

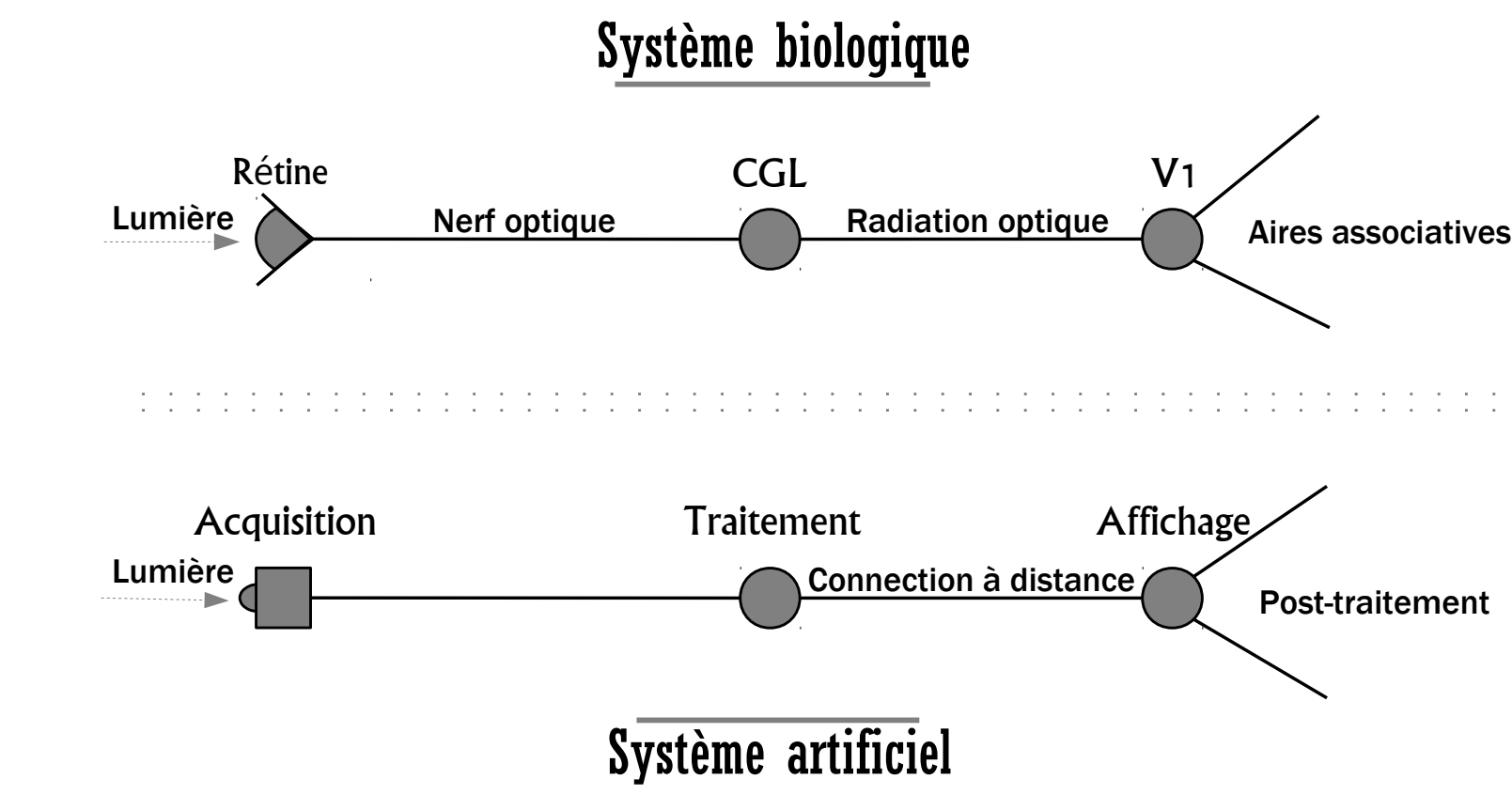
Motivations du projet

Biomimtisme

Qu'est-ce que la biomimtique?  
La biomimtique consiste en la reproduction de fonctions et de capacit s des syst mes biologiques dans des syst mes artificiels afin d'amliorer leurs performances.  
L'objectif est donc de s'inspirer de ce que la s lection naturelle a permis d'obtenir au cours de l'volution pour rsoudre des probl mes d'ing nierie limitant les syst mes artificiels actuels : rsistance et adaptabilit  aux contraintes environnementales, collecte et utilisation de l' nergie dans l'environnement, autonomie dans les d placements et les prises de dcisions, ...

