

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 observés lors du traitement numérique de vidéos. Ces artéfacts, tels que le bruit gaussien, la compression, la distorsion des couleurs et le flou, sont souvent introduits pendant l'enregistrement, le codage ou la transmission des vidéos. L'objectif 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 créé avec du code Python.

Pertinence du projet

Notre projet est donc 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 (vue amplement dans le cours de IMN359 avec la DCT locale), 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 dans la vie courante, car les artéfacts vidéo affectent la qualité des vidéos que nous consommons quotidiennement, que ce soit sur des plateformes de streaming telles que YouTube ou, plus récemment, Twitch, lors de visioconférences Zoom et Teams, ou même 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 utilisés ([OpenCV](#), 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** : Déformations géométriques dues aux propriétés de la lentille, comme le barillet.

- **Flou** : Modélisé par un noyau de convolution (plus souvent appelé kernel dans notre projet), utile pour reproduire les effets de mouvements dans les vidéos.

* Les explications par rapport aux artéfacts non présentés ont été enlevés.

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.¹
- **NumPy** : Librairie Python *open-source* utilisée pour la manipulation efficace des tableaux et les opérations mathématiques nécessaires.²
- **Python** : Langage de programmation *open-source*, choisi pour sa flexibilité et la richesse de son écosystème.³

Données nécessaires

- **Fichier vidéo source** : Une vidéo en format .mp4 utilisée comme point de départ. La vidéo doit être suffisamment nette et avec des couleurs (une vidéo d'un mur blanc ne serait pas optimale pour voir la majorité des artéfacts).
- **Propriétés** : Vidéo avec un nombre constant d'images par secondes (FPS), une résolution stable, et une bonne qualité de compression initiale afin de bien pouvoir voir les effets des artéfacts étudiés (une vidéo déjà compressée ne serait pas une bonne vidéo à utiliser pour voir les effets de la compression si le paramètre utilisé est plus grand).

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 (kernel), ou la qualité de compression. Ceux-ci vont être utilisés pour exagérer 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 (niveau de bruit, qualité de la compression, propriétés pour la distorsion de lentille (k1, k2), etc.).

¹ <https://opencv.org>

² <https://numpy.org>

³ <https://www.python.org>

- **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
 - Une série d'images .png, chacun présentant un artéfact spécifique :
 - chromatic_aberration.png
 - color_branding.png
 - **compression.png**
 - **dust_particles.png**
 - frame_dropping.png
 - gaussian_noise.png
 - lens_distortion.png
 - motion_blur.png
 - salt_pepper_noise.png
 - vignetting.png

** Ceux en gras sont ceux étudiés pendant notre projet.*

Représentation des données utilisées

Les vidéos sont représentées comme une séquence d'images, chaque image étant une matrice 2D. Les transformations sont appliquées à chaque image individuellement pour maintenir la cohérence temporelle (commencer de l'image 1 et finir à l'image 450) avec pour exception du *frame dropping* qui, lui, va sauter des images de la vidéo à la place de les changer.

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 afin d'approfondir nos connaissances. Cette approche a également mis en emphase 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. Même après des heures de travail, il est possible que certaines applications ne soient pas idéales (spécifiquement pour le flou de mouvement, qui est difficile à répliquer, si pas impossible artificiellement).

Discussion

Pertinence et performance des outils utilisés

Les outils utilisés (*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. Il a été une merveilleuse découverte et va sûrement être utilisé plus tard dans nos vies.

NumPy, lui, a permis une gestion efficace des opérations matricielles, essentielles pour traiter les différentes images.

Les performances ont été satisfaisantes pour les vidéos de résolution moyenne (720p et moins), mais le traitement a ralenti pour des vidéos en haute résolution (1080p et plus), en particulier lors de manipulations intensives comme le flou de mouvement ou le calcul des distorsions de lentilles. La longueur de la vidéo jouait aussi particulièrement un rôle dans la longueur du traitement. Nous avons 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, mais une connaissance approfondie de *OpenCV* et de *Python* serait nécessaire pour cela.

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 (ou de faire passer une vidéo à travers le script deux fois) 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 1 : Image de montgolfières avec de la compression

Les artéfacts de compression s'agissent de distorsions qui apparaissent sous forme de blocs, de flou ou de bandes de couleur dans les images et les vidéos en raison d'une compression agressive.

