

Master 1 - Ingénierie Informatique

---

**Travaux d'Etude et de Recherche**  
**-**  
**GraphoScan**  
**Mémoire Intermédiaire**

---

Thibault CHARPIGNON  
Benoît GALLET  
Emmanuel HERRMANN  
Martin RÉTY

*Encadré par :* Matthieu EXBRAYAT

6 Février - 13 Mars 2017

## Contents

<b>1</b>	<b>Résumé du projet</b>	<b>3</b>
1.1	Présentation . . . . .	3
<b>2</b>	<b>Introduction au domaine</b>	<b>3</b>
<b>3</b>	<b>Analyse de l'existant</b>	<b>3</b>
3.1	Fonctionnalités déjà implémentées . . . . .	3
3.2	Fréquence d'acquisition limitée . . . . .	3
3.3	Impossibilité de bouger la feuille . . . . .	3
3.4	Problèmes divers . . . . .	4
<b>4</b>	<b>Besoins non fonctionnels et fonctionnels</b>	<b>4</b>
4.1	Augmentation de la cadence d'acquisition . . . . .	4
4.1.1	Programmation parallèle . . . . .	4
4.1.2	Complexité . . . . .	5
4.1.3	Modularité . . . . .	5
4.1.4	Généralisation . . . . .	5
4.2	Tracking de la plume . . . . .	5
4.2.1	Analyse de la complexité . . . . .	6
4.2.2	gestion mouvements . . . . .	6
<b>5</b>	<b>Prototypes et résultats de tests préparatoires</b>	<b>7</b>
<b>6</b>	<b>Planning</b>	<b>7</b>
<b>7</b>	<b>Bibliographie</b>	<b>7</b>

# 1 Résumé du projet

## 1.1 Présentation

Ce projet de TER prolonge un travail déjà entamé l'année dernière par deux étudiants de Polytech

- Présentation rapide du projet : Présenter brièvement le sujet du TER
- Les origines, la demande du paléographe : Dans quel but fait-on ça (avoir une reproduction 3D des gestes de la main pour distinguer des écoles d'écritures, ...)

## 2 Introduction au domaine

- Bibliothèques : utilisation bibliothèques graphique pour du traitement de flux vidéo dans le but de faire du tracking sur le résultat
- Langages, Frameworks : opencv matlab c++

## 3 Analyse de l'existant

### 3.1 Fonctionnalités déjà implémentées

Le programme tel qu'il nous a été fourni dispose de plusieurs fonctionnalités élémentaires à son bon fonctionnement. Parmi ceci, nous pouvons compter :

- Le calibrage du dispositif dans sa globalité (caméras + surface d'écriture)
- La synchronisation des deux caméras pour une reconstitution en trois dimensions des gestes lors de l'écriture

### 3.2 Fréquence d'acquisition limitée

Jusqu'alors, le programme ne possède pas une fréquence d'acquisition suffisante de l'image. En effet, cette dernière n'est de l'ordre que de huit images par seconde. De ce fait, cette valeur ne permet pas une reconstitution précise en trois dimensions des gestes du calligraphe. L'objectif ici est donc de se rapprocher le plus possible de la fréquence maximale d'acquisition des caméras du dispositif, soit trente images par seconde et ainsi gagner en précision lors du traitement.

### 3.3 Impossibilité de bouger la feuille

Avec la configuration actuelle, le calligraphe a l'impossibilité de bouger la feuille sur laquelle il écrit sous peine de perdre les réglages définis auparavant. Instinctivement, la personne qui écrit peut souhaiter bouger cette feuille et ainsi gagner en confort. Le but serait donc de trouver un moyen de gérer un

changement de position de la feuille sans que cela n'affecte les résultats, en recalculant les réglages par exemple.

### 3.4 Problèmes divers

Initialement, l'éclairage du dispositif se faisait à l'aide de deux spots disposés de part et d'autre du calligraphe. Le problème majeur d'un tel moyen est la présence d'ombres à certains endroits rendant le traitement des images difficile, ainsi que la chauffe des lampes pouvant se révéler gênant à la longue. Un autre problème à gérer sont les gestes "inutiles" à l'acquisition que le calligraphe peut faire. En effet ce dernier peut par exemple vouloir étendre son bras pour se relaxer, geste qui sera pris en compte par le programme dans la reconstitution.

## 4 Besoins non fonctionnels et fonctionnels

Une séparation des tâches était déjà effective dans le projet original, sur lequel travaillaient deux personnes : une personne travaillait sur l'acquisition stéréo, pendant que la seconde s'occupait de la modélisation 3D de la plume. Cette séparation a été conservée dans ce TER : GALLET Benoît et HERRMANN Emmanuel s'occupent de la première partie, tandis que CHARPIGNON Thibault et RÉTY Martin ont pris la deuxième.

### 4.1 Augmentation de la cadence d'acquisition

Naturellement, différentes idées nous sont venues pour augmenter la cadence d'acquisition de la vidéo en stéréo. Nous les détaillons ici, même si par la suite cette liste sera sûrement étoffée. Le programme fonctionne suivant plusieurs étapes : Tout d'abord, une image est capturée à partir des deux caméras, puis les deux images sont encodées dans une vidéo. Une fois la capture finie, la vidéo est traitée afin de régler les problèmes de distorsion. Grâce à cette grande boucle qui capture les images deux par deux (une image par caméra), les vidéos finales commencent et terminent exactement au même moment, et permettent donc d'avoir exactement au même moment la feuille d'écriture filmée sous deux angles différents. La modélisation en 3D sous OpenGL est alors possible. La seule variable est que plus la cadence d'acquisition est élevée, plus il y aura de fps sur les vidéos finales, et plus la modélisation 3D de la plume sera précise. De plus, notre architecture et notre code doivent être assez robuste, pour que si un jour une troisième voire une quatrième caméra soient rajoutées, le nombre de fps ne redescende pas drastiquement.

