

Motivations du projet

Biomimétisme

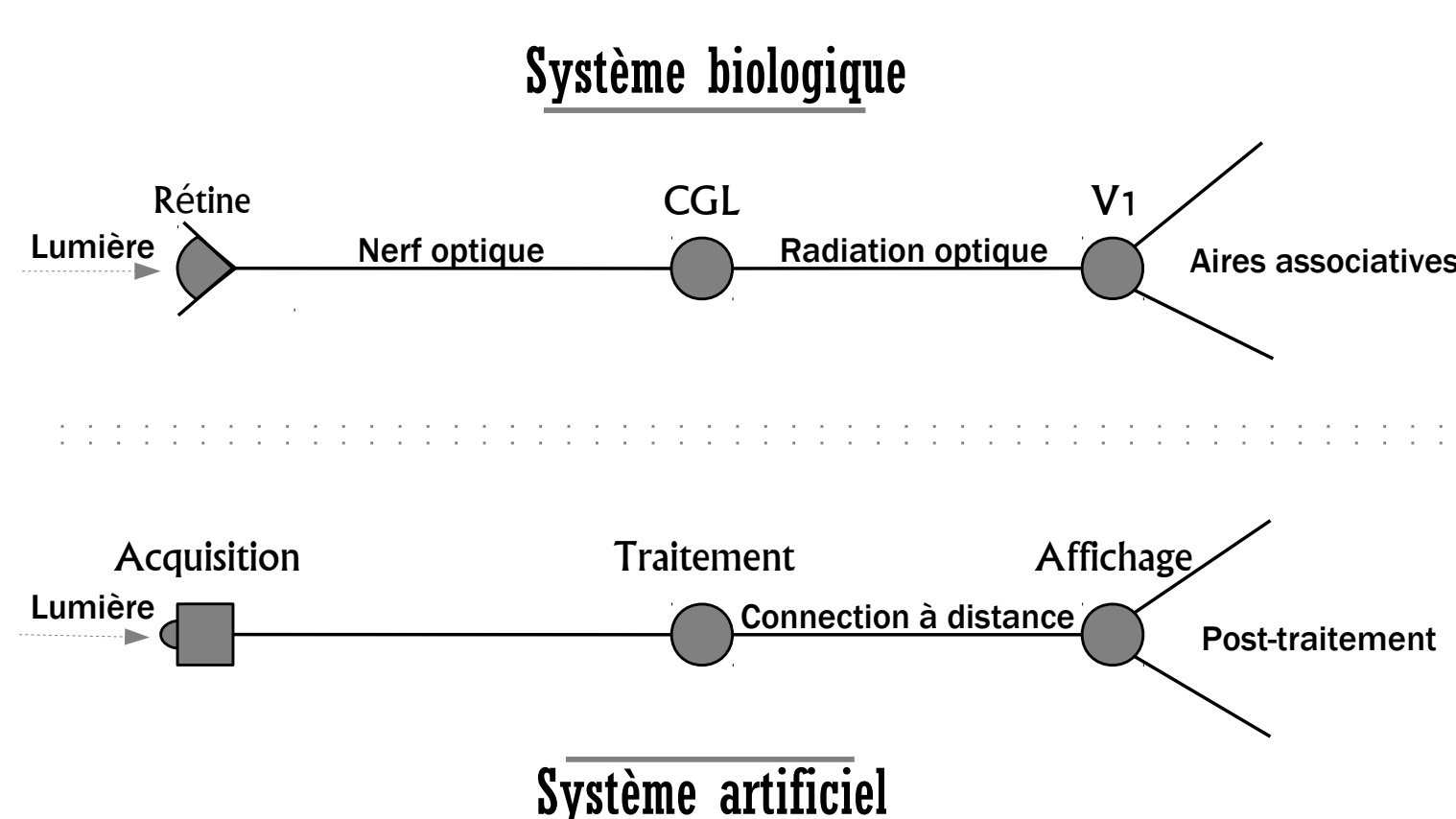
Qu'est-ce que la biomimétique ?

