Département des sciences  
Faculté de l’informatique  
Université de Sherbrooke

Sujet de recherche

Par   
Justine Dauphinais (DAUJ8984)  
Julien Massicotte (MASJ1787)

Remise à  
François Rheault

Sherbrooke  
15 décembre 2024

# Introduction

### Sujet

Notre projet porte sur l'ajout d'artéfacts vidéo couramment observés lors du traitement numérique de vidéos. Ces artéfacts, tels que le bruit, la compression, la distorsion des couleurs, et le flou, sont souvent introduits pendant l'enregistrement, le codage ou la transmission des vidéos. L'objectif de ce projet est de démontrer et d'analyser les différents types d'artéfacts visuels qui peuvent être créés artificiellement dans des vidéos et d’explorer comment chacun peut être créer.

### Pertinence du projet

Il est donc étroitement lié aux concepts de traitement du signal numérique et de compression des médias, qui ont été couverts durant le cours. L’ajout d’artéfacts comme le bruit, la compression JPEG, ou la distorsion des lentilles, implique l’application de concepts mathématiques et algorithmiques tels que la transformée de Fourier, le filtrage, et l’échantillonnage, permettant de mieux comprendre les limitations et les défis du traitement numérique des vidéos.

Importance des concepts

Ces concepts ont une importance considérable dans la vie courante, car les artéfacts vidéo affectent directement la qualité des vidéos que nous consommons quotidiennement, que ce soit sur des plateformes de streaming, lors de visioconférences, ou lors de l'utilisation de caméras de surveillance. Comprendre comment ces artéfacts sont créés et comment ils peuvent être atténués est essentiel pour les professionnels de l'audiovisuel, les ingénieurs, et les chercheurs qui cherchent à améliorer la qualité des médias numériques, des métiers qui pourraient potentiellement être les nôtres après nos études.

### Connaissance nécessaire

Une compréhension de base des concepts d’ondes et de signaux (bruit, échantillonnage, etc.), des techniques d’imagerie numérique (résolution, compression, etc.), et des outils de traitement d’image ([OpenCV](https://opencv.org/), Python) est requise.

Grands concepts nécessaires

Les grands concepts nécessaires pour une bonne compréhension du sujet sont :

* **Bruit** : Le bruit est une variation aléatoire des niveaux de couleur ou de luminosité dans une image (souvent perçu comme des points), qui peut rendre la vidéo moins nette ou plus "sale".
* **Compression** : Réduction de la taille des fichiers vidéo, pouvant introduire des pertes de qualité visibles comme des blocs ou des flous.
* **Distorsions optiques** : Déformations géométriques dues aux propriétés de la lentille, comme le barillet.
* **Flou** : Modélisé par un noyau de convolution, utile pour reproduire les effets de mouvements dans les vidéos.
* **Distorsion d’objectif** : Liée à l’optique des caméras, elle modifie la géométrie d’une image capturée.
* **Aberration chromatique** : Décalage des couleurs, souvent visible sur les bords d’une image, causé par une lentille non corrigée.

# Méthodologie & Analyse

### Outils et logiciels utilisés

* ***OpenCV*** : Librairie *open-source* de traitement d'images et de vidéos utilisée pour les manipulations vidéo. Gratuit et largement utilisé dans le domaine de l'imagerie numérique.
* ***NumPy*** : Librairie Python *open-source* utilisée pour la manipulation efficace des tableaux et les opérations mathématiques nécessaires.
* ***Python*** : Langage de programmation *open-source*, choisi pour sa flexibilité et la richesse de son écosystème.
* **Logiciels** : Aucun logiciel propriétaire n’a été utilisé, rendant le projet accessible et reproductible.
* **Matériel** : Un ordinateur standard avec une vidéo trouvée sur YouTube et coupée pour minimisée le temps de traitement de celle-ci pour l’entrée.

