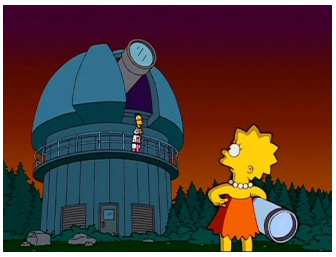
SAÉ. C2
Pollution lumineuses et images astronomiques

Samuel Delepoulle et Rémi Cozot

19 octobre 2023



1 Mise en situation

À la fin de vos études, vous êtes embauché à l'observatoire de Springfield, un établissement astronomique renommé, avec un sentiment d'excitation indescriptible. Votre rêve de travailler aux côtés de chercheurs chevronnés et d'explorer les merveilles de l'univers semble enfin devenir réalité. Vous êtes prêt à plonger dans le monde de l'astronomie, à contribuer à de nouvelles découvertes et à faire votre marque dans ce domaine fascinant.

Cependant, à votre arrivée à Springfield, une réalité sombre vous frappe de plein fouet. Vous découvrez que la ville a récemment pris une décision controversée. Incapable de gérer l'excédent d'électricité produit par la centrale atomique locale, les autorités ont choisi de la réutiliser pour illuminer la ville de manière extravagante. Des panneaux publicitaires lumineux clignotent, des lampadaires éblouissants envahissent les rues, et même la célèbre place de la ville est transformée en une véritable scène de Broadway avec des projecteurs brillants.

Pour l'observatoire, cette décision a un impact dévastateur. La pollution lumineuse provoquée par cet éclairage excessif rend impossible l'observation du ciel nocturne. Les photos astronomiques sont désormais brouillées par cette lumière envahissante, rendant le travail des chercheurs extrêmement

difficile. Vous vous retrouvez confronté à un nouveau défi : comment pouvez-vous contribuer à la recherche astronomique dans de telles conditions ?

(a) Photo capturée avec des lumières parasites (pollution lumineuse)

(b) Image traitée par un algorithme de retrait du gradient

FIGURE 1 – Exemple d'algorithme de retrait du gradient

2 Comment traiter la pollution lumineuse ?

La pollution lumineuse se manifeste comme un signal supplémentaire dans l'image. La forme de ce signal est celui d'un gradient ajouté. Pour effectuer un traitement de ces images il y a deux étapes :

estimer le gradient : il existe plusieurs méthode pour l'évaluer. La méthode la plus simple consiste à considérer qu'il est linéaire, ce qui est souvent une bonne approximation). Il existe d'autres solutions, par exemple utiliser des polynômes d'ordre supérieur. Il existe actuellement des méthodes encore plus performantes basées sur l'estimation de fonctions par des réseaux de neurones.

retirer le gradient : la solution la plus simple est en général de réaliser une simple soustraction du gradient à l'image de départ pour obtenir l'image traitée.

Pour simplifier vos tests, au départ, le gradient vous est fourni dans une image. Vous aurez simplement à effectuer la différence.

3 cahier des charges

3.1 Défi « basique » (simple)

Vous réaliserez donc une application qui permet de traiter les images astronomiques et d'en soustraire la pollution lumineuse. Votre application devra être capable de charger une image et d'en soustraire le gradient (le gradient étant stocké dans un fichier que vous n'avez pas à calculer).

Pour cela, vous développerez une interface en python et vous utiliserez la bibliothèque `opencv`. Cette bibliothèque permet de réaliser de nombreux traitement d'images et est écrite en C++. Elle peut peut également être appelée en python.

ressources

— `https://docs.opencv.org/4.x/`
— `https://pypi.org/project/opencv-python/`

3.2 Défis « intermédiaires »

note : les deux parties suivantes sont indépendantes et peuvent être réalisées dans l'ordre de votre choix.

— implémentez au moins une méthode de détection du gradient

— effectuez une comparaison sur l'efficacité des traitements en python et en C++

3.3 Défis « avancés »

— implémentez plusieurs méthodes de détection du gradient
— prévoyez un traitement par lot d'images, par exemple, plusieurs images dans le même dossier. Attention, le gradient n'est pas forcément le même !
— comparez l'efficacité de vos méthodes sur plusieurs aspects : résultat, temps d'exécution, utilisation de la mémoire, etc.
— effectuez tout autre ajout que vous trouvez pertinent.

note il n'est pas nécessaire d'avoir réalisé toutes les étapes pour obtenir une bonne note. Par exemple, bien réaliser un défi simple et un défi intermédiaire, si le code est bien organisé, lisible et commenté doit permettre d'avoir la moyenne.

3.4 Défis « experts »

— Proposez et implémentez une architecture logicielle permettant de représenter une « suite » de traitements d'images comme un graphe acyclique orienté de traitements atomiques. Dans ce modèle, récupérer les fichiers images d'un répertoire, lire une image, évaluer le gradient, supprimer le gradient sont des traitements atomiques.
— Proposez un langage simple permettant de définir une « suite » de traitement et implémentez un « transpileur » transformant un fichier dans votre langage en un code C++/python permettant d'effectuer les traitements

Remarque : pour simplifier l'analyse du fichier de langage celui-ci pourra suivre une syntaxe JSON.

3.5 Livrables

Vous rendrez sur moodle :
— les fichiers sources de votre application
— au minimum un fichier exécutable (pour Windows ou pour Linux)
— un fichier README.md qui contiendra :
  — Comment compiler votre programme ;

— Comment l'exécuter (préciser la plateforme d'exécution sur laquelle vous avez testé) ;
— Une liste concise des réalisations

3.6 Remarques

— Le travail sera réalisé par binômes.
— la notation prendra en compte la qualité de la réalisation, les commentaires du code ainsi que la difficulté de solution utilisée.
— Tout plagiat (entre étudiants ou en rendant des fichiers « récupérés ») sera sanctionné par la note 0.