



# Rapport de Projet technologique

---

Etudiants :  
Marc CERUTTI

31 janvier 2019

## Table des matières

<b>I</b>	<b>Présentation du sujet</b>	<b>3</b>
1	Analyse de l'existant	4
2	Besoins-non fonctionnels	5
3	Besoins fonctionnels	6
<b>II</b>	<b>Architecture</b>	<b>7</b>
4	Vue Global et conventions	8
5	Interface Qt	9
6	Bibliothèque d'analyse	10
6.1	Filtre d'images . . . . .	11
6.2	Carte de disparité . . . . .	12
6.3	Calibration Camera . . . . .	13
6.4	Carte de profondeur . . . . .	14
7	Simulation	15
<b>III</b>	<b>Tests</b>	<b>16</b>
8	Tests unitaires	17
9	Performances	18
<b>IV</b>	<b>Bilan</b>	<b>19</b>
10	Difficultés rencontrés	20
11	Critique et améliorations potentielles	21
<b>V</b>	<b>Annexes</b>	<b>22</b>
A	Hiérarchie du projet	23
B	Rapport Initial	25
C	Rapport 05 novembre 2018	26



## Première partie

# Présentation du sujet

L'objectif de ce projet de L3 informatique est à terme de créer un robot suiveur qui suivra son utilisateur à distance de 2 mètres.

On passera cependant par plusieurs étapes.

D'abord la création de nos fonctions d'analyse et une interface graphique à partir de la bibliothèque OpenCv et Qt respectivement. OpenCv implémente notamment des algorithmes pour faire les filtres, les cartes de disparité et les cartes de profondeur nécessaires à l'analyse d'images pour l'intelligence artificielle, et Qt un système d'interface complet pour pouvoir créer une interface permettant de tester nos fonctions d'analyse d'images.

Il faudra par contre se concentrer sur les fonctions d'analyse parce qu'elles doivent être optimisé pour les systèmes embarqué.

Ensuite la création sous Unity d'une simulation avec les différents paramètres permettant de créer des comportements par défaut et testant les différents algorithmes. Elle devra prévoir les spécificités du matériel et prévoir différents tests pour différents environnements et situations.

## 1 Analyse de l'existant

## 2 Besoins-non fonctionnels

### 3 Besoins fonctionnels

Deuxième partie

# Architecture



## 4 Vue Global et conventions

## 5 Interface Qt

## 6 Bibliothèque d'analyse

## 6.1 Filtre d'images

## 6.2 Carte de disparité

### 6.3 Calibration Camera

## 6.4 Carte de profondeur

## 7 Simulation



Troisième partie

# Tests

## 8 Tests unitaires

## 9 Performances

Quatrième partie

# Bilan

## 10 Difficultés rencontrés

## 11 Critique et améliorations potentielles

Cinquième partie

## Annexes

Table des figures

## A Hiérarchie du projet

l3-rvc

```
.
|-- 3DRV.pro
|-- cmdHierarchie
|-- ImageAnalyserTest.pro
|-- main.cpp
|-- maintests.cpp
|-- mainwindow.cpp
|-- mainwindow.h
|-- mainwindow.ui
|-- PRIORITES.txt
|-- Rapport
|   |-- Brouillons
|   |   |-- 14_12_2018.txt
|   |   |-- Calibration.txt
|   |   |-- Procédure Matériel
|   |   |-- rapport_05_11_2018.txt
|   |   '-- rapport_05_12_2018.txt
|   |-- hierarchie.txt
|   |-- img
|   |   '-- ubx-logo.png
|   |-- rapport.pdf
|   '-- rapport.tex
|-- README.md
|-- subWindows
|   |-- bmparamdialog.cpp
|   |-- bmparamdialog.h
|   |-- bmparamdialog.ui
|   |-- sgbbmparamdialog.cpp
|   |-- sgbbmparamdialog.h
|   '-- sgbbmparamdialog.ui
'-- tools
    |-- cameracalibration.cpp
    |-- cameracalibration.h
    |-- cvqtinterface.cpp
    |-- cvqtinterface.h
    |-- imageanalyser.cpp
    |-- imageanalyser.h
    |-- imageanalysertests.cpp
    |-- imageanalysertests.h
    |-- projectdebugger.cpp
    '-- projectdebugger.h
```



5 directories , 35 files

## B Rapport Initial

Scission du groupe initial du fait de problèmes internes concernant des choix d'implémentation de l'interface graphique.

