École Polytechnique de l'Université de Tours
64, Avenue Jean Portalis
37200 TOURS, FRANCE
Tél. +33 (0)2 47 36 14 14
www.polytech.univ-tours.fr

Informatique Industrielle

RAPPORT DE PROJET DE FIN D'ETUDE			
Projet :	Réalité Augmentée sur iPhone 4 : détection de logos dans un environnement urbain		
Emetteur :	Rémi DEVINANT	Coordonnées : remi.devinant@etu.univ-tours.fr	
Date d'émission :	16/12/2010		
Validation			
Nom	Date	Valide (O/N)	Commentaires
Mathieu DELALANDRE			
Historique des modifications			
Version	Date	Description de la modification	
00	12/2010	Version initiale	

TABLE DES MATIERES

INTRODUCTION

La dernière année de cycle ingénieur par apprentissage en Informatique Industrielle comprend la réalisation d'un projet de fin d'étude académique. Ce dernier consiste en la réalisation d'une réponse à une demande émanant d'un « client » ici représenté par Mathieu Delalandre enseignant chercheur à Polytech' Tours département informatique.

Ce projet consiste en la réalisation de la partie logicielle embarquée d'une application de réalité augmentée. Le but est de proposer les moyens corrects pour offrir à un utilisateur une interface lui permettant de capturer un flux vidéo sur lequel s'afficheront des informations relatives aux logos présents dans l'environnement urbain.

Dans le cadre de cette réalisation, il est nécessaire d'entreprendre la réalisation de l'architecture logicielle permettant de capturer du flux vidéo à une cadence définie par la vitesse de déplacement du Smartphone, dans notre cas un Apple iPhone 4, en fonction des relevés de coordonnées de l'accéléromètre et du gyroscope.

Dans ce rapport, je commencerais par une introduction sur le challenge de la réalité augmentée, je poursuivrais par la présentation de l'iPhone et de son OS. Je présenterais ensuite les réalisations de ce projet. Une partie sera allouée à la gestion de projet.

Ce projet a aussi rencontré des contraintes et des difficultés que j'expliquerais, et dont je donnerais les conséquences sur l'avancement.

OBJECTIFS DU PROJET

L'objectif sera de réaliser l'application qui devra rentrer dans un contexte de système embarqué. C'est un système dynamique à contraintes temps réelles. La réalisation consiste à développer une partie de l'application.
L'application devra se décomposer en plusieurs parties :

- Partie gestion du gyroscope et accéléromètre donc de la position 3D de l'iPhone
- Partie gestion de la vitesse de capture de la caméra grâce aux informations gyroscope et accéléromètre
- Partie gestion de la récupération et enregistrement d'images provenant de la pile d'images dans la mémoire du Smartphone sachant que ce téléphone dispose d'une mémoire de type DMA pour le périphérique de capture d'images

Le matériel nécessaire consiste en :

- Un Iphone de génération 4
- Un ordinateur de type Mac de Apple sous une plateforme Intel (sur lequel XCode, Interface Builder et le iPhone SDK sont installés dans leur dernière version)

LA REALITE AUGMENTEE

La réalité augmentée consiste à se servir des périphériques d'acquisition d'images de n'importe quel système pour y afficher les informations que l'on veut par-dessus le flux d'images intercepté.

Au début de ce concept, le principe consistait à afficher des formes par-dessus le flux d'images. L'intérêt était de proposer par exemple des jeux interactifs avec l'environnement de l'utilisateur. La perception du monde réel par l'utilisateur est ainsi altérée et donc dite augmentée.

Actuellement, le challenge est de proposer des outils qui permettent d'offrir des informations utiles à l'utilisateur dans son déplacement dans un environnement. On superpose le monde réel avec des informations qui permettent d'augmenter les détails de la perception (par exemple les lignes de métros accessibles par rapport au positionnement et au visu de l'utilisateur).

Les applications actuelles utilisent principalement juste le flux vidéo conjugué avec les coordonnées GPS qui permettent de situer le téléphone et donc de repérer ce qui entoure l'utilisateur dans son environnement.

