

Étudiant	Simon ROOSENS
Matricule	e226434
Cours	M1-DEDI0022-C-a-W-Software Middleware
Travail	Rapport Examen

## Table des matières

Description .....

Comparaison avec d’autres middlewares, .....

Fonctionnalités .....

Types de filtres implémentés .....

Interface Software (API) .....

La suite .....

Présentation .....

1

1

2

2

3

4

4

## Description

Dans le cadre du cours de Software Middleware, il nous a été demandé d’analyser un besoin, de définir un scope et d’implémenter les fonctions d’une API. Le besoin identifié est un ensemble des fonctions d’amélioration d’une image à l’aide de traitements d’adoucissement et autres à partir d’un middleware. Le besoin d’intégration de fonctions de traitement d’image dans un outil est un cas d’usage commun pour un développeur d’outils et celui-ci devrait pouvoir ajouter ces fonctions rapidement et sans formation spécifique (notamment qu’est ce que le traitement d’image, quels sont les algorithmes de traitement et comment les utiliser, ...). Les APIs similaires identifiées exposent de nombreuses fonctions mais peuvent être gourmandes en ressources et semblent complexes à intégrer. C’est pourquoi nous proposons dans ce travail un middleware simple, efficace, fiable et suffisamment commenté pour permettre à un intégrateur d’ajouter cette librairie de fonctions au sein de son projet.

## Comparaison avec d’autres middlewares,





Nom	Type	Description	Lien	Cons
GIMP	Tool/API	Logiciel d’édition et de retouche d’image gratuit et libre de droits. Est open source et expose donc ses APIs	<a href="https://www.gimp.org/downloads/">https://www.gimp.org/downloads/</a>	Gratuit mais beaucoup de fonctionnalités, GIMP n’est pas l’outil le plus simple à utiliser. Les APIs sont quant à elles forts complexes à intégrer.
OpenCV	API	Librairie spécialisée dans le traitement d’images en temps réel	<a href="https://opencv.org/">https://opencv.org/</a>	Gratuit et rapide mais expose une interface de programmation assez complexe à utiliser. De plus, la gestion des erreurs est très rudimentaire



## Fonctionnalités

Le middleware **ImageDenoizerAPI** expose les fonctions suivantes,

- Chargement d'une image
- Choix d'un type de filtre de réduction de bruit
- Choix de paramètres relatifs au filtre sélectionné
- Sauvegarde de l'image traitée

## Types de filtres implémentés

Type	Description	Illustration
Gaussian Blur	Le flou gaussien applique un traitement sur une image en utilisant des algorithmes de fonctions gaussiennes. Est couramment utilisé pour réduire le bruit sur une image en ajoutant un effet de flou paramétrable.	<div>Previous</div>  <div>Denoized</div> 
Median Blur	Est un filtre médian non-linéaire utilisé également pour réduire le bruit sur une image ou même un signal vidéo ou autre. Il est largement utilisé car permet dans certains cas de réduire le bruit d'une image tout en conservant ses contours.	<div>Previous</div>  <div>Denoized</div> 

Non-Local Means	Permet de réduire le bruit d'une image d'une manière différente des 2 algorithmes plus haut. En effet, il n'applique pas de moyenne sur un groupement de pixels localisés autour d'un pixel mais plutôt une moyenne sur l'ensemble des pixels de l'image. Il permet de diminuer la perte de détails comparé aux 2 autres filtres.	<div> <div>Previous</div>  </div> <div> <div>Denoized</div>  </div>
-----------------	---	---

## Interface Software (API)

L'ensemble du projet est codé en C++, ce qui permet d'utiliser les concepts d'orienté objet tout en ayant une main mise sur les mécanismes implémentés.



L'API utilise en son sein une librairie bien connue du traitement d'image qui se nomme OpenCV. Elle permet de réaliser de nombreuses tâches liées au traitement vidéo et de l'image en temps réel (Fig. 2). Il a été clair que partir sur ce type de solution en back-end allait permettre un taux de réutilisabilité assez important, il sera uniquement nécessaire d'ajouter les interfaces aux différentes méthodes à utiliser (Fig. 3). L'API se chargera ensuite de communiquer l'image traitée via un signal. De plus, cette interface a été construite de sorte que les paramètres qui lui sont transmis sont vérifiés de manière transparente car la librairie OpenCV est limitée vis à vis de sa gestion des erreurs.

