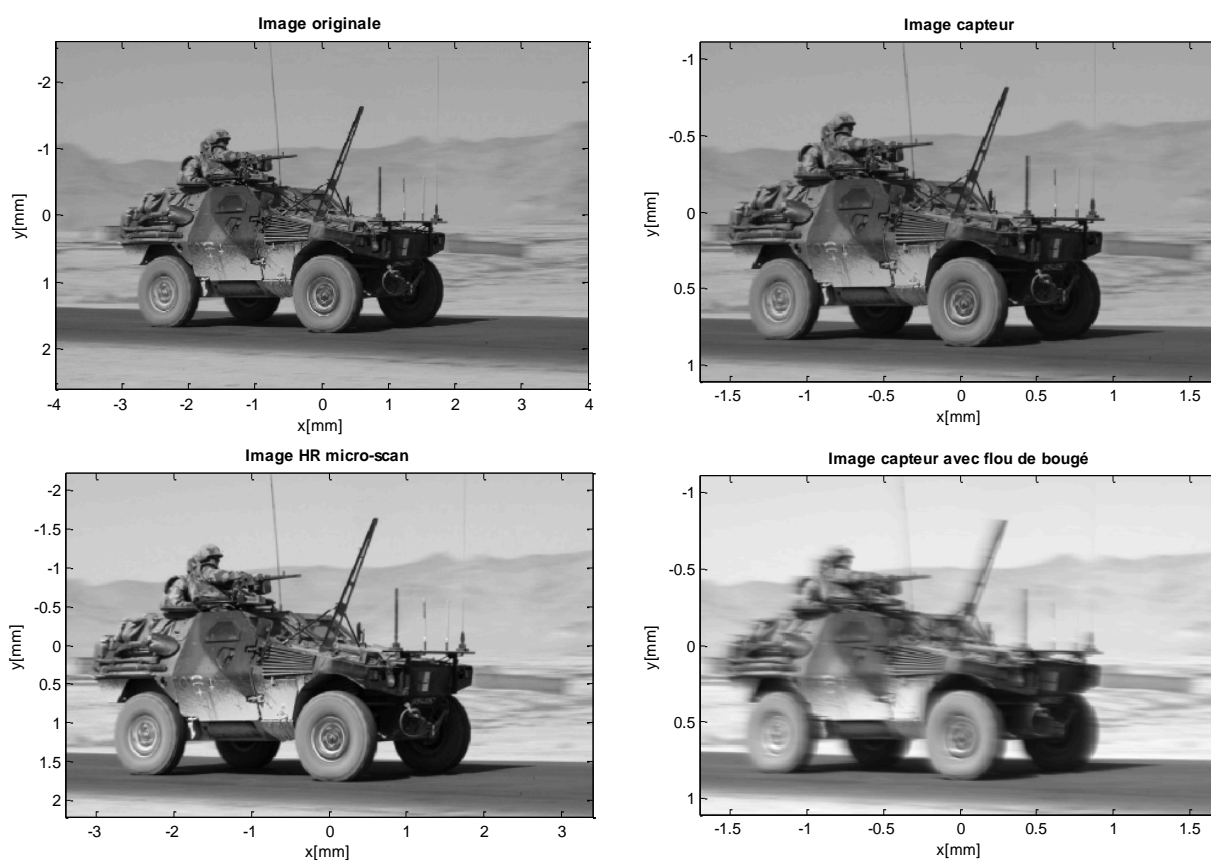


# Projet de Fin d'Étude : Développement d'un module de modélisation capteur (DETSAMPLING)



6 octobre 2014 - 19 décembre 2014

Entité :	Polytech Annecy Chambéry/Université de Savoie	DGA Maîtrise de l'Information
Adresse :	5 chemin de Bellevue 74944 ANNECY LE VIEUX	La Roche Marguerite 35170 BRUZ
Coord. Tél :	04 50 09 66 09	02 99 42 90 11

**Tuteur entreprise :** Frédéric LOAËC, Gregory SWIATHY  
**Tuteur école :** Pascal MOUILLE  
**Stagiaire :** Rémi LACROIX

+33 6 83 24 77 89  
[lacroixremi0@gmail.com](mailto:lacroixremi0@gmail.com)

