



Ecole Polytechnique de l'Université de Tours

Département Informatique

64 avenue Jean Portalis

37200 Tours, France

Tél. +33 (0)2 47 36 14 14

[polytech.univ-tours.fr](http://polytech.univ-tours.fr)

**Projet Recherche & Développement**

**2020-2021**

# Mesures multiples et rapides de sarcomères



**POLYTECH<sup>®</sup>**  
**TOURS**

**Entreprise**

UFR des Sciences Pharmaceutiques de Tours



**Tuteur entreprise**

Côme PASQUALIN

**Étudiant**

Lucas DOS SANTOS (DI5)

**Tuteur académique**

Donatello CONTE

# Liste des intervenants

## Entreprise

UFR des Sciences Pharmaceutiques de Tours  
31 Avenue Monge  
37200 Tours, France  
[pharma.univ-tours.fr](mailto:pharma.univ-tours.fr)



Nom	Email	Qualité
Lucas DOS SANTOS	<a href="mailto:lucas.dossantos@etu.univ-tours.fr">lucas.dossantos@etu.univ-tours.fr</a>	Étudiant DI5
Donatello CONTE	<a href="mailto:donatello.conte@univ-tours.fr">donatello.conte@univ-tours.fr</a>	Tuteur académique, Département Informatique
Côme PASQUALIN	<a href="mailto:come.pasqualin@univ-tours.fr">come.pasqualin@univ-tours.fr</a>	Tuteur entreprise



# Avertissement

Ce document a été rédigé par Lucas Dos Santos susnommé l'auteur.

L'entreprise UFR des Sciences Pharmaceutiques de Tours est représentée par Côme Pasqualin susnommé le tuteur entreprise.

L'Ecole Polytechnique de l'Université de Tours est représentée par Donatello Conte susnommé le tuteur académique.

Par l'utilisation de ce modèle de document, l'ensemble des intervenants du projet acceptent les conditions définies ci-après.

L'auteur reconnaît assumer l'entière responsabilité du contenu du document ainsi que toutes suites judiciaires qui pourraient en découler du fait du non respect des lois ou des droits d'auteur.

L'auteur atteste que les propos du document sont sincères et assume l'entière responsabilité de la véracité des propos.

L'auteur atteste ne pas s'approprier le travail d'autrui et que le document ne contient aucun plagiat.

L'auteur atteste que le document ne contient aucun propos diffamatoire ou condamnable devant la loi.

L'auteur reconnaît qu'il ne peut diffuser ce document en partie ou en intégralité sous quelque forme que ce soit sans l'accord préalable du tuteur académique et de l'entreprise.

L'auteur autorise l'école polytechnique de l'université François Rabelais de Tours à diffuser tout ou partie de ce document, sous quelque forme que ce soit, y compris après transformation en citant la source. Cette diffusion devra se faire gracieusement et être accompagnée du présent avertissement.



# Pour citer ce document

Lucas Dos Santos, *Mesures multiples et rapides de sarcomères* : , Projet Recherche & Développement, Ecole Polytechnique de l'Université de Tours, Tours, France, 2020-2021.

```
@mastersthesis{
  author={Dos Santos, Lucas},
  title={Mesures multiples et rapides de sarcomères: },
  type={Projet Recherche \& Développement},
  school={Ecole Polytechnique de l'Université de Tours},
  address={Tours, France},
  year={2020-2021}
}
```



# Table des matières

Liste des intervenants	<b>a</b>
Avertissement	<b>b</b>
Pour citer ce document	<b>c</b>
Table des matières	<b>i</b>
Table des figures	<b>iv</b>
<b>1 Remerciements</b>	<b>1</b>
<b>2 Introduction</b>	<b>2</b>
1 Acteurs, enjeux et contexte .....	2
2 Objectifs.....	2
3 Hypothèses .....	2
3.1 Solution 1 : Parallélisation des calculs .....	3
3.2 Solution 2 : Détection de contours d'un sarcomère .....	3
4 Gestion de projet .....	3
<b>3 Description générale</b>	<b>5</b>
1 Environnement du projet .....	5
2 Caractéristiques des utilisateurs .....	5
3 Fonctionnalités du système .....	5
4 Structure générale du système.....	6
4.1 Solution 1 : Parallélisation des calculs .....	6
4.2 Solution 2 : Détection de contours d'un sarcomère .....	7

<b>4</b>	<b>Etat de l'art</b>	<b>8</b>
1	ImageJ : SarcOptim .....	8
2	La technologie CUDA .....	9
3	Parallélisation des calculs.....	9
	JCuda .....	9
	Python.....	10
	Etude comparative .....	10
4	Analyse d'images : Détection de contours.....	10
<b>5</b>	<b>Analyse et conception</b>	<b>11</b>
1	Analyse .....	11
1.1	Hypothèses utilisées .....	11
1.2	Spécifications.....	11
2	Modélisation proposée.....	12
<b>6</b>	<b>Bilan et conclusion</b>	<b>13</b>
1	Bilan du semestre 9 .....	13
	<b>Annexes</b>	<b>14</b>
<b>A</b>	<b>Planification, gestion de projet</b>	<b>15</b>
1	Evolution du projet .....	15
2	Description des tâches.....	16
	Réunion 1 : Rendez-vous M. Conte .....	16
	Tâche 1 : Découverte du sujet .....	16
	Réunion 2 : Rendez-vous Côte Pasqualin.....	16
	Tâche 2 : Prise en main de ImageJ .....	16
	Réunion 3 : Rendez-vous Côte Pasqualin.....	16
	Tâche 3 : Analyse des technologies existantes .....	16
	Tâche 4 : Analyse des script ImageJ .....	16
	Tâche 5 : Cahier de spécifications .....	16
	Tâche 6 : Rédaction du rapport .....	17
	Tâche 7 : Tests des différentes technologies .....	17
<b>B</b>	<b>Description des interfaces</b>	<b>18</b>
1	Interface materielle/logicielle.....	18
2	Interface homme/machine .....	18
<b>C</b>	<b>Cahier de Specifications</b>	<b>19</b>
1	Spécifications fonctionnelles .....	19