L'ing nierie biomimtique est donc souvent utilis e dans le d veloppement de technologies de pointe comme la robotique, mais aussi dans la recherche car il permet de mieux comprendre les m canismes biologiques tudies.

Dans le cadre de ce projet, c'est la rtine et le syst me visuel animal qui servent de mod le.



Low-Tech

L'une des limites actuelles de l'ing nierie touche la puissance de calcul des syst mes utilis s.

L'ing nierie "high-tech" cherche   cr er les syst mes les plus performants possibles sans se soucier de la consommation d' nergie et de puissance de calcul.  
En r ponse, l'ing nierie "low-tech" se concentre sur l'int gration de programmes dans des syst mes artificiels poss dant une faible puissance de calcul ou une autonomie  nerg tique limite, tels que les syst mes embarqu s (drones, syst mes robotiss s autonomes) ou r pondant   des contraintes bien pr cises, tels que les rtines artificielles.

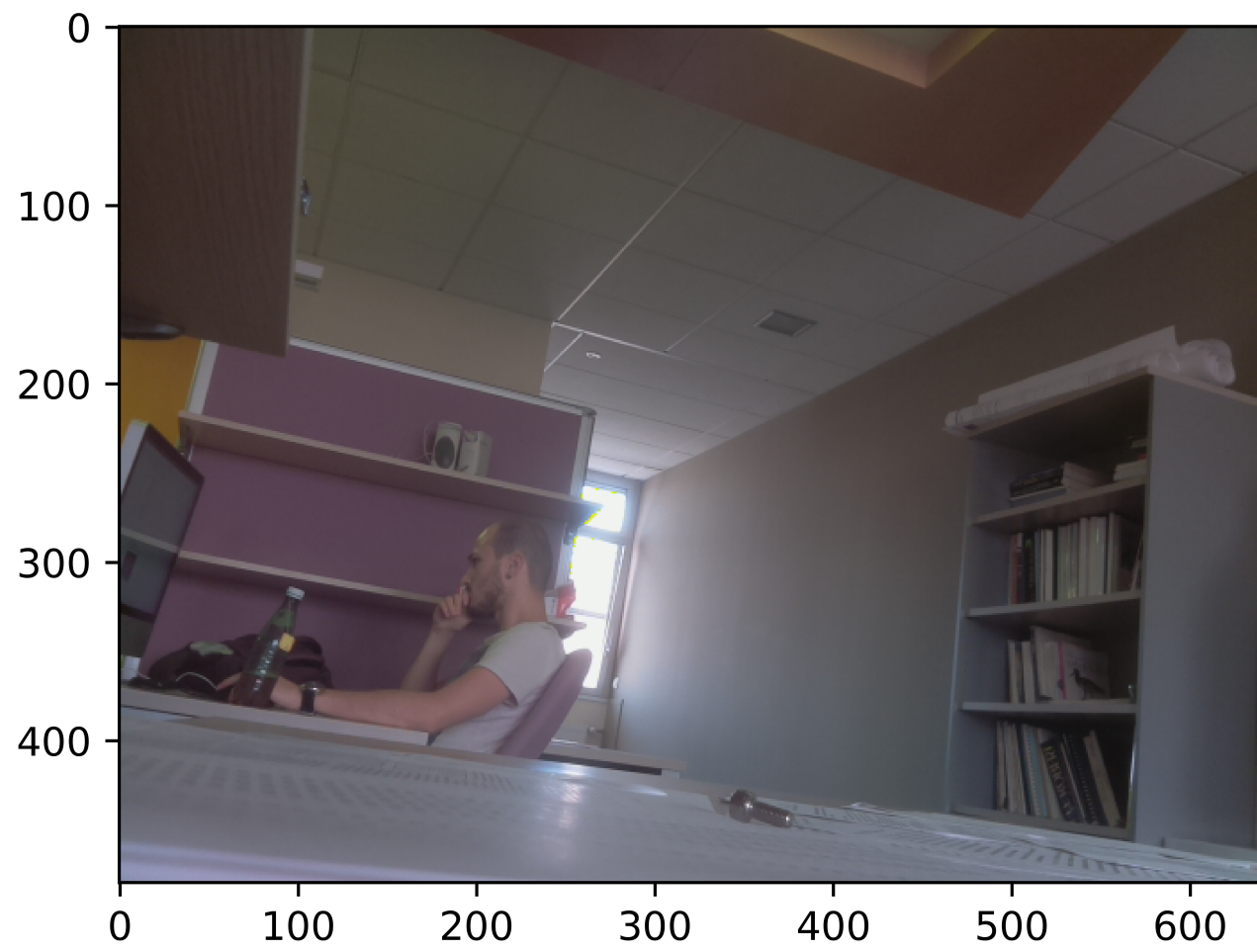
Dans le cadre de OpenRetina, c'est une ing nierie low-tech et Open Source qui est visee, afin de permettre son adaptation et son utilisation par le plus de syst mes possible.