#### 4.1.1 Programmation parallèle

Grâce à la programmation parallèle, qu'elle soit au niveau du CPU avec de l'openmp ou des threads, ou au niveau du GPU avec CUDA, nous pensons pouvoir accélérer l'acquisition des images, et donc des fps sur les vidéos finales.

Nous pensons regarder quelles parties peuvent être faites en parallèle, peut-être est-il possible d'uniquement récupérer une image tous les 3 centièmes de secondes (pour les 30 fps) dans le programme principal, et de faire tous les autres traitements dans des régions parallèles, avec par exemple un thread qui s'occupe d'ajouter la prochaine image à la vidéo, un autre thread qui enlève la distorsion de l'image, etc.

#### 4.1.2 Complexité

Reprendre le code pour en examiner sa complexité est une autre piste envisagée pour augmenter les fps. Nous pensons séparer cette idée en deux étapes : Tout d'abord regarder la complexité de l'algorithme dans sa généralité, pour se rendre compte s'il y a problème ou non à ce niveau là, et voir les morceaux posant plus problème que le reste, puis faire des tests plus précisément sur ces parties pour voir précisément ce qui ne va pas. Une analyse légèrement différente pourra être effectuée, avec des tests fonctionnels calculant quelle partie prend le plus de temps. Ces tests sont très complémentaires des tests de complexité, à eux deux ils devraient mettre en exergue les problèmes principaux du code actuel.

#### 4.1.3 Modularité

Outre cette analyse de la complexité, une mise au propre du code devra être effectuée. En effet, tout se trouve dans la fonction main, dans deux grandes boucles. Une partie de notre travail sera donc de modulariser cette fonction, de la séparer en plusieurs méthodes afin de gagner en clarté. La complexité ne devrait pas être touchée car nous rajoutons l'option `-O3` lors de la compilation afin d'avoir un code en une ligne. Ce nettoyage permettra de naviguer plus facilement dans le code, et de faciliter sa maintenance par la suite.

#### 4.1.4 Généralisation

Sinon, le dernier axe sur lequel travailler sera la généralisation du nombre de caméra. Pour l'instant, tout dans le code est fait pour deux caméras, avec du code dupliqué deux fois pour chaque action. La généralisation pour  $n$  caméras sera facilitée par la modularisation du code, et permettra par la suite de rajouter une ou plusieurs caméras sans modification majeure du code, uniquement en changeant quelques *#define*.

### 4.2 Tracking de la plume

Nous avons récupéré la partie "Tracking" du projet Graphoscan. Celle-ci permet de traiter les vidéos produites par les caméras et d'en ressortir une trace des mouvements effectués par le calligraphe.

L'étudiant de Polytech qui a travaillé sur cette partie a recherché différents algorithmes permettant d'effectuer ce tracking. Son étude se focalise sur les algorithmes suivants:

- Tracking Learning Detection (TLD)
- Kernelized Correlation Filters (KCF)

Ces deux algorithmes sont basés sur l'apprentissage de toutes les apparences observées de l'objet et d'une estimation des erreurs pour ensuite les éviter.

#### 4.2.1 Analyse de la complexité

L'analyse des deux algorithmes faites par l'étudiant a montré que dans notre cas l'algorithme KCF est le plus efficace. Son étude est basée sur plusieurs critères, la déviation moyenne des deux vidéos, le nombre de frame et le temps de calcul. Seul le premier critère est réellement différent entre les deux méthodes. C'est cette différence qui a orienté son choix vers l'algorithme KCF.

Notre premier axe de recherche sera orienté vers une étude complémentaire des ces algorithmes pour vérifier la véracité de l'analyse précédente. Pour cela nous allons réutiliser les critères d'études et ensuite essayer d'en trouver d'autres pour confirmer le choix.

Dans un second temps il nous faudra rechercher d'autres algorithmes ou méthodes de programmation pour améliorer le tracking.

#### 4.2.2 gestion mouvements

Outre les difficultés de choix des algorithmes, nous faisons face également à des contraintes physique liées aux mouvements du Calligraphe. Par exemple il doit prendre des temps de repos afin de garder sa fluidité d'écriture en faisant des gestes de relaxation du poignet. Ces mouvements ne doivent pas être pris en compte par l'algorithme de tracking afin d'éviter des erreurs sur la représentation du mouvement.

Pour palier ce problème, nous avons pensé à sauvegarder à un temps  $T$  et à un temps  $T+1$  une image de la partie suivie. Puis analyser la différence entre les deux images et en ressortir un résultat positif ou négatif. Cela revient à prendre la dernière image où le calligraphe écrit et une autre image qui permettra de voir si le mouvement est la continuité de l'écriture ou un mouvement parasite.

-Général

- Idées de rajouter une caméra pour ajouter un flux vidéo de suivi de la plume
- Améliorations diverses pour la lumière (comme un panneau LED)

- Compatibilité Windows - Linux - Mac
- Zone de capture

## **5 Prototypes et résultats de tests préparatoires**

A définir plus tard

## **6 Planning**

Deux sous-groupes Martin et Thibault sur le tracking et ce qui tourne autour, Benoit et Emmanuel les caméras et l'amélioration de l'acquisition stéréo Ajouter diagramme de gantt actuel + prévisionnel (?)

## **7 Bibliographie**

A remplir au fur et à mesure de l'exploration du projet (librairies fournies et sites de recherche spécialisés) + Noter les sites visités dans un fichier partagé