1.1	Définition de la fonction de traitement : Mesurer la variation de taille d'un sarcomère .....	19
	Description de la fonction : .....	19
2	Spécifications non fonctionnelles .....	20
2.1	Contraintes de développement et conception .....	20
2.2	Contraintes de fonctionnement.....	20
2.2.1	Performances.....	20
<b>D</b>	<b>Bibliographie</b>	<b>21</b>
<b>E</b>	<b>Glossaire</b>	<b>22</b>
<b>F</b>	<b>Acronymes</b>	<b>23</b>



# Table des figures

## 2 Introduction

2.1 Trello.....	3
2.2 Diagramme de Gantt .....	4
3.1 Schémas du système.....	5
3.2 Diagramme de classe : CUDA .....	6
3.3 Diagramme de classe : Détection de contours .....	7

## 4 Etat de l'art

4.1 Plugin SarcOptim .....	8
4.2 CUDA .....	9

## 5 Analyse et conception

5.1 Technologie CUDA .....	12
5.2 Analyse d'images .....	12

## A Planification, gestion de projet

A.1 Outil Trello.....	15
A.2 Diagramme de Gantt final .....	15

# 1

## Remerciements

Premièrement, je tiens à remercier mon encadrant, Donatello Conte, pour son suivi ainsi que sa réactivité. En effet, M. Conte a toujours suivi l'avancée de ce projet mais m'a également permis, durant le confinement, d'accéder à un ordinateur équipé d'une carte graphique NVIDIA, essentielle à cette phase d'analyse du besoin.

Je souhaite également remercier Côme Pasqualin ainsi que son collègue François Gannier pour leur investissement tout au long de ce projet. Effectivement, étant très disposés à me répondre, j'ai pu mieux comprendre leur besoin mais aussi mieux comprendre leur méthodologie.

# 2

## Introduction

### 1 Acteurs, enjeux et contexte

Côme Pasqualin étudie les **sarcomères** [3] qui sont des unités contractiles dans les cellules cardiaques. La mesure précise des contractions de cette cellule est importante pour les scientifiques qui travaillent sur les maladies cardiaques. Afin d'étudier les sarcomères, ils utilisent le logiciel **ImageJ** sur lequel ils ont développé un plugin : [2]. Ce plugin leur permet de travailler sur une vidéo préalablement enregistrée et d'effectuer une **Transformée de Fourier (FFT)** sur chaque image de celle-ci, afin d'en déduire la variation de taille d'un sarcomère.

Le principal défaut de ce plugin est qu'il permet d'étudier un seul sarcomère à la fois sur une vidéo préalablement enregistrée. Aujourd'hui, il souhaiterait travailler en temps réel et sur plusieurs sarcomères en même temps.

### 2 Objectifs

L'objectif de ce projet est donc de permettre aux chercheurs d'étudier les sarcomères en temps réel mais aussi de pouvoir travailler sur plusieurs sarcomères en parallèle. En effet, ils veulent pouvoir brancher leur caméra directement sur leur machine et le logiciel leur permettra de calculer les variations de taille des sarcomères sans pour autant enregistrer la vidéo et ainsi économiser du temps dans leurs recherches.

### 3 Hypothèses

Lors de la phase de découverte du sujet, nous avons envisagé deux types de solutions pour ce projet :

- Parallélisation des calculs
- Détection de contours d'un sarcomère

### 3.1 Solution 1 : Parallélisation des calculs

Aujourd'hui le plugin SarcOptim fonctionne très bien avec une seule cellule. Cependant, travailler sur plusieurs sarcomères en parallèle nécessite une plus grande puissance de calcul car une FFT doit être appliquée sur tous les sarcomères en temps réel.

Cette solution a donc pour objectif de réduire le temps de calculs, via la technologie **CUDA** [1], en les effectuant directement sur processeur graphique : **Graphics Processing Unit (GPU)**. Cette solution peut être envisageable car elle permet de construire une nouvelle brique sur le logiciel préexistant.

### 3.2 Solution 2 : Détection de contours d'un sarcomère

Cette solution se base sur l'analyse d'images, elle nous permet de réduire considérablement le temps de calcul. En effet, si nous parvenons à détecter les contours de chaque cellule, il est possible de suivre le déplacement des pixels. Cependant, cette méthode ne permet pas directement d'interagir avec ImageJ. Néanmoins, il est possible de réaliser un script afin d'exécuter l'algorithme créé et afficher les résultats directement sur celui-ci.

## 4 Gestion de projet

Afin de mener à bien ce projet, j'ai tout d'abord utilisé l'outil Trello. Il m'a permis de lister toutes les tâches à effectuer par semaine, de détailler chaque tâche en sous-tâches ou encore de pouvoir planifier des tâches telles que les réunions.