Figure 1 : Exemple d'application de réalité augmentée

L'APPLE IPHONE 4

L'iPhone 4 est un smartphone particulièrement complet. Il se trouve être bien adapté pour ce genre d'exercice au niveau matériel. La seule limitation étant l'iOS, l'OS embarqué de ce smartphone.
En effet il est très fermé et ne propose que des librairies qui n'ouvrent pas le fonctionnement interne du système d'exploitation. C'est une des limitations rencontrés lors de la réalisation de ce projet. D'ailleurs la réalisation pratique du projet aura été impossible à cause de ces limitations qui nécessitaient soit trop de recherches, soit simplement pas de possibilités de répondre aux contraintes demandées.
Par ailleurs, avec un téléphone plus ouvert ou un système « jailbreaké » (cracké), la réalisation aurait été possible.
Au niveau matériel, on retrouve ce qui nous serait utile c'est-à-dire un accéléromètre, un gyroscope et une caméra de bonne facture.

Figure 2 : l'Apple iPhone 4

Les caractéristiques précises du mobile sont les suivantes :

- Processeurs Apple A4 APL0398 basé sur l'ARM Cortex A8 à 1Ghz
- Puce graphique intégré au processeur A4 -> PowerVR SGX 535
- Mémoire RAM embarqué -> 512Mo DRAM Samsung
- Accéléromètre ST Micro LIS331DLH 3-axis
- Gyroscope L3G4200D Digital 3-axis

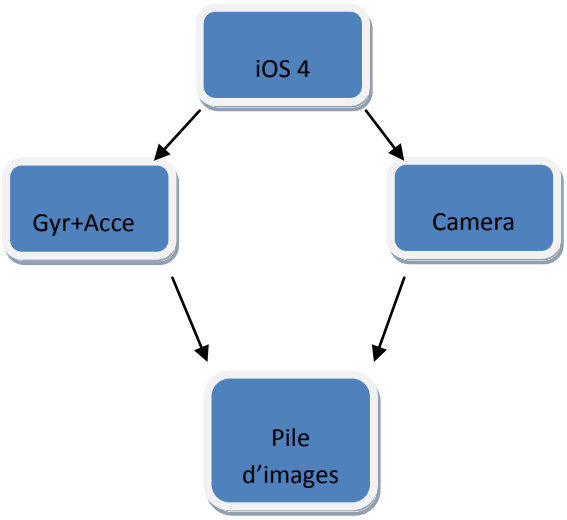
Comme le souligne les caractéristiques précédentes, l'iPhone 4 dispose de 2 outils particulièrement précis pour une localisation spatiale du téléphone. L'accéléromètre est couplé à un gyroscope.
Dans la suite de ce rapport je vais expliquer pourquoi ce système composé de 2 éléments améliore la qualité de localisation, d'accélération du téléphone. Cette précision va permettre de rendre compte très précisément des mouvements de l'utilisateur. En effet, dans ce projet le déplacement est un facteur particulièrement important, il est la base du recoupement des images comme je l'expliquerais dans la suite de ce rapport.

ARCHITECTURE GENERALE DU SYSTEME

Les composants principaux de ce projet sont les suivants :

- La caméra de l'iphone 4
- Le gyroscope
- L'accéléromètre
- La pile d'images enregistrées depuis la caméra

L'iOS4 doit faire appel au gyroscope et à l'accéléromètre pour récupérer les données de position 3D et vitesse de déplacement de l'iPhone. Il fait aussi appel à la caméra pour récupérer les images voulues en fonction de ces données. Ces images + données gyroscope + données accéléromètre doivent être liés pour ne récupérer que les images les plus pertinentes (éviter la répétition d'images trop proche) Il faudra donc accéder à la pile d'images du flux vidéo et faire une sélection intelligente.



Le schéma ci-dessus représente une interaction entre les composants principaux du smartphone. L'iOS 4 par le biais des APIs et drivers utiles fait appel au gyroscope, accéléromètre et à la caméra. Cet appel se fait dans le même temps ce qui peut être un problème au vu du fait que ces périphériques utilisent le même bus de périphériques. C'est un des points importants de toute cette étude, il faut chercher à synchroniser la sélection d'images avec les coordonnées gyroscopiques et accéléromètre associées.

LA PROBLEMATIQUE DU RECOUPEMENT D'IMAGES

Le but de ce projet est de permettre à l'application de faire une sélection d'images dans un flux vidéo pour que le traitement d'images fait à posteriori soit moins lourd et inutilement redondant.
En réalité, ce problème s'apparente à éliminer le recoupement des images dans le flux vidéo. Lorsque l'utilisateur déplace le Smartphone et que la caméra est active le nombre d'images enregistrées reste le même.

