

# Introduction à la recherche

Dans cette dernière partie du cours vous allez devoir lire, comprendre, synthétiser et expérimenter sur un article de recherche associé à un code source. Le but est de découvrir une nouvelle méthode de traitement d'image présentée sous la forme d'un article de recherche, de s'approprier le code source publié par les auteurs et de reproduire une partie des expériences présentées dans l'article.

## ❗ Important

Vous travaillerez en monome ou par groupe de 2 ou 3. Vous devrez rendre, sur votre dépôt git au plus tard à la date indiquée par mail :

1. Un rapport (format pdf),
2. Le code source vous ayant permis de réaliser les expériences présentées dans votre rapport.

**Plagiat :** La récupération de tout ou portion du travail d'une tierce personne, qu'il s'agisse d'un étudiant ou autre, provenant d'un site internet, d'une vidéo ou tout autre support, n'est permise qu'avec la permission de l'auteur, ce dernier devant être cité dans le code et dans le rapport. Un travail utilisé sans référence à son auteur constitue un plagiat. Le plagiat est interdit par le règlement intérieur de l'école (Article 3). En cas de plagiat, le conseil de discipline peut prononcer des sanctions allant jusqu'à l'exclusion définitive de l'étudiant.

## Organisation du rapport

---

Le rapport comportera 3 parties :

1. Synthèse de l'article
2. Etude de l'implémentation
3. Expérimentations

## Synthèse de l'article

Cette partie est un résumé critique de l'article (longueur : environ 6000 caractères). Vous donnerez en particulier les éléments suivants :

- le contexte : quelle est la problématique générale abordée ?
- les objectifs : quel est le but de la méthode proposée dans l'article ?
- les hypothèses : pourquoi l'approche proposée est-elle pertinente pour atteindre les objectifs ?
- la méthode : comment fonctionne la méthode proposée ?
- méthodologie de validation : quels sont les tests proposés par les auteurs pour valider leurs hypothèses ? pour valider la méthode ?
- les résultats : quels sont les résultats des tests ?
- opinion personnelle : quel est votre impression personnel sur cet article ? (intérêt, qualité d'écriture, des hypothèses, de la méthode, des tests...)

## Etude de l'implémentation

Cette partie a pour but d'étudier l'implémentation fournie par les auteurs ou un tiers (longueur : environ 3000 caractères). Vous donnerez en particulier les éléments suivants :

- la portée des programmes fournis : le code fourni permet-il de reproduire la totalité des expériences présentées dans l'article ?
- la structure : quelles sont les interfaces fournies ? le code est-il bien structuré ? pensez-vous pouvoir le réutiliser dans une autre contexte ?
- la fidélité : le code fourni correspond-il à ce qui est décrit dans l'article ? Si non, quelles sont les différences notables ?

## Expérimentations

Cette partie a pour but de vérifier que tout où partie des expériences réalisées dans l'article sont reproductibles (longueur : environ 3000 caractères). Vous donnerez en particulier les éléments suivants :

- expériences reproduites : décrivez les expériences que vous avez choisies de reproduire.
- configuration de tests : précisez les conditions d'expérimentation (choix des images, des paramètres).
- analyse des résultats : analysez les résultats obtenus, sont-ils conformes aux résultats présentés dans l'article ?
- expériences complémentaires : pouvez-vous proposer des expériences complémentaires pour évaluer la méthode ?

# Code rendu

---

Votre dépôt devra contenir (en plus du rapport) tout le code utilisé pour préparer la partie *Expérimentations* du rapport. On devra en particulier trouver :

- un fichier *ReadMe* détaillant
  - le contenu du dépôt
  - expliquant comment utiliser les scripts et exécutables
  - expliquant la procédure de *build* (si applicable)
- des scripts et/ou exécutables pour générer les résultats et figures montrés dans la partie *Expérimentations* du rapport,
- tous les codes sources utilisés avec référence à leurs auteurs.

## Articles proposés

---

Vous trouverez ci-dessous la liste des articles proposés. Vous pouvez proposer d'autres articles, sous réserve de validation.

### ! Note

Certains articles sont plus courts que d'autres. Au vu des délais donnés, il n'est pas demandé d'atteindre une compréhension intégrale des articles les plus longs. Le corrolaire étant que la compréhension attendue des articles les plus courts est plus poussée que celle des articles les plus longs (le temps de travail attendu est équivalent).

- SLIC superpixels, découpage régulé et adaptatif de l'image : [📄 article](#) [code](#).
- SEAM carving, redimensionnement intelligent d'images : [📄 article](#) [code](#) (parmi beaucoup d'autres).
- Non-local means, filtre de débruitage très performant : [📄 article](#) [code](#).
- Guided filters, bilateral filters nouvelle génération : [📄 article](#) [code](#).
- Image foresting transform, algorithme de segmentation d'images basé sur l'algorithme de Dijkstra: [📄 article](#) [code](#).
- Efficient Graph-Based Image Segmentation, algorithme de segmentation d'images : [📄 article](#) [code](#).
- Extreme image completion, reconstitution d'images à partir d'un très faible nombre de pixels : [📄 article](#) [code](#).
- Interactive segmentation, sélection interactive d'objets dans une image : [📄 article](#) [code](#).
- Détection de structure linéaire 3D, détection des vaisseaux sanguins en imagerie médicale : [📄 article](#) [code](#).
- LSD, détection de segments avec une approche *a contrario* : [code](#) et [article](#).
- MTO, détection d'objets à très faible brillance de surface en imagerie astronomique : [article](#) [code](#).
- Waterpixels, découpage régulé et adaptatif de l'image : [article](#) [code](#).
- PatchMatch, édition d'image structurelle : [code](#) et [article](#) [code](#) (autre).

- Exposure Fusion, high dynamic range image : [article](#) et [code](#).