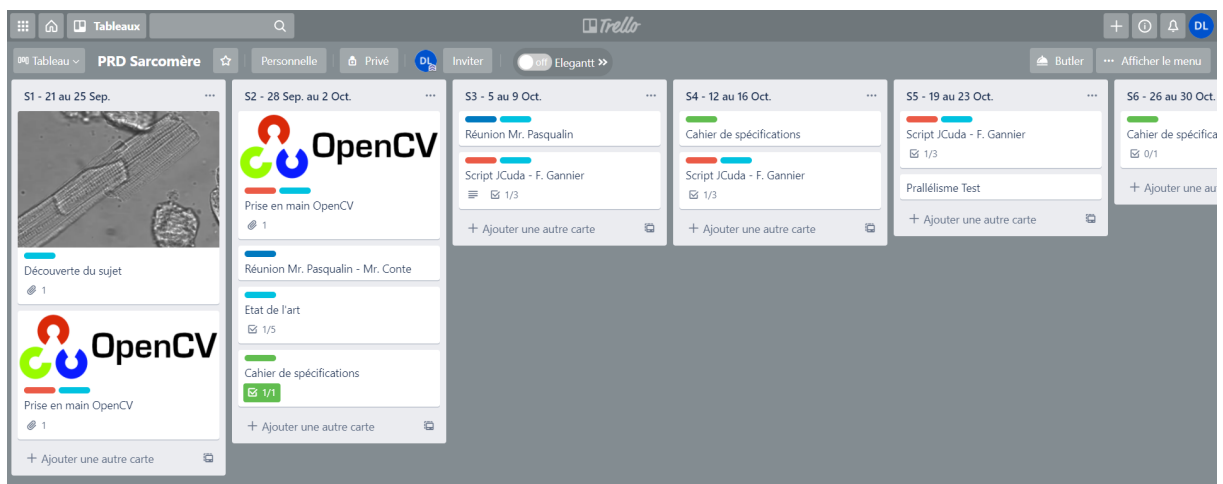


Figure 2.1 – Trello

J'ai également utilisé le logiciel Gantt Project qui m'a permis de me fixer des objectifs à atteindre et de pouvoir rectifier le planning en fonction des nouvelles tâches qui ont été rajoutées au cours du projet. Ce diagramme m'a été très utile pour garder une vision globale de l'avancée du projet.

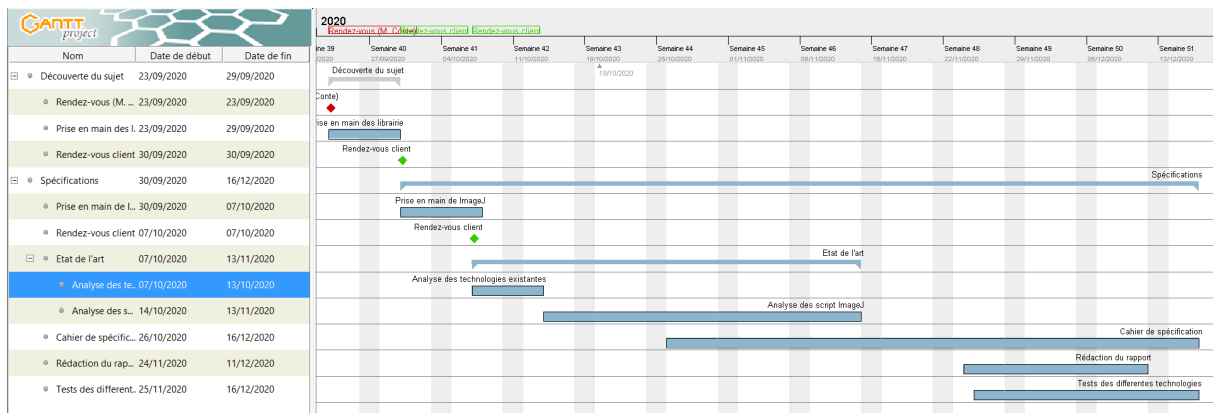


Figure 2.2 – Diagramme de Gantt

De plus, j'ai fourni un rapport hebdomadaire par mail à mon encadrant pour qu'il puisse suivre l'avancement du projet.

# 3

## Description générale

### 1 Environnement du projet

Actuellement, les chercheurs utilisent le logiciel ImageJ, c'est pourquoi l'application développée doit pouvoir interagir avec ce logiciel. De plus ce projet dépend d'un certain type de matériel, l'utilisation de la technologie CUDA nécessite une carte graphique NVIDIA. Si le premier type de solution est retenu, il est indispensable pour l'utilisateur d'en posséder une.

Les langages utilisés pour mener à bien ce projet sont le langage Python et le langage Java. ImageJ étant développé en Java, ce langage nous permettra une interaction entre le logiciel et les scripts Python si une des deux solutions est retenue.

### 2 Caractéristiques des utilisateurs

Les utilisateurs de cette application seront les chercheurs. Pour utiliser ce logiciel, une connaissance du domaine d'étude est nécessaire afin de comprendre les résultats retournés par celui-ci. En revanche une connaissance de l'informatique n'est pas nécessaire.

### 3 Fonctionnalités du système



Figure 3.1 – Schémas du système

Le but de cette application est de mesurer les variations de taille d'un ou plusieurs sarcomères. Le système dispose d'un flux vidéo entrant, il doit dans un premier temps détecter les sarcomères puis effectuer un traitement afin d'en déduire la variation de taille de ces cellules. Pour finir le logiciel doit retourner le résultat du traitement sous forme de courbes.

## 4 Structure générale du système

Afin de réaliser cette application, deux types de solutions sont envisagés. J'ai donc réalisé deux diagrammes de classes car la méthode de calcul diffère en fonction la solution choisie.

### 4.1 Solution 1 : Parallélisation des calculs

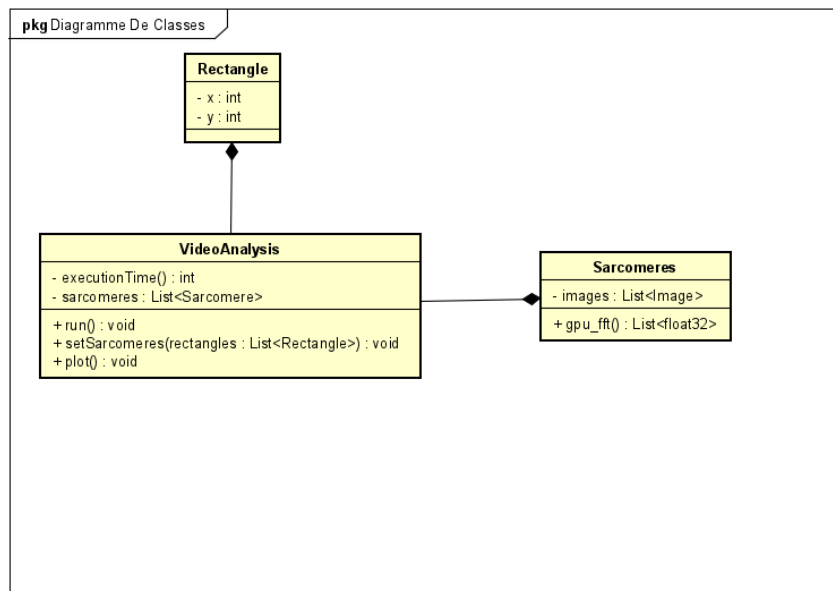


Figure 3.2 – Diagramme de classe : CUDA

Ce diagramme de classes décrit les classes nécessaires afin d'effectuer les calculs via la technologie CUDA. En effet, pour cette solution, l'utilisateur doit définir les contours d'un sarcomère, symbolisés par un rectangle, directement sur le flux vidéo. De plus, la classe *VideoAnalysis* se charge de convertir les objets *Rectangle* en sous-images pour chaque image de la vidéo.