# Table des matières

<b>Remerciements .....</b>	<b>i</b>
<b>Table des figures et illustrations .....</b>	<b>ii</b>
<b>Table des tableaux .....</b>	<b>iii</b>
<b>Table des matières .....</b>	<b>iv</b>
<b>Introduction .....</b>	<b>1</b>
<b>1 Description du PFE .....</b>	<b>2</b>
1.1 Définition du sujet .....	2
1.2 Organisation du PFE .....	2
1.3 Environnement logiciel .....	3
<b>2 Module DETSAMPLING .....</b>	<b>4</b>
2.1 Notions générales .....	4
2.2 Description du ré-échantillonnage.....	6
2.3 Analyse de l'existant .....	8
2.4 Développement d'un code optimisé .....	14
2.4.1 Codage.....	14
2.4.2 Analyse des performances.....	19
2.4.3 Exemples et applications du code optimisé.....	22
2.5 Recherches bibliographiques et perspectives .....	27
<b>3 Évolutions de la toolbox .....</b>	<b>30</b>
3.1 Restructuration de la toolbox .....	30
3.2 Mémentos .....	32
3.3 Revue du code .....	33
<b>Conclusion technique .....</b>	<b>34</b>
<b>Bilan personnel .....</b>	<b>34</b>
<b>Références .....</b>	<b>35</b>
<b>Résumé .....</b>	<b>36</b>
<b>Annexes .....</b>	<b>I</b>
Annexe I : Qu'est-ce qu'une toolbox ?.....	I
Annexe II : Interfaçage et fichier MEX.....	I
Annexe III : Proposition de projet.....	II
Annexe IV : Planning prévisionnel du projet .....	IV
Annexe V : Pseudo-code de la fonction <i>imDetMatrice</i> .....	V
Annexe VI : Pseudo-code de la fonction <i>imDetPixel</i> .....	VI
Annexe VII : Tests sur une image technique.....	VII
Annexe VIII : Calcul de surface d'un polygone.....	VIII
Annexe IX : Description des fonctions de gestion .....	IX
Annexe IX : Description des fonctions de gestion (suite).....	X
Annexe X : Considération sur la normalisation des valeurs.....	XI

## Introduction

---

Dans le cadre de ma dernière année d'école d'ingénieur à Polytech Annecy-Chambéry – site d'Annecy j'ai été amené à effectuer un projet de fin d'étude, PFE, de 10 semaines allant du 6 octobre au 19 décembre 2014 dans les locaux de Polytech. Ce projet s'inscrit dans un partenariat avec le centre Maîtrise de l'Information de la Direction Générale de l'Armement. En effet, j'ai réalisé mon stage de fin d'année *Développement d'un module Matlab pour la modélisation de capteurs optroniques* pour ce centre DGA. N'ayant pas pu réaliser mon PFE auparavant, il m'a été proposé de poursuivre ma collaboration à travers plusieurs activités.

Dans un premier temps, il s'agit de créer un module de modélisation d'un détecteur matriciel et de l'intégrer dans la Toolbox<sup>1</sup> Optronique MATLAB de la DGA. Dans un environnement de simulation l'image utilisée en entrée de capteur est déjà échantillonnée. C'est une image *bitmap*, un ensemble de points (pixels) contenus dans un tableau 2D. Chacun de ces points possède une ou plusieurs valeurs décrivant sa couleur. L'opération d'échantillonnage pour le capteur simulé est donc une opération de ré-échantillonnage *resampling* de l'image d'entrée. Ce processus génère en sortie une image répondant aux caractéristiques souhaitées pour cette image. L'enjeu est d'offrir un paramétrage large et maîtrisé du ré-échantillonnage d'images. Une implémentation C/MEX<sup>2</sup> des algorithmes doit permettre de réduire significativement les temps de calculs.

Dans un second temps, il est demandé de réfléchir aux évolutions possibles de la *toolbox*. Le but est de livrer une version opérationnelle de l'outil au terme du projet. Une revue critique du code doit permettre d'établir une version figée. Des modifications au niveau de la gestion de l'aide et/ou de la maintenance sont à envisager.