Pourtant il existe une réelle problématique :

- Si les déplacements du Smartphone sont rapides
    - Nécessité d'un grand nombre d'images car chacune est espacé spatialement
- Si les déplacements du Smartphone sont lents
    - Nécessite de réduire le nombre d'images car le déplacement à 30 images par seconde d'enregistrement on aura beaucoup d'images similaires

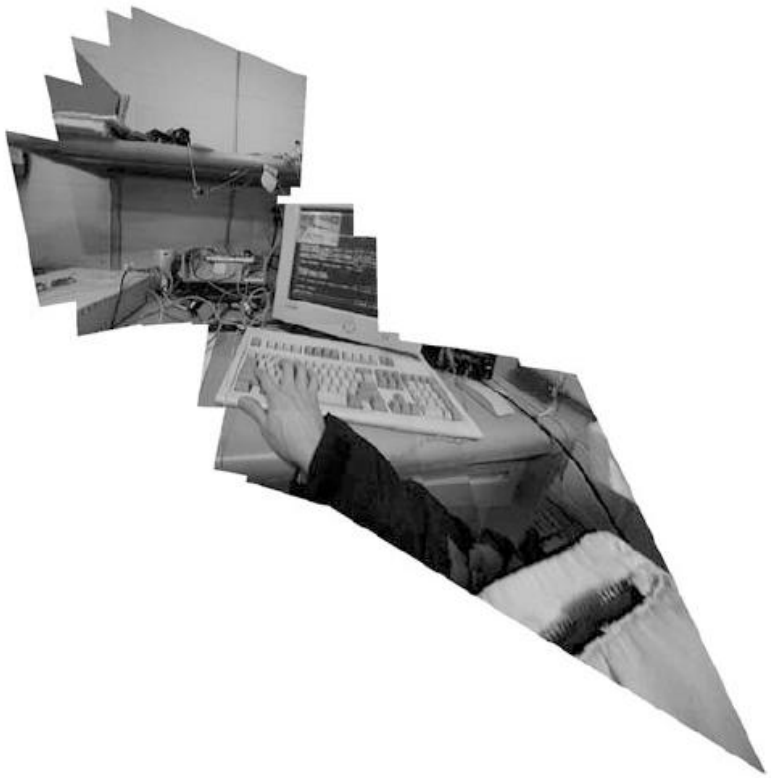


Figure 3 : Recoupement d'images dans un plan

Comme on peut le voir sur l'image ci-dessus, les plans d'images sont les uns par-dessus les autres.

Dans le cas d'un déplacement rapide :

Plan image	Plan image	Plan image	Plan image

Ci-dessus les images se recoupent sur de petites parties. La majeure partie du plan image est donc utile car on le retrouve par sur un autre plan.

Dans le cas d'un déplacement lent :

Pl		P	Plan image	Plan	Plan image

Ci-dessus, on peut observer que certaines images se recoupent sur de très grandes parties. La majeure partie de l'image se trouve donc redondante entre 2 plans images. C'est la raison pour laquelle il est nécessaire d'envisager une sélection des images enregistrées dans notre application.

PRINCIPE DE RECUPERATION DES IMAGES EN FONCTION DES DONNEES SPATIALES DU TELEPHONE

LA RECUPERATION DES IMAGES

La première phase est la récupération des images par le biais de l'enregistrement vidéo. Le traitement de compression du taux d'images gardées pour le traitement de reconnaissance des formes sera fait à posteriori.
Il est nécessaire de savoir quand une image est enregistrée et placée dans la mémoire contenant le flux vidéo. Pour cela le plus utile serait que l'OS puisse renvoyer une information stipulant « J'ai une nouvelle image en mémoire». La récupération dans la mémoire RAM voire DMA serait le plus propre.
Dans le cas de l'iOS 4, cette fonction n'est pas accessible, sauf dans le cas où elle est placée dans la mémoire morte. Il est donc nécessaire de mettre en place une forme de « timer ». Celui-ci serait une représentation temporelle de chaque sauvegarde d'image. A chaque « coup » du timer, on sauvegarde les coordonnées gyroscopique et de l'accéléromètre obtenus dans l'application. Une fois les coordonnées obtenues, on récupère l'image obtenue au « coup » du timer. Celle-ci se trouve probablement dans une mémoire « buffer » de type DMA pour Direct Access Memory. Ce principe est simpliste et représente une base dans le développement de l'application.
On peut alors récupérer le pointeur sur la case mémoire contenant l'image doublé des coordonnées obtenues.
Le principe :