Pour finir, pour chaque image de la vidéo, la fonction *gpu\_fft()* est appelée pour effectuer un calcul de FFT locale, c'est-à-dire, sur chaque sous image.

4.2 Solution 2 : Détection de contours d'un sarcomère

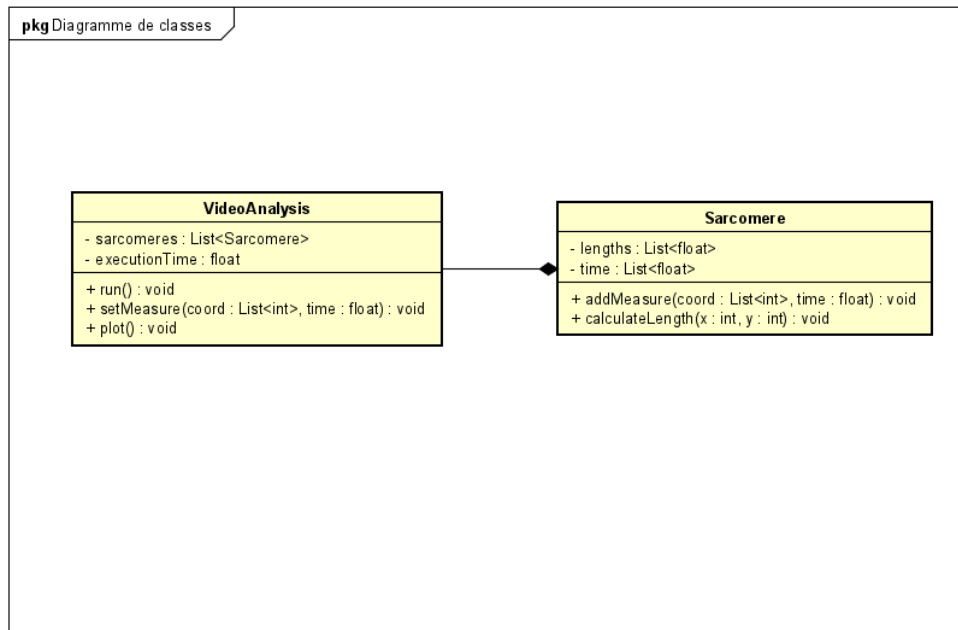


Figure 3.3 – Diagramme de classe : Détection de contours

Dans cette deuxième solution, la classe *VideoAnalysis*, se charge de détecter les contours d'un sarcomère dans la fonction *run()*. Lors de la première exécution, l'application se charge de créer des objets *Sarcomere* à l'aide de deux positions *x*, *y* et *time* : le temps d'exécution du programme. Par la suite, ces valeurs sont ajoutées dans les tableaux de la classe *Sarcomere*. Ainsi, grâce à cette classe nous récupérerons toutes les tailles successives de nos sarcomères à un temps d'exécution donné.

Pour finir, la fonction *plot()* se charge de tracer les graphiques de variation de taille en fonction du temps pour chaque sarcomère.

# 4

## Etat de l'art

### 1 ImageJ : SarcOptim

Une première phase de prise en main du logiciel ImageJ a été nécessaire afin de bien comprendre le fonctionnement du plugin SarcOptim. Ce plugin a été développé par Côme Pasqualin ainsi que son collègue François Ganier. Il était impératif de bien comprendre son fonctionnement pour pouvoir développer une solution adaptée.

Pour rappel, ce plugin effectue les calculs sur une vidéo enregistrée comportant un seul sarcomère. Pour ce faire, les chercheurs tracent une ligne de référence directement sur la cellule et l'algorithme se charge d'appliquer une FFT sur chaque image de la vidéo. Une fois les calculs effectués, l'algorithme retourne deux courbes, la première représentant le spectre fréquentiel et la seconde la variation de taille du sarcomère.

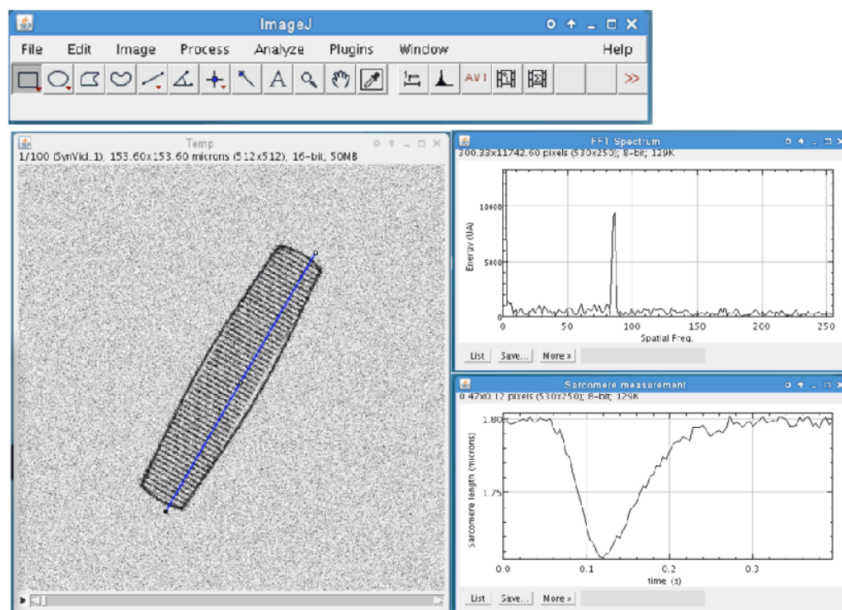


Figure 4.1 – *Plugin SarcOptim*

## 2 La technologie CUDA

La technologie CUDA est une technologie créée par NVIDIA. Elle nous permet d'effectuer des calculs complexes directement sur carte graphique (*GPU*) plutôt que sur processeur (*CPU*) en utilisant le compilateur C. La grande difficulté de cette technologie est que les calculs doivent être parallélisables. En effet, afin d'optimiser le temps de calcul, ceux-ci sont parallélisés en différents blocs (*Grid*) eux-mêmes décomposés en Thread.