M thodes computationnelles

Le projet OpenRetina est compos  d'un ensemble de programmes majoritairement crits dans le langage de programmation Python.  
Ces scripts r alisent une s rie de t ches qu'on peut d crire en 4 cat gories, ci-dessous :

Grab

Rcuprer une image ou une vido   partir d'un fichier video, d'une cam ra locale ou d'une cam ra distante (connect e   une Raspberry Pi).  
Cette tape est proche d'une prise d'image classique mais sert surtout   rcuprer les informations enregistr es individuellement dans chaque pixel.



Process

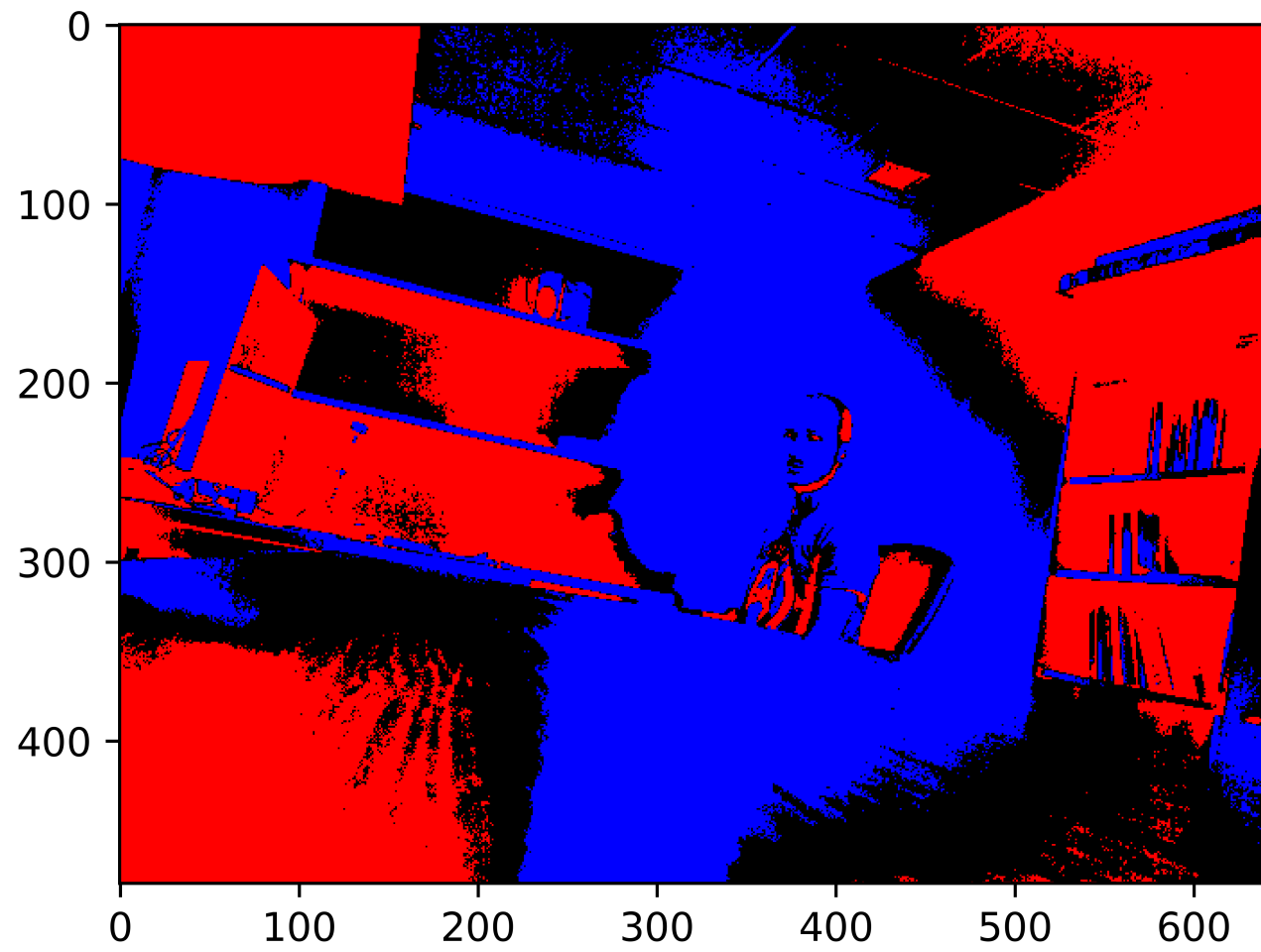
R aliser un traitement sur chaque frame que comporte l'image, afin de r duire la quantit  d'informations transmises et d'acc l rer la temporalit  de la perception.