Après une description du sujet et de la démarche adoptée durant ce projet, nous passerons en revue les codes existants. Ensuite nous décrirons le principe et les algorithmes de ré-échantillonnage. Nous parlerons également de l'optimisation des codes à travers des validations, des tests d'applications et des comparaisons de performance. Pour ce qui est de la *toolbox*, nous discuterons des modifications réalisées et envisagées. Enfin, nous conclurons en exprimant les perspectives à donner à cet outil.

---

<sup>1</sup> Un complément d'information est disponible en annexe. **Annexe I : Qu'est-ce qu'une toolbox ?**

<sup>2</sup> Un complément d'information est disponible en annexe. **Annexe II : Interface et fichier MEX**

# 1 Description du PFE

Cette partie du rapport vise à rendre compte du déroulement du PFE et à expliciter les moyens mis en œuvre pour parvenir aux résultats souhaités par l'équipe encadrante.

## 1.1 Définition du sujet

Le projet de fin d'étude tel que proposé par la DGA et approuvé par l'école est fourni en annexe.

### Annexe III : Proposition de projet

Le PFE était sous la supervision de trois personnes qui ont constituées autant d'interlocuteurs. Mes tuteurs entreprise étaient Frédéric Loaëc et Gregory Swiathy, ingénieurs expert en traitement du signal optronique respectivement en poste au centre DGA Maîtrise de l'Information à Rennes et au centre DGA Techniques Aéronautiques à Toulouse. Nous échangeons sur les éléments du projet par courriel et par téléphone. Pascal Mouille, enseignant chercheur en automatisme était mon responsable école. A ce titre il venait à ma rencontre quotidiennement.

## 1.2 Organisation du PFE

Avant le commencement de toutes activités, une liste de tâches et leur enchaînement a été proposé et approuvé par l'ensemble des acteurs de ce projet. Le document originel est disponible en annexe

**Annexe IV : Planning prévisionnel du projet.** Les 10 semaines que dure le PFE peuvent être décomposées en 3 sections :

- Développement d'un module de ré-échantillonnage (DETECTOR SAMPLING)
- Révision de la *toolbox* d'optronique
- Rédaction du rapport et création de la présentation

La figure suivante présente le planning effectif des activités.

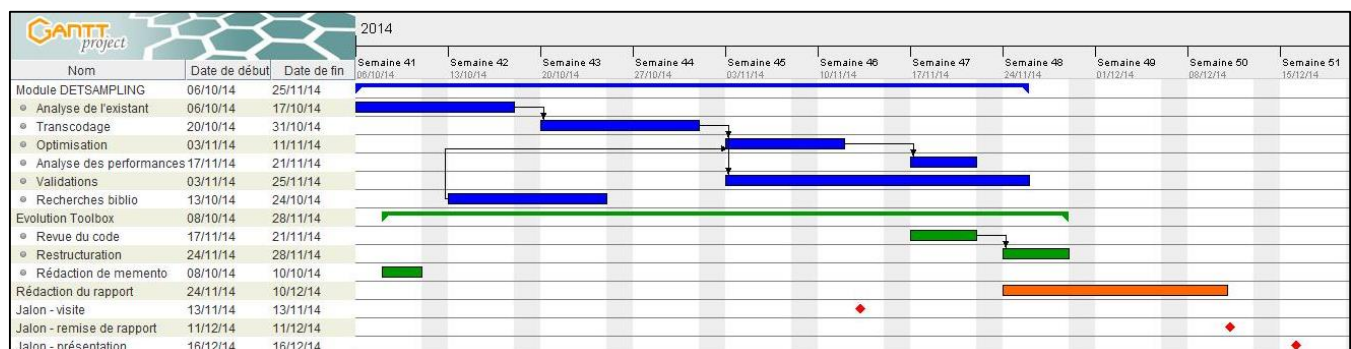


Figure 1-1 Diagramme de Gantt

Durant les premières semaines, un travail d'analyse et de recherche a été mené. En effet, la création du module de ré-échantillonnage ne s'est pas faite *ex nihilo* mais à partir de codes MATLAB existants. Ces scripts et fonctions m'ont été fournis par la DGA. Les recherches bibliographiques avaient pour but de découvrir si des méthodes et des algorithmes plus performants pouvaient répondre à nos besoins techniques.