- La caméra de l'iPhone 4 enregistre 30 images par secondes :

| 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 14 | … |

Ci-dessus la représentation de la pile d'images horizontalement. On récupère dans l'application chaque pointeur sur chaque case mémoire. On se retrouve donc avec une sauvegarde représentative de l'image sans pour autant chercher à occuper une taille mémoire lourde et complexe à utiliser.
On cherche alors à « tagger » cette image. C'est à dire lui attribuer des propriétés qui la concernent (position 3D).

LA RECUPERATION DES DONNEES GYROSCOPIQUE ET DE L'ACCELEROMETRE

PRINCIPE DE L'ACCELEROMETRE

L'iPhone 4, comme la plupart des Smartphones actuels, est équipés d'un accéléromètre permettant de récupérer les accélérations du téléphone sous l'effet des mouvements de l'utilisateur. L'accéléromètre de l'iPhone 4 est un Accéléromètre ST MicroElectronics LIS331DLH 3-axis.

Le principe de cet accéléromètre est de reporter l'accélération sur 3 axes. On peut ainsi enregistrer des mouvements dans l'espace.

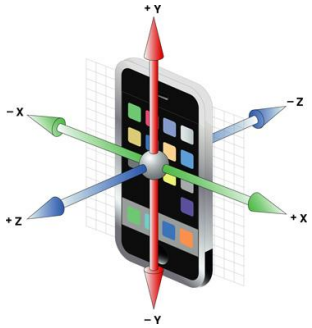


Figure 4 : Axe accéléromètre sur iPhone

Cet accéléromètre est disponible depuis l'iPhone 2G et permet une récupération assez fine des mouvements de l'utilisateur.

PRINCIPE DU GYROSCOPE

Depuis l'iPhone 4, Apple a ajouté un gyroscope qui lui permet de connaître les mouvements de rotations. Ce qui rend la prise en compte du déplacement du mobile beaucoup plus précise. En effet, il rend compte des rotations autour des mêmes axes que ceux de l'accéléromètre.

Figure 5 : gyroscope iPhone

On se retrouve donc avec des coordonnées disponibles sur les 3 axes en termes de rotation, et d'accélération, de déplacement. C'est ce qui va nous permettre de connaître la position et les déplacements du mobile autour d'une personne par exemple qui pour l'instant sera pensée fixe. Par la suite on pourra envisager de penser que la personne est en mouvement, ce qui est plus proche de notre objectif en termes de réalité augmentée.

PRINCIPE DU « TAGGAGE » DES IMAGES ENREGISTREES PAR LA CAMERA

On se retrouve avec toutes les données nécessaires à un traitement à posteriori pour les reconnaissances de formes.

Le principe est le suivant. On a un pointeur sur une zone mémoire contenant l'image. On peut sauvegarder ce pointeur en l'accompagnant des données « acceleration.x » ou y ou z (tel que le permet l'iOS) doublées des données de rotation sur x, y ou z. Les bibliothèques permettant de récupérer ces données gyroscope et accéléromètre sont connues et simple à mettre en œuvre.

D'un point de vue codage :
L'objective-C est un langage objet qui permet la création de nouveaux objets ou structure. C'est dans ce cadre là que je peux me permettre de créer en en temps voulu des nouveaux objets images + données, comme par exemple :

```
//Structure algorithmique de l'objet image + coordonnées
Class framesAnd3DDatas {
        double accelX ;
        double accelY ;
        double accelZ ;
        double roratX ;
        double rotatY ;
        double rotatZ ;
        int * pointerOnFrameMemory ;
}
```

Cette classe pourra être manipulée à posteriori pour envisager l'élimination des images se recouvrant. Pour cela, il est intéressant de se concentrer sur la vision par ordinateur pour mieux comprendre le recouvrement des images et la gestion des coordonnées obtenues depuis le repère propre à l'iPhone et le repère du plan image.
Ainsi en faisant une projection du des coordonnées du Smartphone vers le plan image, on saura positionner l'image dans l'espace. Pour effectuer tout ce traitement, il est nécessaire de se pencher sur des ouvrages de vision informatique. Je m'arrêterais plus précisément sur la vision par ordinateur plus loin dans ce rapport.

