est la plus proche que l'image moyenne va être choisie comme l'image-clef du plan. Mais quand le plan change très rapidement, le choix de l'image-clef sera compliqué car le calcul de l'image moyenne n'est pas précis.

On se concentre seulement sur les 2 premières méthode car il y a trop de bruit dans la troisième méthode qui va influencer le résultat. Et on a réussi de les implémenter à l'aide de *cv2.goodFeaturesToTrack* et de *cv2.calcOpticalFlowPyrLK* dans la script *Sparse-Optical-Flow*, on peut calculer facilement le nombre de points d'intérêt de chaque trame des plans dans une vidéo.

```
Démarrage du programme

Index decoupage: [63, 104, 147, 209, 254, 297, 362, 410, 513, 680, 713, 746]
nombre de plan: 13

Index image-clef: [31, 83, 125, 178, 231, 275, 329, 386, 461, 596, 696, 729, 757]
Index image-caractéristique: [44, 86, 121, 199, 246, 272, 342, 373, 411, 568, 706, 739, 749]

Toutes les images-clefs sont sauvegardées !!

Process finished with exit code 0
```

Figure 4.5: Résultat de notre script

Afin d'illustrer l'effet de notre méthode, on a écrit une script, dans laquelle on peut découper la vidéo en plans et trouver les images-clef de chaque plan automatiquement. En plus, notre script peut aussi sauvegarder les images-clef automatiquement.

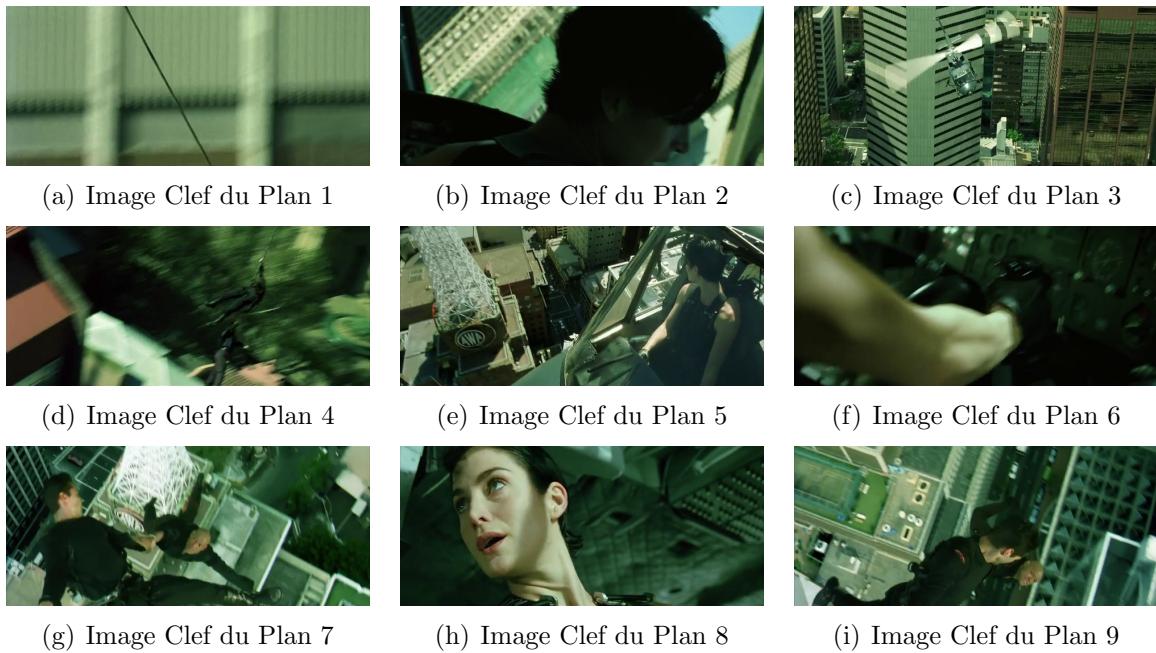


Figure 4.6: Images-clef obtenues par *la méthode de milieu* de la vidéo 5 (il y a 13 plans mais on en présente seulement 9 ici)