Network

Transmettre les informations (requ tes ou donn es) entre les acteurs.

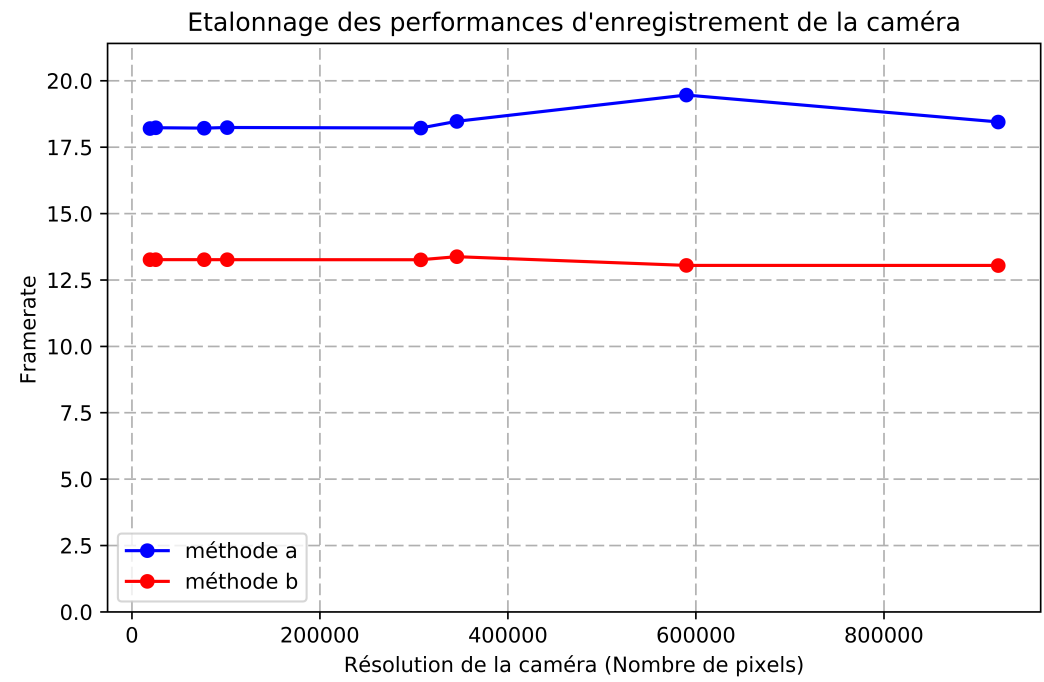
Display

Afficher l'image nouvellement trait e afin de contr ler les rsultats.