### Données nécessaires

* **Fichier vidéo source** : Une vidéo en format .mp4 utilisée comme point de départ. La vidéo doit être suffisamment nette et sans artéfacts initiaux pour que les manipulations soient clairement observables.
* **Propriétés** : Vidéo avec un nombre constant d’images par secondes (FPS), une résolution stable, et une bonne qualité de compression initiale.

### Manipulations effectuées

* **Ajout d'artéfacts** : Utilisation des fonctions définies dans le script Python (*gaussian\_noise, motion\_blur*, etc.) pour introduire des distorsions spécifiques sur les frames de la vidéo.
* **Division des vidéos** : Création de fichiers vidéo individuels pour chaque type d'artéfact.
* **Paramétrage dynamique** : Ajustement des intensités et des paramètres pour chaque artéfact, comme le niveau de bruit, la taille du noyau pour le flou, ou la qualité de compression. Ceux-ci vont être utilisés pour exagéré les artéfacts afin de pouvoir bien les observer durant notre analyse.

### Entrées et sorties

* **Entrées** :
  + Fichier vidéo source au format .mp4.
  + Paramètres de configuration pour chaque artéfact (e.g., niveau de bruit, intensité de vignetage).
* **Sorties** :
  + Une série de fichiers vidéo .mp4, chacun présentant un artéfact spécifique :
    - **chromatic\_aberration.mp4**
    - **color\_branding.mp4**
    - **compression.mp4**
    - **dust\_particles.mp4**
    - **frame\_dropping.mp4**
    - **gaussian\_noise.mp4**
    - **lens\_distortion.mp4**
    - **motion\_blur.mp4**
    - **salt\_pepper\_noise.mp4**
    - **vignetting.mp4**

### Représentation des données utilisées

Les vidéos sont représentées comme une séquence de frames, chaque frame étant une matrice 2D ou 3D (selon qu’elle est en niveaux de gris ou en couleur). Les transformations sont appliquées à chaque frame individuellement pour maintenir la cohérence temporelle avec pour exception du *frame dropping* qui lui va sauter des images de la vidéo.

### Données acquises et traitements effectués

* **Données acquises** : Vidéos modifiées représentant chaque artéfact.
* **Traitements effectués** :
  + Application de filtres (Gaussien, *motion blur*, etc.).
  + Transformation des couleurs pour simuler le color banding et l’aberration chromatique.
  + Manipulation géométrique pour simuler la distorsion des lentilles.

### Résultats obtenus et difficultés rencontrées

Les résultats de l'expérimentation montrent une visualisation claire des différents artéfacts vidéo appliqués. À chaque étape, leur impact visuel a été observé et analysé à divers niveaux d’intensité, permettant de mieux comprendre leur comportement et leur interaction avec les données de la vidéo source. Cette approche a également mis en lumière les limitations des algorithmes de traitement d’image, notamment leurs effets sur la qualité visuelle finale lorsqu'ils sont appliqués de manière excessive ou non optimisée.

Une des principales difficultés a été de déterminer comment appliquer chaque effet de manière cohérente pour représenter fidèlement les différents types d'artéfacts vidéo. Cela nécessitait une compréhension approfondie des principes physiques ou mathématiques derrière chaque artéfact, comme la distorsion de lentilles, le bruit ou l'aberration chromatique. Trouver les bons paramètres pour chaque transformation, tout en veillant à conserver une représentation réaliste et perceptible des effets, a demandé des ajustements itératifs et une expérimentation constante. Cette étape a été essentielle pour s’assurer que chaque artéfact soit à la fois identifiable et conforme aux attentes théoriques.

# Discussion

### Pertinence et performance des outils utilisés

Les outils utilisés, notamment Python, OpenCV et NumPy, se sont avérés parfaitement adaptés au projet. OpenCV, en particulier, est optimisé pour le traitement d'images et de vidéos et offre des fonctions puissantes pour manipuler des frames en temps réel. NumPy a permis une gestion efficace des opérations matricielles, essentielles pour les transformations appliquées.