La biomimétique consiste en la reproduction de fonctions et de capacités des systèmes biologiques dans des systèmes artificiels afin d'améliorer leurs performances.

L'objectif est donc de s'inspirer de ce que la sélection naturelle a permis d'obtenir au cours de l'évolution pour résoudre des problèmes d'ingénierie limitant les systèmes artificiels actuels : résistance et adaptabilité aux contraintes environnementales, collecte et utilisation de l'énergie dans l'environnement, autonomie dans les déplacements et les prises de décisions, ...

L'ingénierie biomimétique est donc souvent utilisé dans le développement de technologies de pointe comme la robotique, mais aussi dans la recherche car il permet de mieux comprendre les mécanismes biologiques étudiés.

Dans le cadre de ce projet, c'est la rétine et le système visuel animal qui servent de modèle.



Low-Tech

L'une des limites actuelles de l'ingénierie touche à la puissance de calcul des systèmes utilisés.

L'ingénierie "high-tech" cherche à créer les systèmes les plus performants possibles sans se soucier de la consommation d'énergie et de puissance de calcul. En réponse, l'ingénierie "low-tech" se concentre sur l'intégration de programmes dans des systèmes artificiels possédant une faible puissance de calcul ou une autonomie énergétique limitée, tels que les systèmes embarqués (drones, systèmes robotisés autonomes) ou répondant à des contraintes bien précises, tels que les rétines artificielles.

Dans le cadre de OpenRetina, c'est une ingénierie low-tech et Open Source qui est visée, afin de permettre son adaptation et son utilisation par le plus de systèmes possible.

Méthodes computationnelles

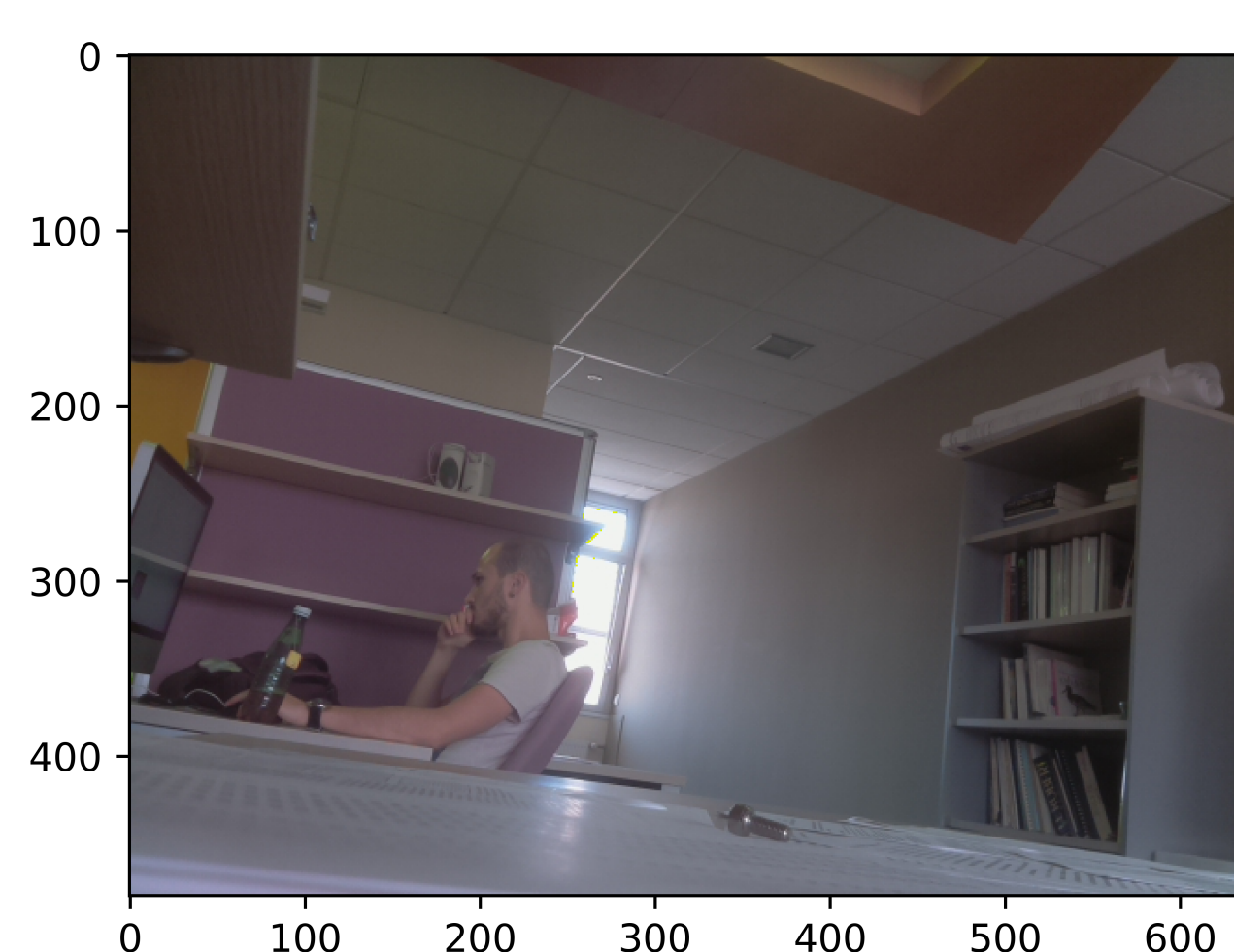
Le projet OpenRetina est composé d'un ensemble de programmes majoritairement écrits dans le langage de programmation Python.