Le transcodage consiste à implémenter un algorithme issu d'un langage informatique vers un autre langage. Ici, il s'agissait de recoder les fonctions de ré-échantillonnage MATLAB en langage C/MEX. Un langage compilé tel que le C est bien plus rapide à l'exécution qu'un langage interprété, où les instructions sont traduites au fur et à mesure.

La phase d'optimisation avait pour but d'élargir les possibilités de paramétrage du ré-échantillonnage tout en diminuant une nouvelle fois les temps de calcul. Après validation des fonctions optimisées en C les algorithmes ont été recodés en MATLAB. Une seconde phase de test et un comparatif des performances de chacune des versions du code ont été réalisés.

## 2.4.2 Analyse des performances

Afin de mesurer l'efficacité des modifications apportées au code original, les performances des fonctions de ré-échantillonnage ont été testées. La complexité des images à traiter n'influence pas les temps de calculs. Le niveau de détails ne rentre pas en jeu, seul la taille  $iN$  de l'image a un impact.

Les fonctions de ré-échantillonnage ont été testées pour différentes tailles d'images. Les dimensions des images sont issues du standard VGA. La liste des tailles d'images utilisées est disponible dans le Tableau 2-6. Les valeurs utilisées pour remplir les matrices 2D sont générées aléatoirement pour chaque format Figure 2-21. Les 4 versions du *resampling* reçoivent la même mire de test. Le temps d'exécution est l'intervalle de temps entre l'entrée et la sortie de la fonction.

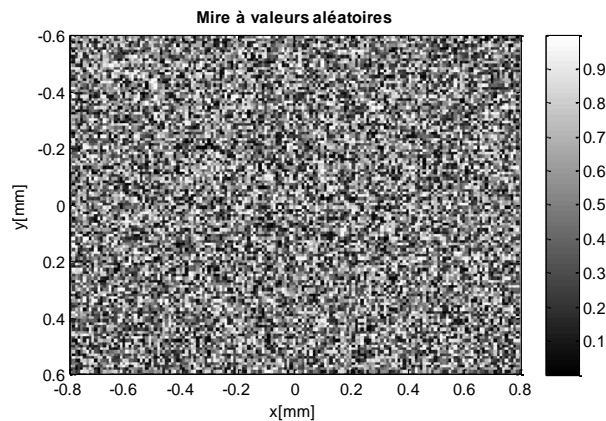


Figure 2-21 Exemple d'image aléatoire 160×120

Les résultats obtenus sont disposés dans un graphe  $iN = f(t)$  sous la forme d'une loi puissance :

$$iN = B.t^a$$

La taille des images d'entrées varie de 0.02 à près de 30 MPixel. La Figure 2-22 permet d'estimer la taille des images traitées pour un temps donné. Le graphe montre également le temps mis par les fonctions pour réaliser le ré-échantillonnage d'une image de taille précise.