Elles proviennent d'une surutilisation d'algorithme de compression pour réduire le plus possible la taille du fichier d'une image ce qui peut amener à une perte d'information. Cet artefact est lié à comment l'image est compressée.

Dans notre cas, nous avons utilisé la compression JPEG qui vient avec la librairie utilisée, mais nous l'avons mis à son plus extrême afin de pouvoir bien démontrer ce que de la compression fait à une image.

Nous pouvons voir que les grandes étendues de couleurs similaires, telles que le ciel, deviennent des blocs uniformes. Toutefois, les endroits avec plus de détails ou de changements de couleurs drastiques (particulièrement les montgolfières, utilisent clairement plus de carrés. Il est possible de voir des formes plus petites avec plus de détails, indiquant que ces endroits sont moins compressés afin de garder une meilleure fidélité à l'image originale.

C'est à ce que nous nous attendions maintenant que nous avons étudié la DCT locale en IMN359.

Bruit Gaussien



Figure 2 : Image de montgolfières avec du bruit Gaussien

Le bruit fait référence aux variations aléatoires de luminosité ou aux informations de couleur dans les images et les vidéos. Il apparaît sous forme de taches granuleuses et peut masquer les détails les plus fins. Pour qu'un bruit soit considéré comme étant Gaussien, il faut que le bruit ait la densité de probabilité d'une distribution gaussienne.^{4 5}

Il apparaît dans les images en raison du bruit du capteur provoqué par un mauvais éclairage et/ou une température élevée et un bruit de circuit électronique.⁶

Encore une fois, nous avons pousser l'algorithme à son maximum afin d'avoir un effet simple à apercevoir dans une image. C'était à ce que nous nous attendions, et nous sommes contents des résultats obtenus pour cet artéfact.

⁴ <https://vitrinelinguistique.oqlf.gouv.qc.ca/fiche-gdt/fiche/3299119/bruit-gaussien>

⁵ [https://web.archive.org/web/20160918164948/https://miac.unibas.ch/SIP/06-Restoration.html#\(8\)](https://web.archive.org/web/20160918164948/https://miac.unibas.ch/SIP/06-Restoration.html#(8))

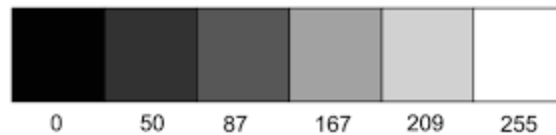
⁶ [https://web.archive.org/web/20160918164948/https://miac.unibas.ch/SIP/06-Restoration.html#\(14\)](https://web.archive.org/web/20160918164948/https://miac.unibas.ch/SIP/06-Restoration.html#(14))

Bruit sel et poivre (ou bruit impulsif)



Figure 3 : Image de montgolfières avec du bruit 'salt and pepper'

Le bruit poivre et sel apparaît sous forme de pixels aléatoires dans une image définie entre noir (0) et blanc (255). Cela ressemble à des points blancs et noirs parsemés sur l'image.⁷



Réalistiquement, le bruit sel et poivre ne ressemble pas à cela. Voici une image plus fidèle à la réalité :

7

<https://www.sciencedirect.com/science/article/pii/S2215098618321050#:~:text=Impulse%20noise%20usually%20corrupts%20images,values%20%5B0%2C55%5D.>



Figure 4 : Image de Lena avec du bruit de sel

En réalité, le bruit de sel est plus difficile à voir. Toutefois, le bruit de poivre :



Figure 5 : Image de Lena avec du bruit de poivre

Cela ressemble plus à ce que nous avons obtenu, ce qui veut dire qu'un ajustement serait nécessaire pour le bruit de sel.

Flou de mouvement (*motion blur*)



Figure 6 : Image de montgolfières avec du flou de mouvement (*motion blur*)

Le flou de mouvement est la traînée apparente d'objets en mouvement dans une image ou dans l'ensemble de l'image en raison d'un mouvement rapide ou de longs temps d'exposition.