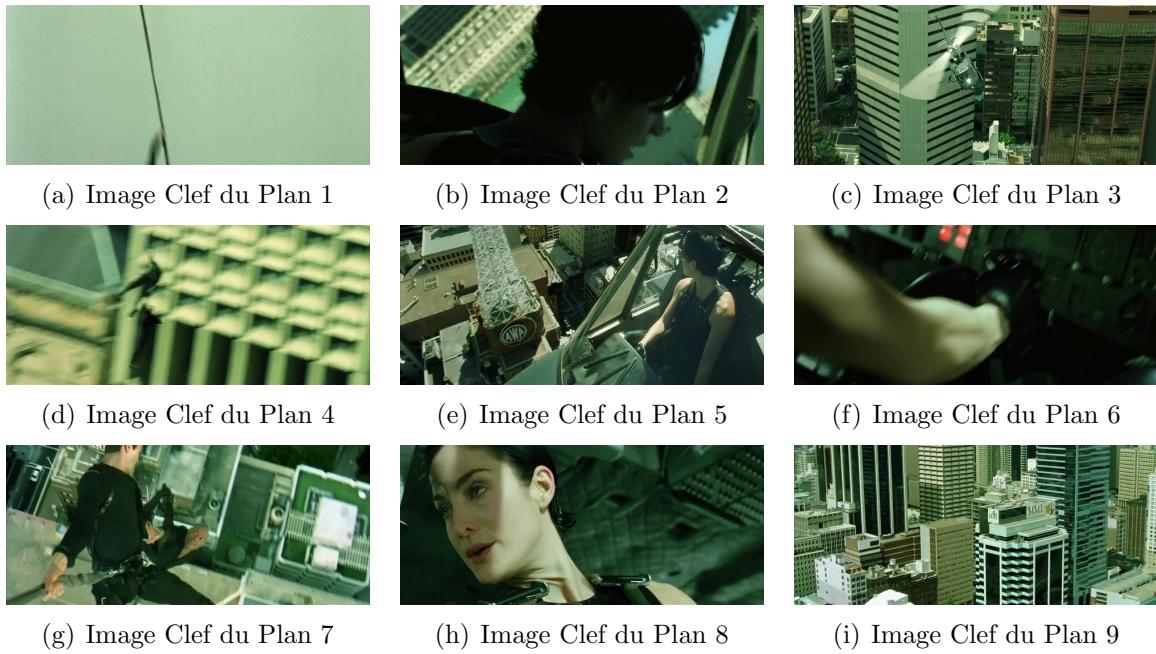


Figure 4.7: Images-clef obtenues par *la méthode de points d'intérêt* de la vidéo 5 (il y a 13 plans mais on en présente seulement 9 ici)

Comme présenté dans la figure 4.6, la figure 4.7 et la figure 4.5, les résultats sont satisfaisants étant donnée que les images-clef choisies représentent bien les scènes de la vidéo. Même si le premier plan qui change très rapidement, pour les autres plans, les images-clef

correspondent très bien aux autres plans de la vidéo. Et les images-clef obtenues par les deux méthodes sont un peu différentes.

4.3 Une technique d'identification de plan

Après avoir découpé le vidéo en plans, on doit proposer une technique pour identifier chaque plan. Comme déjà présenté dans la section 3.4, le type de plan peut être identifié par l'histogramme 2D des composantes (V_x, V_y) du flot optique. Dans cette section, on va continuer d'utiliser cette méthode. Le principe est le suivant : on calcule une moyenne temporelle de l'histogramme des composantes (V_x, V_y) du flot optique sur toute la durée du plan. On la met ensuite dans un algorithme pour obtenir un résultat de classification pour ce plan.