Ces scripts réalisent une série de tâches qu'on peut décrire en 4 catégories, ci-dessous :

Grab

Récupérer une image ou une vidéo à partir d'un fichier video, d'une caméra locale ou d'une caméra distante (connectée à une Raspberry Pi).

Cette étape est proche d'une prise d'image classique mais sert surtout à récupérer les informations enregistrées individuellement dans chaque pixel.



Process

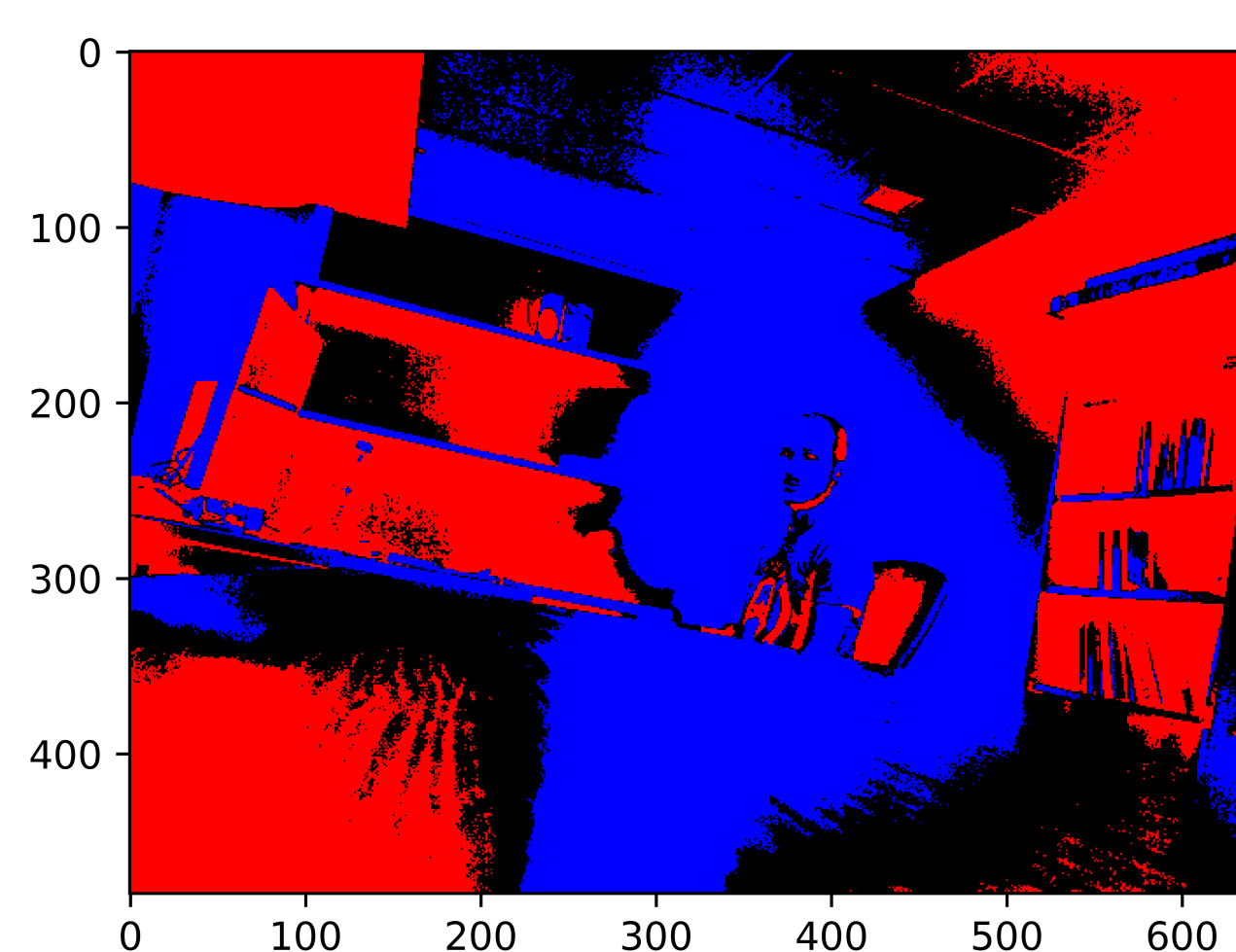
Réaliser un traitement sur chaque frame que comporte l'image, afin de réduire la quantité d'informations transmises et d'accélérer la temporalité de la perception.