Fonctionnalités de base déjà présentes à ce moment :

- Fonctions de conversion basique d'images de OpenCv vers Qt.
- Mise en place d'une bibliothèque graphique d'analyse adaptant les fonctions d'OpenCv.
- Ouverture de dialogues particulier pour les cartes de disparité.
- Filtres de flou Gaussien, de Laplacien, et de séparation d'image pour les cartes de disparité implémenté.
- Fonction de déboguage d'image pour l'affichage OpenCv crée.

## C Rapport 05 novembre 2018

### Rapport du groupe CERUTTI

Le 05 novembre 2018

Interface graphique :

- Rajout d'une check box pour remettre à l'image d'origine automatiquement dans la fenêtre principal.
- Rajout de l'affichage de l'approximation de l'efficacité des fonctions.
- Mise à jour du code des boutons, des menus, et des fenêtres pour implémenter ces nouvelles mécaniques.
- Rajout d'une fenêtre pour les paramètres de StereoBM.
- Rajout des boutons de Sobel et Flou gaussien.

Analyse d'images :

- Amélioration des algorithmes pour prendre en charge les images en GrayScale, permet d'enlever les conversions redondantes, et l'utilisation des fonctions de manière successive facilité.
- Rajout de showMatrice(cv : :Mat mat) dans ImageAnalyser pour le déboguage.
- Rajout de computeEfficiency(double time, func, args) pour avoir une approximation de l'efficacité des algorithmes.
- Rajout des fonctions Sobel et Flou gaussien.

Mise en place d'un exécutable pour les tests d'analyse d'images.

Pistes d'améliorations :

- Implémentation des tests
- Recherche des meilleurs paramètres pour algorithmes stereo
- Factorisation du code pour les fenêtres BM et SGBM
- Recherche sur la possible utilisation de threads
- SteroVar non essayé openCv
- Recherche analyse de flux vidéo
- Recherche object tracking et template

### Remarques professeur :

Pas d'image dans Git  
Signaler fuites mémoire, même bibliothèque  
Sobel demandé n'est pas juste l'utilisation la fonction de OpenCv, l'objectif était de faire un gradient  
Voire problème de carte de disparité  
Trop de découpage de code  
Laisser le code cvtColor dans les fonctions nécessaires sinon cela ne facilite pas la lecture de code  
Mettre des références à la place de la recopie de matrice

**Prochaine séance :**

Simplifier le code

Renommer les variables avec convention (code et fichiers) et commenter le code

Voire carte disparité problème

Prévoir quoi faire avec le matériel

## D Rapport 5 Décembre 2018

Rapport du groupe CERUTTI Le 05 Décembre 2018

Correction depuis dernier rendu :

-Adoption d'une convention de nommage :

Attribut |\_name\_complete Variables locales |name\_complete Méthodes |nameComplete Classes |NameClass Fichiers |nameclass.\*

-Changement de la fonction Sobel en Gradient, et implémentation en vrai Gradient.

-Correction des cartes de disparité inversé

-Amélioration des performances en enlevant la recopie des return et en passant des références en paramètre.

Calibration : -Implémentation de findOneCalibration. Pour une image particulier il trouve les paramètres de calibration, et les mets dans un fichier. Il envoie ensuite dans la matrice de sortie l'image non distordue.

-Test valgrind, 32 erreurs externes (supposé lié à la bibliothèque Qt).

Interface graphique :

-Rajout d'une option TestCameraCalibrate, dans les menus prend l'image courant d'interface et applique findOneCalibration.

Remarque, certaines images sont distordus comme la set1/10\_20\_43\_159.jpg après calibration. Possibilité que le ChessBoard soit trop loin, couplé au fait qu'il n'y a qu'une seule valuation de calibration.

Pistes d'améliorations :

-Filtre pour les cartes de disparité (si cela n'impacte pas les performances)

-Créer un système de sauvegarde général pour calibration et performances.

-Améliorer le système de calibration par multiples valeurs et prise en charge automatique de calibration par dossier d'images.

-Améliorer l'interface pour prendre en compte la calibration de différentes caméras.

### Remarques professeur :

Sobel erreur

Syntheses à améliorer

-enlever qtdesigner.txt

-mettre algos

-résultats

-problème

-pas de code dans rapport

Erreur récurrentes (?)

Rigueur sur le code, moins sur la théorie

Rapport pas un bilan d'activité