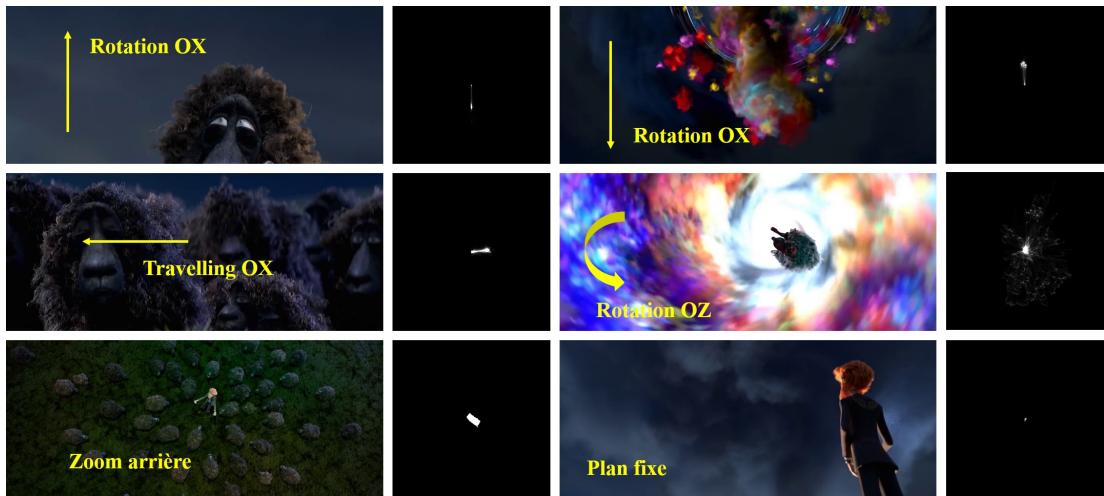


Figure 4.8: Présentation de quelques plans et les histogrammes associés

D'un point de vue mathématique, différents types de plans auront des effets correspondants sur les valeurs des différentes parties de l'histogramme. Par exemple, quand la caméra se déplace vers la droite, l'orientation de l'histogramme sera la gauche, donc la somme de partie gauche sera plus grande que la somme de partie de droite. De même façon, si le mouvement de la caméra est vers le haut, l'orientation de l'histogramme sera vers le bas, donc la somme de partie gauche sera plus grande que la somme de partie de droite.

Cependant, selon la variation de l'histogramme, il est difficile de distinguer Rotation vertical et Travelling vertical. De plus, pour un plan de rotation OZ ou un plan de zoom avant, il est difficile d'analyser la variation des valeurs dans l'histogramme. Donc, dans cette section, on propose de utiliser une méthode de *Machine Learning* pour faire la classification.

4.3.1 Méthode de k-plus-proches-voisins

Supposons que pour chaque plan, on a un histogramme moyen de composantes (V_x, V_y) du flot optique. L'objectif est de identifier le type de plan, c'est bien un problème de classification. On utilise la méthode de k-ppv (k-NN en anglais).

Chaque pixel de l'histogramme peut être considéré comme une caractéristique. Avant d'appliquer le classificateur k-NN, on doit réduire la dimension de l'espace de caractéristiques. On peut faire l'Analyse en Composantes Principales (ACP) pour diminuer la complexité de computation.

k-NN est une technique supervisée. Dans *training data set*, pour chaque histogramme, le type de plan associé est déjà connu. Tous les histogrammes sont représentés par un point dans l'espace de caractéristiques. La classification d'un plan inconnu (*testing data*) consiste à lui attribuer la classe la plus représentée parmi ses k plus proches voisins dans les objets d'apprentissage.

4.3.2 Méthode de Réseau neuronal convolutif

Pour ce problème typique de classification en fonction de figure, le CNN - Convolutional neural network est aussi un bon choix.

On propose un simple modèle de CNN + une couche entièrement connectée. CNN est utilisé pour extraire les caractéristiques de figure, une grande nombre de couches de CNN peut nous aider à extraire plus de caractéristiques, mais plus de capacité de computation sera utilisée. La couche entièrement connectée est utilisée pour prédire le type de classification.

Dans ce cas, les données que l'on donne au model sont les figures de l'histogramme, et les différents types de plan sont les résultats de classification. A l'aide de la propagation backward et forward, on peut finalement trouver le meilleur poids pour chaque noeud d'une couche à les noeuds dans une autre couche adjacente.

Une difficulté de réseau neuronal est de fixer les paramètres et les hyper-paramètres comme le taux d'apprentissage (learning rate), le nombre d'itérations (epochs), le nombre de couche cachant (num of hidden layer), la taille du batch (batch size), le type de optimiseur (optimiser), etc. Un bon choix de paramètre peut nous donner un très bon résultat classification.

Ces deux techniques proposées sont toutes supervisées. Comme on n'a pas d'ensemble des données de formation, on ne peut pas appliquer des méthodes de *Machine Learning* ou former un modèle de classification. Donc on ne peut pas vérifier si ces deux propositions marchent ou pas.

Cependant, on peut analyser des difficultés sur la classification de type de plans.



Figure 4.9: Présentation de quelques plans et les histogrammes associés

L'histogramme (V_x, V_y) va être affecté par le mouvement des objets dans le plan. Par exemple, dans la figure 4.9, les deux premières cas sont des prises de plan fixe, mais leurs histogrammes associés ne correspondent pas aux caractéristiques décrites dans la section 3.4. C'est parce que la femme et le hélicoptère continuent de vibrer. En outre, dans les deux cas en bas, le plan est travelling vers certaine direction. Le mouvement de fond est conforme à la théorie, mais les deux personnes suspendues sous l'hélicoptère se déplacent avec la caméra, parallèlement à d'autres actions. Cela fait que l'histogramme n'a pas de caractéristiques typiques de plan travelling.

Pour conclure, l'histogramme de composantes (V_x, V_y) peut indiquer les différents types de plan si les objets dans l'image sont stationnaires par rapport au système de référence au sol. Si ils sont en mouvement, l'identification de plan va être plus compliquée. On propose d'utiliser des classificateur d'apprentissage, mais on ne peut pas vérifier leur applicabilité car il manque data set.