Network

Transmettre les informations (requêtes ou données) entre les acteurs.

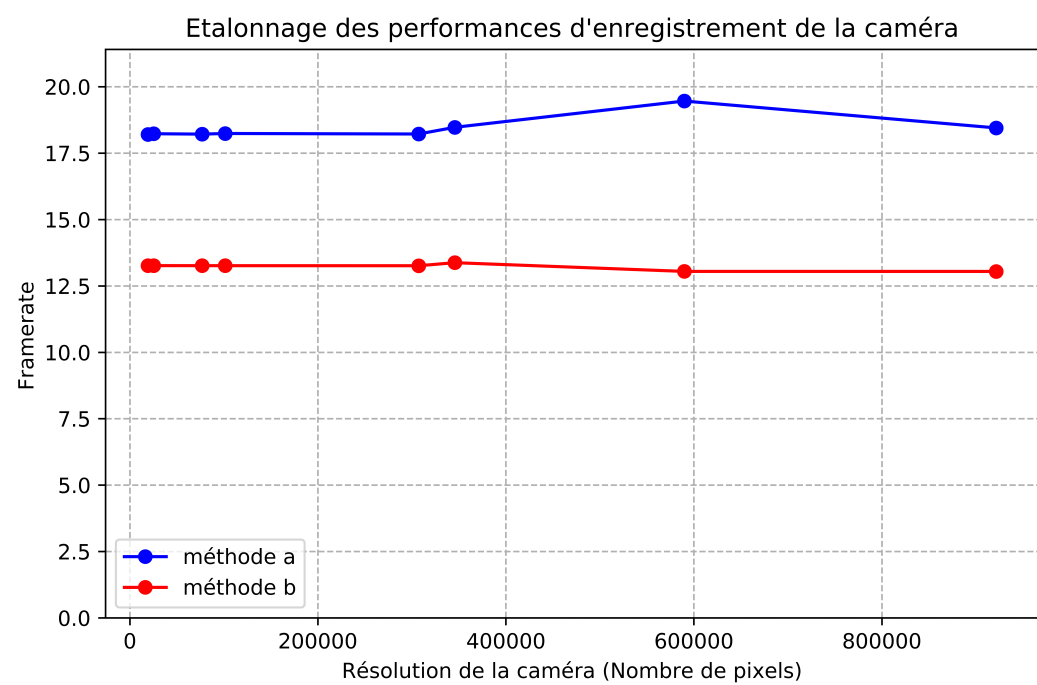
Display

Afficher l'image nouvellement traitée afin de contrôler les résultats.