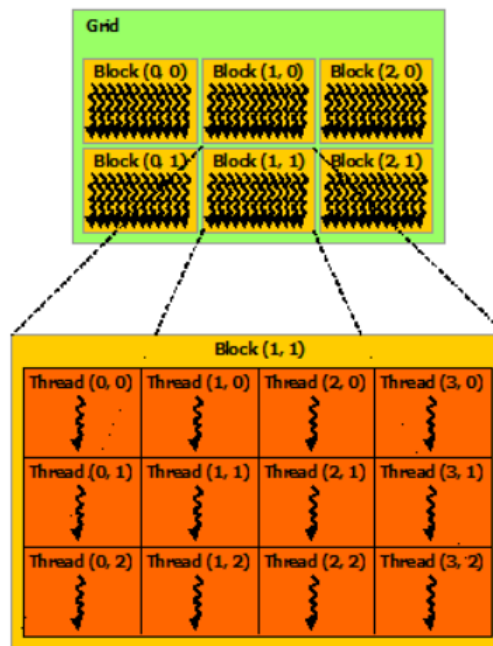


Figure 4.2 – *CUDA*

Si les calculs ne peuvent pas être parallélisés, cette technologie ne pourra pas apporter de plus value à notre solution.

## 3 Parallélisation des calculs

Pour ce type de solutions, deux technologies sont envisagées :

- JCuda
- Python

### JCuda

JCuda est une API qui permet l'interaction entre le noyau C et le langage Java. ImageJ étant un logiciel développé en Java, tous ses plugins doivent l'être également. C'est pourquoi François Gannier a commencé à développer une solution en JCuda mais celle-ci n'étant pas efficace, elle a été abandonnée. J'ai, dans un premier temps, récupéré cet algorithme afin de comprendre pourquoi cette solution n'était pas viable. Effectivement, l'algorithme a été mal implémenté ce qui avait pour conséquence une parallélisation inexistante et donc un temps d'exécution plus long.

Aujourd'hui cette technologie reste très peu utilisée et très peu connue. Il est donc assez difficile de se documenter sur JCuda.

## Python

Aujourd'hui Python est un langage très utilisé pour la recherche car il nous permet d'effectuer des calculs très simplement de manière optimisée. De plus, il nous permet également une interaction avec le noyau C via la librairie PyCuda et la librairie Numpy dispose déjà d'un calcul de FFT. De plus Python est en constante évolution et l'application restera maintenable lors des évolutions futures des machines. Python étant plus diversifié que JCuda, la documentation est par conséquent plus accessible.

## Etude comparative

Une étude comparative de ces deux technologies est nécessaire. J'ai donc développé un script Python afin de calculer une FFT parallélisée, sur une image donnée. Par la suite, je développerais un script JCuda afin d'effectuer les mêmes calculs sur une même image donnée. Pour finir, je comparerais les résultats des temps d'exécutions obtenus afin de connaître la solution la plus optimisée.

## 4 Analyse d'images : Détection de contours

Cette méthode consiste à appliquer un filtre de détection de contours sur chaque image de la vidéo. Grâce à cette technique nous pouvons détecter le contour des sarcomères. Cette solution est prometteuse car elle nous permettrait de passer outre les calculs de FFT. En effet, si nous parvenons à détecter le contour d'un sarcomère il nous suffit de calculer une norme de vecteur afin de récupérer la taille de la cellule sur chaque image de la vidéo.

J'ai donc créé un script Python, qui nous permet d'appliquer un filtre de détection de contours sur une image via la librairie OpenCV afin de voir s'il était possible de détecter correctement les contours d'un sarcomère.

# 5

## Analyse et conception

### 1 Analyse

#### 1.1 Hypothèses utilisées

Durant cette phase d'analyse, le confinement m'a limité pour effectuer les tests pour la technologie CUDA. En effet, comme j'ai pu le préciser précédemment, une carte graphique NVIDIA est nécessaire afin de pouvoir l'utiliser. Il m'a donc été impossible d'effectuer des tests en début de confinement.

Aujourd'hui, je privilégie la solution par détection de contours. Effectivement, cette solution paraît plus efficace et ne nécessite pas de matériel particulier.

#### 1.2 Spécifications

Pour résumer, afin de créer cette application, deux types de solutions ont été envisagés :

- 1 - Parallélisation via la technologie CUDA
- 2 - Détection de contours via une analyse d'images

Pour le premier type de solution, il existe deux langages possibles : Java (JCuda) et Python (PyCuda). Java nous permet une compatibilité directe avec le logiciel ImageJ car celui-ci a été développé en Java. Néanmoins, Python reste un langage très optimisé pour ce type de calcul et dispose de librairie comportant déjà des calculs de FFT. Une étude comparative de ses deux solutions est donc nécessaire.

Pour finir, une comparaison entre la technologie CUDA et l'analyse d'images, sur une vidéo en temps réelle est également nécessaire. En effet, cette solution nous permet de réduire la complexité des calculs, il est donc important de la considérer.