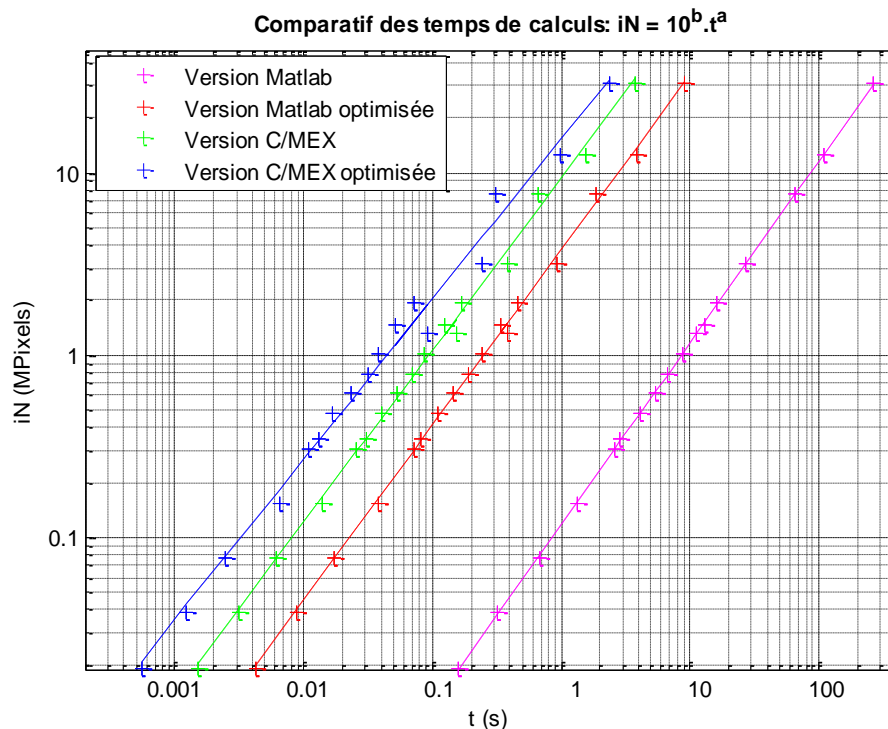


Figure 2-22 Résultats du test de performance

## Résumé

---

Un Projet de Fin d'Étude de 10 semaines a été réalisé à Polytech Annecy-Chambéry pour le compte du centre Maîtrise de l'Information de la Direction Générale de l'Armement (DGA MI).

Le département CGN2/SDO dispose et utilise des codes de calcul MATLAB pour modéliser des capteurs infrarouges. Ce projet a consisté à développer un module de modélisation d'un détecteur matriciel à partir de codes existants et d'intégrer ce module dans la Toolbox Optronique MATLAB de la DGA. Tout l'enjeu a été d'offrir un paramétrage large et maîtrisé du ré-échantillonnage d'images. Une implémentation C/MEX des algorithmes a permis de réduire significativement les temps de calculs. Une prospection a été entreprise afin d'apporter des éléments de réponse sur le ré-échantillonnage d'images ayant une rotation d'angle.

En parallèle à ce travail, des évolutions ont été apportées à la *toolbox*. Elles concernent la structure de l'outil et sa gestion. Les modules sont désormais plus autonomes. Une revue critique du code a permis de livrer une version fonctionnelle de la *toolbox* Optronique avec le module de ré-échantillonnage nouvellement créé.

**Mots clés:** ré-échantillonnage – *toolbox* – MATLAB – interfaçage C/MEX



---

## Annexes

---

### **Annexe I : Qu'est-ce qu'une toolbox ?**

Une *toolbox* MATLAB est une collection de fonctions, le plus souvent cohérentes et dépendantes les unes des autres. Celles-ci sont intégrées au sein de l'environnement de développement MATLAB comme toutes autres fonctions natives. Une documentation et des démonstrations peuvent être établies puis intégrées dans l'aide MATLAB. L'intérêt de telles structures est de faciliter le développement grâce à une utilisation simplifiée de fonctions dédiées à un domaine. Un certain nombre de boîtes à outils est disponible sous licence comme par exemple *Image Processing Toolbox* pour le traitement d'image.

[2] *Développement d'une toolbox pour la modélisation de capteurs optroniques*, partie 3, p.8

### **Annexe II : Interfaçage et fichier MEX**

MEX est l'acronyme de MATLAB EXecutable. Les fichiers MEX sont des fichiers écrits en C, C++ ou en Fortran, qui une fois compilés, peuvent être utilisés dans MATLAB de la même manière que les fichiers M. Le fait de passer par des routines C, C++ améliore la vitesse d'exécution de certains codes MATLAB notamment lors de l'utilisation massive de boucles *for* ou *while*.

Ces fichiers ont une extension différente *mexw32* ou *mexw64* selon que la version MATLAB soit 32 ou 64 bits. Les versions 32 bits de MATLAB intègre le compilateur C lcc contrairement aux versions 64 où il est nécessaire d'en installer un autre. MATLAB supporte le compilateur C/C++ issu du kit Windows SDK 7.1, de Visual C++ 2013 et de GNU gcc/g++ sous Linux. Dans un souci de compatibilité il est préférable d'utiliser des codes strictement en ANSI C *i.e.* sans notion d'orienté objet. Ainsi, peu importe l'environnement machine, un fichier MEX pourra être généré dans l'une ou l'autre architecture.

L'utilisation de fichiers MEX oblige à adopter une structure claire et à s'y tenir. La standardisation du code est bénéfique aux développeurs novices ou confirmés ; elle permet aussi la création d'outils de gestion et d'automatisation.

[2] *Développement d'une toolbox pour la modélisation de capteurs optroniques*, section 3.1.2, p.9