Les performances ont été satisfaisantes pour les vidéos de résolution moyenne, mais le traitement a ralenti pour des vidéos en haute résolution, en particulier lors de manipulations intensives comme le flou ou le calcul des distorsions de lentilles. La longueur de la vidéo jouait aussi particulièrement un rôle dans la longueur du traitement. Nous avions une vidéo de 15 secondes ce qui donnait 450 images (environ 30 images par seconde). Des optimisations supplémentaires pourraient améliorer ces performances et nous donner la possibilité de traiter des vidéos en plus haute résolution et plus longue sans sacrifier beaucoup de temps.

### Adéquation des données utilisées

La vidéo source (coupée à 15 secondes) était appropriée pour démontrer clairement les artéfacts visuels. Une vidéo nette et sans artefact initial a permis de mieux observer les effets des manipulations appliquées. Cependant, il aurait été intéressant de comparer avec une vidéo ayant déjà des artéfacts pour étudier les interactions entre ceux-ci.

Avec du recul, il vient à l’esprit qu’il était possible de réduire le temps de traitement en réduisant le nombre d’image par seconde dans la vidéo,

### Lien avec les concepts vus en classe

**Traitement de signal** : Les notions de bruit, de filtrage, et de convolution ont été directement appliquées dans l’ajout d’artéfacts comme le flou de mouvement ou le bruit gaussien.

**Compression vidéo** : La compréhension des effets de la quantification et des algorithmes de compression a permis d’introduire des artéfacts de compression réalistes.

**Perception humaine** : Les concepts liés à la sensibilité de l’œil humain ont guidé le choix des paramètres pour s’assurer que les artéfacts soient perceptibles.

**Imagerie numérique** : Les bases de la conversion numérique-analogique ont été cruciales pour comprendre l’impact des transformations appliquées.

### Concepts découverts ou approfondis

**Compression**

****

Figure : Image de montgolfières avec de la compression

**Définition : …**

**Origine : d’où ça vient + si c’était avec l’échantillonnage ou autre**

**Bruit Gaussien**

****

Figure : Image de montgolfières avec du bruit Gaussien

**Définition : …**

**Origine : …**

**Particules de poussières**

****

Figure : Image de montgolfières avec des particules de poussières

**Définition : …**

**Origine : …**

**Flou de mouvement**



Figure : Image de montgolfières avec du flou de mouvement (motion blur)

Définition : …

Origine : …

**Distorsion de lentilles**



Figure : Image de montgolfières avec de la distorsion de lentilles

Définition : …

Origine : …

### Observations pertinentes

Les artéfacts, bien qu’ils soient indésirables, peuvent être intentionnellement ajoutés dans des domaines artistiques ou cinématographiques pour créer un effet spécifique.

Les artéfacts liés à la compression sont particulièrement perceptibles dans les zones à faible contraste, ce qui reflète l’importance de l’analyse spatiale et fréquentielle dans les algorithmes de compression.

Certains artéfacts, comme le bruit gaussien ou le flou de mouvement, peuvent servir de métaphores pour démontrer les limitations des capteurs ou des systèmes de transmission.

### Complexité de l'expérimentation et la pertinence du questionnement initial

Bien que les outils utilisés soient accessibles, la mise en œuvre des différents effets nécessitait une compréhension approfondie des transformations algorithmiques et des concepts de traitement de signal. Beaucoup d’heures ont été mises dans le développement du script et dans la recherche de méthode de réaliser les effets souhaités.

Le questionnement était pertinent car il relie les aspects théoriques vus en classe à des observations pratiques, tout en mettant en évidence l'impact des artéfacts sur la perception visuelle.