## 2 Modélisation proposée

Comme présenté dans la partie "Structure générale du système" [6] il y a un diagramme de classes par type de solution :

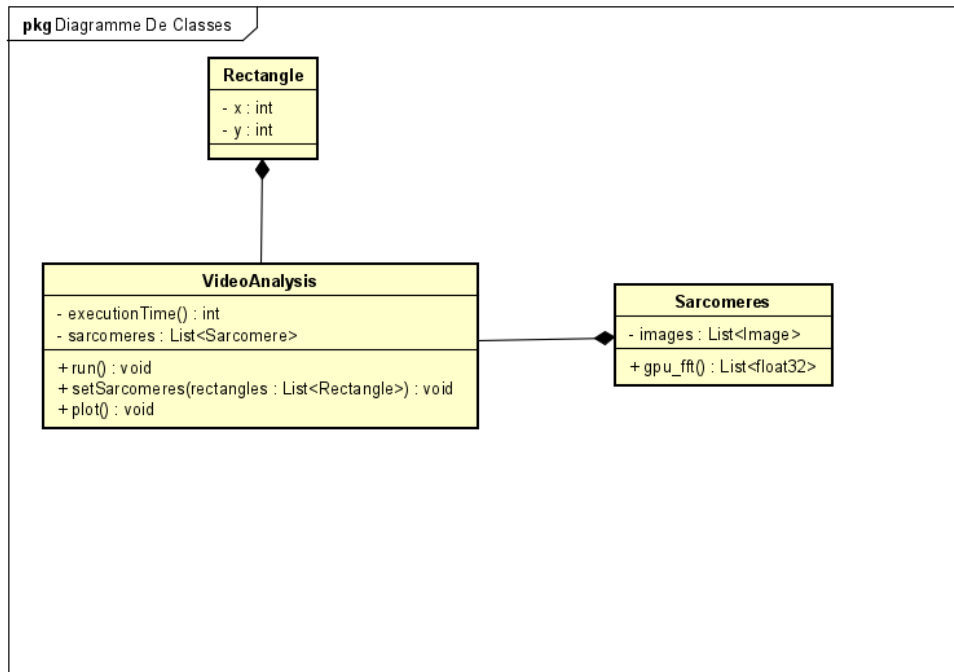


Figure 5.1 – Technologie CUDA

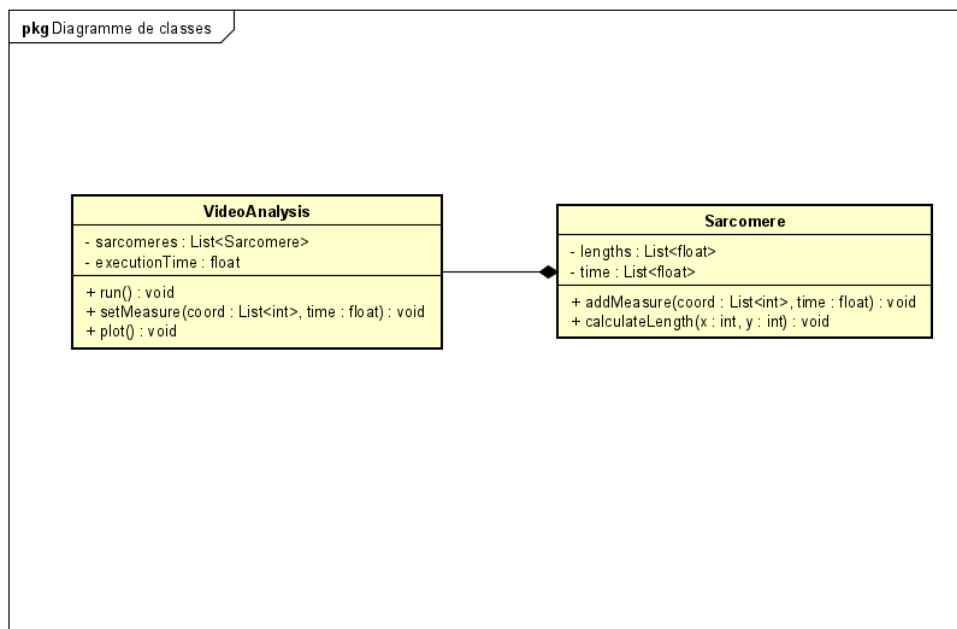


Figure 5.2 – Analyse d'images

# 6

## Bilan et conclusion

### 1 Bilan du semestre 9

Dans un premier temps, je me suis familiarisé avec la problématique de ce projet. J'ai également appris à me servir du logiciel ImageJ avec l'aide de Côme Pasqualin. J'ai pu, par la suite, étudier les scripts existants tels que SarcOptim mais aussi le script JCuda développé par François Gannier. Ce qui m'a permis d'améliorer ma compréhension globale du sujet mais aussi de me familiariser avec la technologie JCuda.

Dans un second temps, j'ai pu commencer une étude comparative des technologies Python et JCuda. Cependant, le confinement ne m'a pas permis de commencer cette étude en temps voulu. Initialement le début de celle-ci était prévu au retour de la pause pédagogique (semaine du 2 Novembre 2020). Or, cette phase a débuté le 25 Novembre 2020 ce qui a engendré un retard dans mon planning. Cette étude va donc continuer après la pause pédagogique du 18 Décembre 2020 au 3 Janvier 2021.

Parallèlement, j'ai débuté un algorithme de détection de contours. Actuellement, je travaille encore sur une détection précise de sarcomère. Afin de tester cet algorithme, j'utilise des images issues de vidéos fournies par Côme Pasqualin. Le but de cet algorithme est de savoir s'il est possible de détecter les contours d'une cellule rapidement et précisément.

Mon objectif est de terminer tous les algorithmes de tests de performances avant la fin du semestre 9 afin de pouvoir planifier une réunion avec Côme Pasqualin. Cette réunion me permettra de présenter toute mon étude de performances aux chercheurs et donc de prendre une décision sur la solution la plus adaptée à ce projet.

Aujourd'hui la solution de détection de contours me paraît la plus adaptée pour ce besoin. Effectivement, les calculs avec CUDA restent assez complexes et ne pourront probablement pas être effectués assez rapidement. De plus, le confinement a été un réel frein durant cette phase de tests et pourrait l'être également durant la phase de développement.