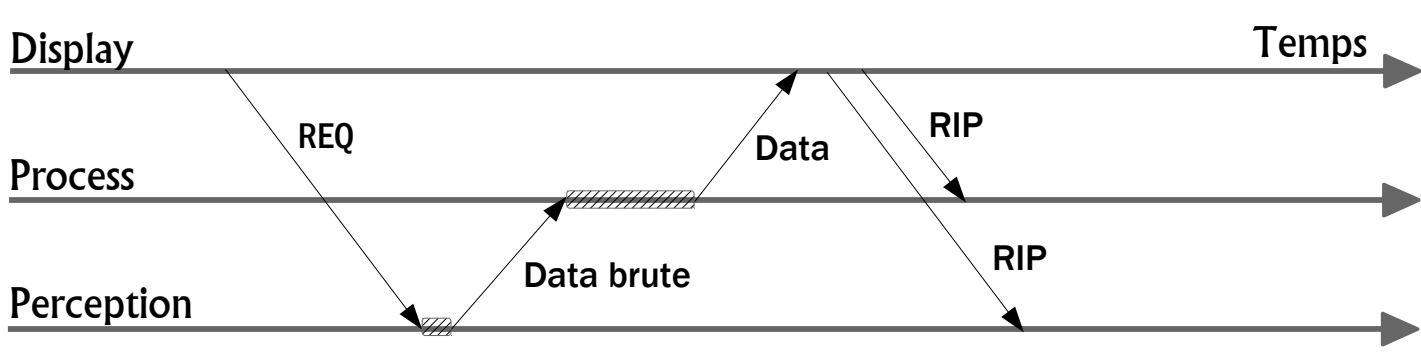


Améliorations apportées au projet

- Travailler avec des videos pour réaliser des traitements hors-ligne
- Réaliser un "benchmark" pour contrôler l'influence de certains paramètres



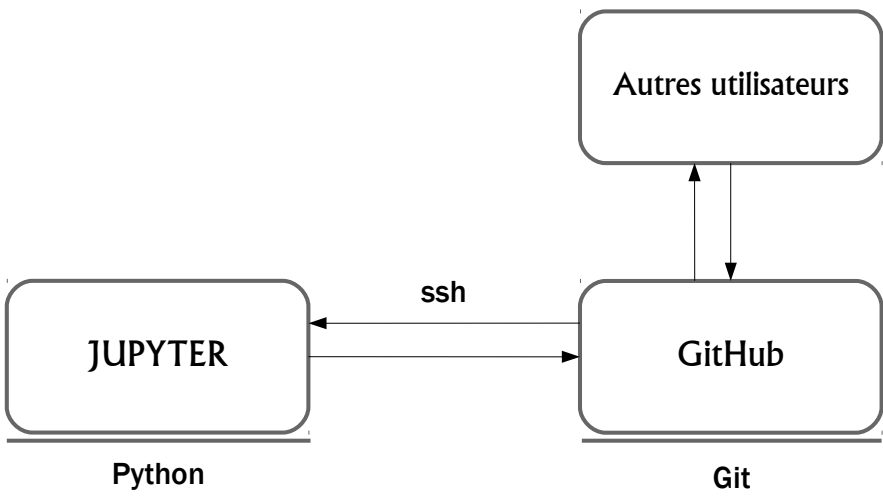
- Architecture en couches



Compétences acquises

- Methodologie (python, notamment OOP; utilisation et maintien d'un logbook; utilisation

de git; communications à distance via les protocoles ssh, approfondissement des connaissances en L^AT_EX)



- Méthodes de traitement d'images et de vidéos
- Compétences d'ingénierie et de reverse-engineering.
- Connaissances neuroscientifiques dans les domaines de la perception et du traitement de l'information visuelle.