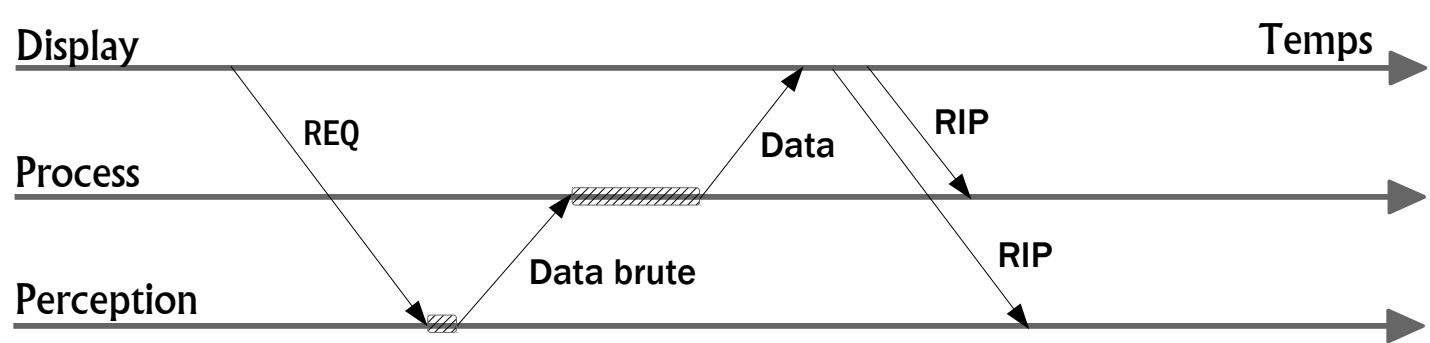


Am liorations apport es au projet

- Travailler avec des videos pour r aliser des traitements hors-ligne
- R aliser un "benchmark" pour contr ler l'influence de certains param tres

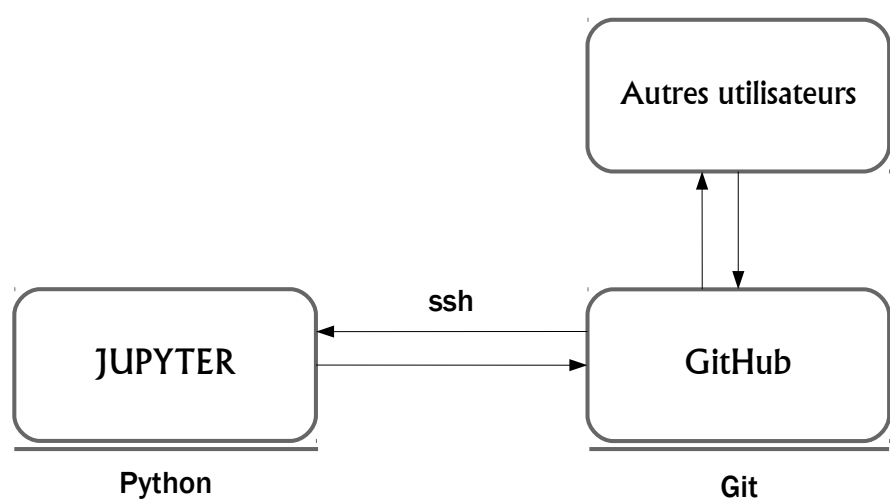


- Architecture en couches



Comp tences acquises

- Methodologie (python, notamment OOP; utilisation et maintien d'un logbook; utilisation de git; communications   distance via les protocoles ssh, approfondissement des connaissances en B t x)



- M thodes de traitement d'images et de vidos
- Comp tences d'ing nierie et de reverse-engineering.
- Connaissances neuroscientifiques dans les domaines de la perception et du traitement de l'information visuelle.