



Rapport de Projet technologique

Etudiants :
Marc CERUTTI

5 décembre 2018

Table des matières

Première partie

Introduction

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.

RENDU FINAL :

Paramètres de fonctions

Carte de disparité

Carte de profondeur

Rapport

Avec graphiques performances

Tests

Soutenance

Deuxième partie

Développement

1 Bibliothèque d'analyse

1.1 Fonctionnalités

1.2 Interface Qt

2 Mise en place simulation

2.1 Spécificité de caméra

2.2 Analyse d'environnement

2.3 Algorithme de de décision

2.4 Tests

3 Situation réelle

Troisième partie
Conclusion

Quatrième partie

Annexes

Table des figures

A Hiérarchie du projet

l3-rvc

```
.
|-- 3DRV.pro
|-- 3DRV.supp
|-- cmdHierarchie
|-- ImageAnalyserTest.pro
|-- main.cpp
|-- maintests.cpp
|-- mainwindow.cpp
|-- mainwindow.h
|-- mainwindow.ui
|-- Rapport
|   |-- Calibration.txt
|   |-- hierarchie.txt
|   |-- img
|   |   '-- ubx-logo.png
|   |-- Procédure Matériel
|   |-- rapport_05_11_2018.txt
|   |-- rapport.pdf
|   '-- rapport.tex
|-- README.md
|-- subWindows
|   |-- bmparamdialog.cpp
|   |-- bmparamdialog.h
|   |-- bmparamdialog.ui
|   |-- sgbmparamdialog.cpp
|   |-- sgbmparamdialog.h
|   '-- sgbmparamdialog.ui
'-- tools
    |-- cameracalibration.cpp
    |-- cameracalibration.h
    |-- imageanalyser.cpp
    |-- imageanalyser.h
    |-- imageanalysertests.cpp
    '-- imageanalysertests.h
```

4 directories , 29 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 22 novembre 2018