# Conclusion

### Faites part de votre réflexion personnelle sur le projet

Ce projet a été une excellente occasion de mettre en pratique les concepts théoriques appris en classe tout en découvrant de nouveaux aspects du traitement vidéo. Il a permis de mieux comprendre les mécanismes sous-jacents à l'apparition des artéfacts visuels et leur impact sur la qualité perçue des vidéos. La diversité des transformations appliquées a révélé à quel point de simples ajustements peuvent influencer la perception visuelle, renforçant ainsi l’importance des détails dans les applications d’imagerie numérique.

### Évaluez l'intérêt du projet et la charge de travail qu'il a nécessitée

L'intérêt du projet réside dans son applicabilité directe à des domaines variés, tels que le streaming vidéo, la surveillance ou la production multimédia. Ce type d’analyse est pertinent pour quiconque cherche à optimiser la qualité visuelle ou à comprendre les limitations des technologies d'imagerie actuelles. Cependant, le projet a été relativement exigeant en termes de charge de travail, nécessitant une gestion minutieuse des paramètres pour chaque artéfact et une compréhension approfondie des outils de traitement vidéo. Bien que les outils utilisés aient facilité l’implémentation, le temps consacré à l’ajustement et aux tests a été conséquent.

### Analysez si des éléments manquent à votre rapport/expérimentation

Bien que le projet soit complet dans sa portée actuelle, certains éléments auraient pu être inclus pour approfondir l’analyse. Par exemple, une évaluation quantitative de l’impact des artéfacts, à travers des métriques de qualité vidéo comme le PSNR (*Peak Signal-to-Noise Ratio*) ou le SSIM (*Structural Similarity Index*), aurait enrichi le rapport. De plus, une comparaison avec des artéfacts présents dans des vidéos réelles aurait permis d’évaluer la fidélité des effets simulés par rapport aux conditions réelles.

### Proposer des pistes pour approfondir le questionnement ou poursuivre les recherches

Pour aller plus loin, il serait intéressant d’étudier les stratégies d’atténuation ou de correction des artéfacts, telles que le post-traitement par intelligence artificielle ou le filtrage avancé. Aussi, intégrer des vidéos provenant de sources variées (e.g., drones, caméras de surveillance) permettrait de tester l’adaptabilité des techniques appliquées à des contextes diversifiés.

[Fin]

## Script

L’exposé oral consistera à 1) Enregistrer un vidéo de 10-12 minutes présentant le questionnement ayant initié le mini-projet, décrire les données utilisées, présenté le choix outils, librairie, logiciel et faire démonstration et un compte-rendu de ce qui a été fait et vulgariser ce qui a été appris par les membres de l’équipe.

1. Questionnement ayant initié le mini-projet [1 minutes]
2. Décrire les données utilisées [1 min]

Décrire la vidéo utilisée et pourquoi elle a été utilisée (couleur, dimension, clarté, manque d’artéfacts déjà présents)

1. Présenté le choix outils (librairie, logiciel) [5 min]

Présentation de la librairie OpenCV, NumPy. Utiliser Python.

OpenCV : OpenCV (*Open Source Computer Vision Library*) est une bibliothèque de fonctions de programmation principalement pour la vision par ordinateur en temps réel. Développé à l’origine par Intel, il a ensuite été soutenu par Willow Garage, puis Itseez (qui a ensuite été acquis par Intel). La bibliothèque est multiplateforme et sous licence libre et open-source sous la licence Apache 2. À partir de 2011, OpenCV propose une accélération GPU pour les opérations en temps réel. [TEXTE DE BASE À CHANGER]

NumPy : NumPy est une puissante bibliothèque Python utilisée pour travailler avec des tableaux et effectuer un large éventail d’opérations mathématiques. Il s’agit du package fondamental pour le calcul scientifique avec Python et prend en charge les tableaux, les matrices, l’algèbre linéaire, les transformées de Fourier, etc.

Présentation du code (à clean up et commenter pour facilité la compréhension. On peut passer sur tout ou seulement sur les artéfacts sélectionnés)

1. Démonstration et compte rendu de ce qui a été fait et vulgariser ce qui a été appris par les membres de l’équipe [5 min]