# Annexes

# A

# Planification, gestion de projet

## 1 Evolution du projet

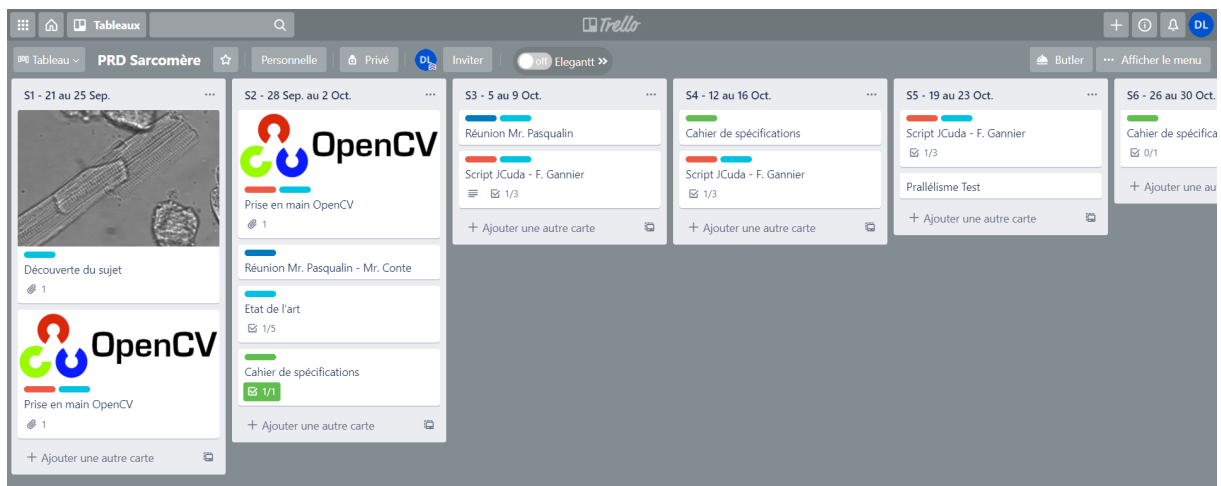


Figure A.1 – Outil Trello

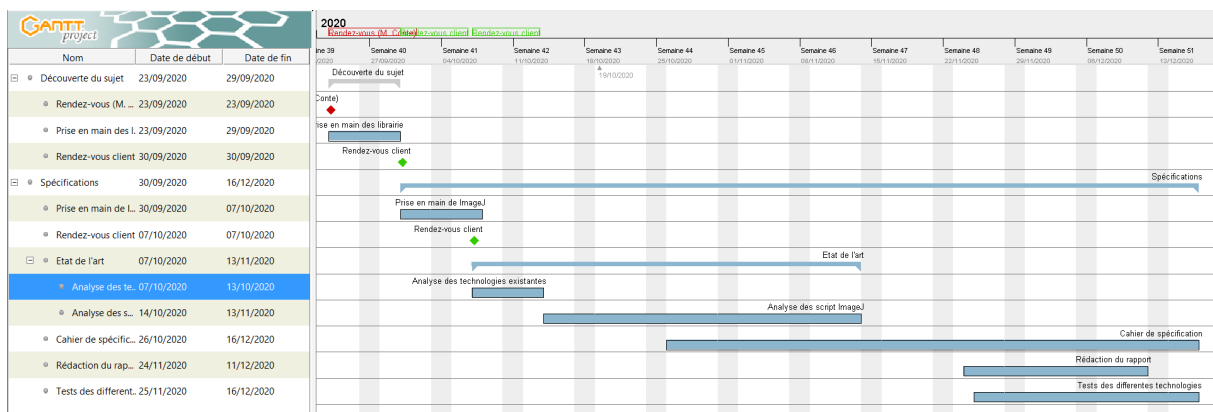


Figure A.2 – Diagramme de Gantt final

## 2 Description des tâches

### Réunion 1 : Rendez-vous M. Conte

- Date : 23/09/2020
- Lieu : Polytech Tours
- Description : Prise de connaissance du sujet et prise de rendez-vous avec Côme Pasquelin.

### Tâche 1 : Découverte du sujet

- Date de début : 23/09/2020
- Date de fin : 29/09/2020
- Duree : 5 jours
- Description : Etude des librairies Python pour la détection de contours.

### Réunion 2 : Rendez-vous Côme Pasqualin

- Date : 30/09/2020
- Lieu : Polytech Tours
- Description : Explication du besoin client, logiciel utilisé et méthode de calcul.

### Tâche 2 : Prise en main de ImageJ

- Date de début : 30/09/2020
- Date de fin : 07/10/2020
- Duree : 6 jours
- Description : Essai du protocole de calcul de variation de taille d'un sarcomère via le plugin SarcOptim.

### Réunion 3 : Rendez-vous Côme Pasqualin

- Date : 07/10/2020
- Lieu : UFR des Sciences Pharmaceutiques de Tours
- Description : Côme a pu m'expliquer plus en détail le contexte du projet, m'aider à mieux comprendre le fonctionnement d'ImageJ mais aussi me fournir le script JCuda existant.

### Tâche 3 : Analyse des technologies existantes

- Date de début : 07/10/2020
- Date de fin : 13/10/2020
- Duree : 5 jours
- Description : Documentation sur JCuda et les FFT Python.

### Tâche 4 : Analyse des script ImageJ

- Date de début : 14/10/2020
- Date de fin : 13/11/2020
- Duree : 23 jours
- Description : Analyse du plugin SarcOptim + script JCuda.

### Tâche 5 : Cahier de spécifications

- Date de début : 26/10/2020
- Date de fin : 16/12/2020
- Duree : 38 jours
- Description : Rédaction du cahier de spécifications.

**Tâche 6 : Rédaction du rapport**

- Date de début : 24/11/2020
- Date de fin : 11/12/2020
- Duree : 14 jours
- Description : Rédaction du rapport.

**Tâche 7 : Tests des différentes technologies**

- Date de début : 25/11/2020
- Date de fin : -
- Duree : -
- Description : Création des algorithmes de calculs de FFT. Cette tâche n'est actuellement pas terminée.