Figure 1 : OpenCV en temps que librairie de traitement photo/vidéo en temps réel. Par exemple, peut être utilisé comme détecteur d'objets.

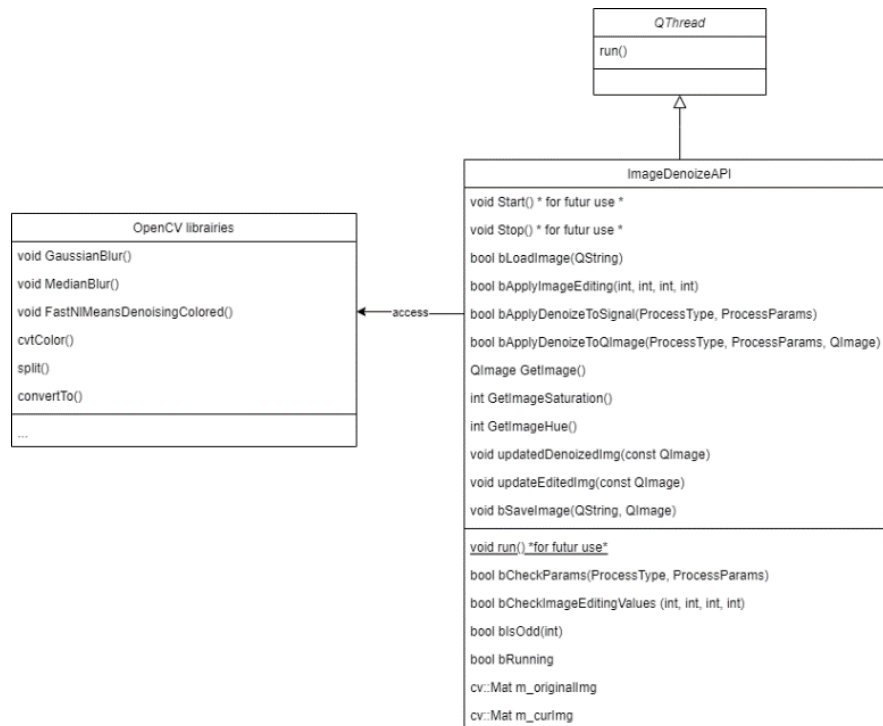


Figure 2: Diagramme de classe de l'API Image Denoizer

L'utilisation du framework **QT**, quant à lui, permet d'aisément construire une API à travers un framework complet et orienté objet. En effet, les différentes classes que **QT** met à disposition permettent une implémentation rapide et sereine via de nombreux objets tel qu'un **QImage**. **QT** permet également un déploiement aisé sur de multiples plateformes si nécessaire.

## La suite

Il serait intéressant de pouvoir ajouter des fonctionnalités supplémentaires telles que,

- Le rendu en temps réel, sans devoir passer par une action de demande de traitement. Des mécanismes de gestion de thread séparé ont été ajoutés mais ne sont pas implémentés.
- La gestion des vidéos serait également une fonctionnalité intéressante, OpenCV en est totalement capable mais il faut prévoir les mécanismes adaptés au sein de l'API pour le chargement, la modification et la sauvegarde de la vidéo importée. Le rendu en temps réel serait également très intéressant pour cette fonctionnalité.
- Proposer des filtres supplémentaires serait un plus, notamment des filtres plus intelligents (p.ex: utilisation du framework CUDA AI) mais peut être coûteux en intégration, dépendant du type de filtre à ajouter
- Proposer différents types de format lors de la sauvegarde du fichier ainsi que des paramètres de compression
- Il reste quelques cas d'erreurs non gérés (vérification du format du fichier importé, nom du fichier ne peut contenir que des caractères ASCII, ...)
- Gérer des traitements plus basiques de l'image comme la luminosité ou le contraste

## Présentation

**ImageDenoizerAPI** est le middleware qui fournit des fonctions de réductions de bruit permettant d'améliorer une image et ceci de manière rapide et aisément intégrable dans tous vos projets favoris. Il expose une interface intuitive possédant des actions de chargement d'image, de traitement du bruit et de sauvegarde. La réduction de bruit est possible via plusieurs types de filtres et les paramètres permettent d'obtenir le résultat escompté. Ce middleware est extensible et peut être exécuté sur un thread séparé, permettant une utilisation fluide car l'exécution sera traitée en arrière-plan de votre application.