Il se produit lorsqu'une photo est prise avec une exposition longue et des mouvements rapides. Cela veut dire que notre recréation du flou de mouvement n'en est pas réellement, puisque la distance entre la caméra et l'objet en mouvement est nécessaire pour cet effet.

Un exemple de flou de mouvement connu est "*Star Rain in the Desert*".⁸



Figure 7 : *Star Rain in the Desert*

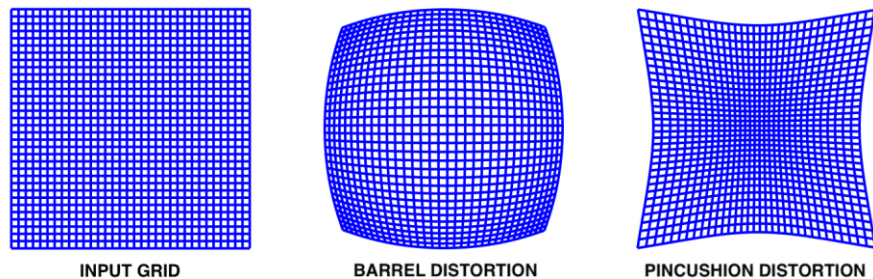
⁸ <https://www.ntnu.no/blogger/richard-hann/2021/10/07/preventing-motion-blur-in-drone-mapping/>

Distorsion de lentilles



Figure 8 : Image de montgolfières avec de la distorsion de lentilles

Il s'agit d'une déformation de l'image courbant les lignes droites proches des extrémités de l'image. Il y a différents types de distorsions :



Celle représentée s'appelle distorsion en coussinet. Souvent, ces effets sont intentionnels et créés avec des lentilles spécialisées.⁹

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 méthodes pour réaliser les effets souhaités.

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

⁹ [What is Lens Distortion?](#)

Conclusion

Réflexion personnelle

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 nous a permis de mieux comprendre les mécanismes des apparitions d'artéfacts visuels et leur impact sur la qualité des vidéos. La diversité des transformations appliquées a révélé à quel point de simples ajustements peuvent influencer la perception visuelle, renforçant l'importance des détails dans les applications d'imagerie numérique et dans la prise de données.

Intérêt du projet

L'intérêt du projet réside dans son applicabilité directe à des domaines variés mentionnés plus tôt. 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 bonne gestion des paramètres pour chaque artéfact et une compréhension approfondie des outils de traitement vidéo. Bien que les outils utilisés aient facilité l'implémentation (*OpenCV* a été exceptionnel pour cela), le temps consacré à l'ajustement et aux tests a été exigeant et beaucoup plus que nous avions planifiés au début.

Éléments manquants

Bien que le projet soit complet dans sa portée actuelle, certains éléments auraient pu être inclus pour approfondir notre analyse.

Par exemple, une évaluation quantitative de l'impact des artéfacts, à travers des métriques de qualité vidéo comme le PSNR¹⁰ (*Peak Signal-to-Noise Ratio*) ou le SSIM¹¹ (*Structural Similarity Index*), aurait enrichi notre rapport avec de vrais nombres pour comparer. Ou, plus simplement, nous aurions pu implémenter des fonctions déjà créées en IMN359 pour voir le pourcentage d'erreur des photos ajoutées dans notre rapport.

De plus, une comparaison avec des artéfacts présents dans des vidéos réelles aurait permis d'évaluer la fidélité de nos effets simulés par rapport aux conditions réelles.

Pistes pour approfondir le questionnement

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. Nous avons eu le temps de regarder ceux-ci vaguement; nous avons même décidé de laisser le script « `artefact_remover.py` » dans le dossier .zip. Les techniques pour enlever des artéfacts pourraient être faisable avec la librairie `cv2`, mais par manque de temps, nous n'avons malheureusement pas été capable d'aller voir plus loin que la base.

¹⁰ <https://www.mathworks.com/help/vision/ref/psnr.html>

¹¹ <https://www.mathworks.com/help/images/ref/ssim.html>

Aussi, intégrer des vidéos provenant de sources variées (drones, caméras de surveillance) permettrait de tester l'adaptabilité des techniques appliquées à des contextes diversifiés. Après tout, il est plausible de penser que les différents artefacts, souvent causés par des irrégularités, changent par rapport à l'environnement donné.

[Fin]