# B

## Description des interfaces

### 1 Interface matérielle/logicielle

Une interface matérielle/logicielle pourra être créée. En effet, je devrais développer un plugin pour ImageJ afin d'afficher tous les résultats sur ce logiciel. Cette interface est indispensable pour le premier type de solution car les chercheurs devront encadrer chaque cellule. En revanche elle n'est pas nécessaire pour la seconde solution.

### 2 Interface homme/machine

Pour le premier type de solution, l'interface homme/machine existe d'ores et déjà : ImageJ. En revanche, pour le second une interface doit être développée. Celle-ci doit afficher le flux vidéo entrant et les courbes de variation de taille des sarcomères.



# Cahier de Specifications

## 1 Spécifications fonctionnelles

### 1.1 Définition de la fonction de traitement : Mesurer la variation de taille d'un sarcomère

#### Description de la fonction :

Entrée : Flux vidéo

Sortie : Courbes de variation de taille de chaque sarcomère

Préconditions : Au moins un sarcomère présent sur la vidéo en entrée

Postconditions :  $\emptyset$

## 2 Specifications non fonctionnelles

### 2.1 Contraintes de développement et conception

Comme j'ai pu le préciser précédemment, une contrainte matérielle existe pour le premier type de solution. En effet, une carte graphique NVIDIA est nécessaire afin d'utiliser la technologie CUDA.

### 2.2 Contraintes de fonctionnement

La contrainte de fonctionnement de ce projet est que nous travaillons sur un flux vidéo en temps réel.

#### 2.2.1 Performances

L'application doit calculer les FFT ou encore détecter les contours à une vitesse assez élevée. En effet, il ne faut pas oublier que cette application travail directement sur un flux vidéo. Par conséquent, si la méthode de mesure est trop couteuse en temps, nous prenons le risque de ne pas détecter cette variation de taille.

# D

## Bibliographie

- [1] « CUDA toolkit Documentation ». In : (2015). URL : <https://docs.nvidia.com/cuda/index.html>.
- [2] Côme PASQUALIN et François GANNIER. « SarcOptiM (High Frequency Online Sarcomere Length Measurement) ». In : (2015). URL : <http://pccv.univ-tours.fr/ImageJ/SarcOptiM/>.
- [3] Côme PASQUALIN et François GANNIER. « SarcOptiM for ImageJ : high-frequency online sarcomere length computing on stimulated cardiomyocytes ». In : (2016). URL : <https://journals.physiology.org/doi/full/10.1152/ajpcell.00094.2016>.

# E

## Glossaire

**CUDA** : Technologie développée par NVIDIA afin d'effectuer des calculs directement sur **GPU** tout en étant parallélisés. [3](#), [26](#)

**GPU** : Processeur graphique. Dans ce contexte, il s'agit d'une carte graphique NVIDIA. [22](#)

**ImageJ** : Logiciel utilisé aujourd'hui par les chercheurs afin d'étudier les sarcomères. [2](#)

**sarcomères** : Unité contractile dans les cellules cardiaques. [2](#), [26](#)

# F

## Acronymes

**FFT** Transformée de Fourier. **2**

**GPU** Graphics Processing Unit. **3**

Lucas Dos Santos

Encadrement : Donatello Conte



En collaboration avec  
UFR des Sciences Pharmaceutiques de Tours

## Objectifs

- Mesurer précisément les variations de tailles de sarcomères



LABORATOIRE D'INFORMATIQUE FONDAMENTALE ET APPLIQUÉE DE TOURS

## Mise en œuvre

- Développement d'algorithmes de tests : CUDA
- Développement d'un algorithme de détection de contours



LABORATOIRE D'INFORMATIQUE FONDAMENTALE ET APPLIQUÉE DE TOURS

## Résultats attendus

- Comparaison des résultats de toutes les solutions envisagées
- Choix d'une solution



LABORATOIRE D'INFORMATIQUE FONDAMENTALE ET APPLIQUÉE DE TOURS



# Mesures multiples et rapides de sarcomèrespolytech@subtitle:

Lucas Dos Santos

Encadrement : Donatello Conte



En collaboration avec UFR des  
Sciences Pharmaceutiques de Tours

## Objectifs

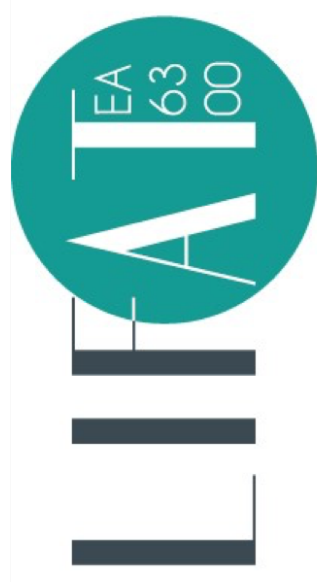
- Mesurer précisément les variations de tailles de sarcomères

## Mise en œuvre

- Développement d’algorithmes de tests : CUDA
- Développement d’un algorithme de détection de contours

## Résultats attendus

- Comparaison des résultats de toutes les solutions envisagées
- Choix d’une solution



# Mesures multiples et rapides de sarcomères

## Résumé

Ce projet a pour objectif de développer une application pour mesurer les variations de taille d'un sarcomères. Deux types de solution sont envisagés, CUDA et la détection de contours via une analyse d'images.

## Mots-clés

Sarcomère, Transformée de Fourier, CUDA, Analyse d'images, Detection de contours

## Abstract

This project's goal is to create a new application to measure a sarcomere length variations. There is two solution type : CUDA technologie and shape detection by image analysis.

## Keywords

Sarcomere, Fast Fourier Transform, Image analysis, Shape detection

## Entreprise

UFR des Sciences Pharmaceutiques de Tours



## Tuteur entreprise

Côme PASQUALIN

## Étudiant

Lucas DOS SANTOS (DI5)

## Tuteur académique

Donatello CONTE