LES APIS DE L'IOS ET LA RECUPERATION D'UNE IMAGE DANS LA MEMOIRE

D'une manière officielle, il semble impossible d'aller récupérer (grâce au SDK) une image d'une vidéo directement dans la mémoire (sauf depuis l'iOS 4.0 mais seulement dans la ROM). Il est possible d'enregistrer un flux vidéo et de le récupérer une fois ce dernier entièrement sauvegardé et placé dans la mémoire accessible par l'utilisateur (la ROM du Smartphone).

Il existe de manière officieuse, c'est-à-dire par installation externe des librairies permettant d'effectuer cette manipulation. Un « repository » propose des librairies utiles à cette action. Ces librairies sont libavformat et libavcodec. De manière plus générale si la volonté est de rester encadré par un iOS propre et non jailbreaké ce type de technique ne convient pas mais je la présente tout de même dans ce rapport pour simplement montrer qu'il est possible avec un système ouvert de réaliser les fonctions que l'on souhaite.

La manière d'exploiter ces librairies est expliquée dans le lien suivant :
http://www.codza.com/extracting-frames-from-movies-on-iphone

Ce problème de limitations propre aux produits Apple a été un frein à la réalisation de ce projet. En effet, le SDK de l'iOS bloque automatiquement les librairies qui ne sont pas propres à son SDK et qui pourraient permettre de faire des traitements sur l'OS qui ne seraient pas autorisés. Au vu de toutes ces limitations le projet a pris une tournure plus théorique. Nous pouvons dire que le Smartphone est entièrement capable de réaliser les fonctions attendues mais pas avec un OS aussi fermé. Nous avons donc pris le parti avec Mathieu

Delalandre de transformer ce projet vers un algorithme de synchronisation de périphériques pour la réalisation de plan 3D.

L'ALGORITHME DE RECUPERATION DES IMAGES

A ce stade du rapport, on est donc théoriquement capable de récupérer des images du flux vidéo dans la mémoire et de les tagger pour pouvoir les identifier. Il subsiste un problème particulièrement important dans ce projet.

Nous pouvons récupérer des images avec des coordonnées, mais dans un premier temps la récupération des images n'est connue par rien ni personne et dans un second temps les coordonnées gyroscopiques sont récupérées beaucoup plus rapidement que les images du flux vidéo. De plus il existe des temps d'occupation de bus de périphériques qui vont créés de nouveaux décalages.

Il est donc nécessaire de synchroniser une récupération d'images avec une récupération de coordonnées. On va donc chercher à modéliser le décalage entre les récupérations pour proposer une solution de synchronisation.

Ce principe est en fait la réalisation d'un algorithme de synchronisation de périphériques au sein d'un OS sur un système mobile. C'est ce qui rentre parfaitement dans le cadre d'un système temps réel. On veut avoir une utilisation parallèle de 3 périphériques dans un laps que l'on veut maîtriser et pouvoir dire « je pourrais faire cet action dans un laps de temps de telle valeur avec une fiabilité de réussite à 100% ».

ALGORITHME DE SYNCHRONISATION DE PERIPHERIQUES SUR UN OS MOBILE

PROBLEMATIQUE GENERALE

Dans ce projet, la problématique s'apparente à une synchronisation de certains périphériques comme dans notre cas un gyroscope, un accéléromètre et une caméra sur un iPhone 4.

Dans le cadre d'un projet de réalité augmentée, notre but est de proposer un moyen d'utiliser la caméra pour sauvegarder du flux vidéo. Afin de soulager le processeur de traitements d'images lourd de reconnaissance de forme, nous proposons de sélectionner les images qui nous intéressent. Comme vu précédemment certaines se recoupent et notre intérêt est de ne pas toutes les sélectionner.

Pour effectuer cette opération, il est nécessaire de permettre une sélection d'images sur des critères de localisation et de déplacement dans l'espace. Ce qui signifie que chaque image doit être liée à des coordonnées gyroscopiques et d'accélération.

Le réel problème de ce projet est que nous entrons dans une problématique de temps réel. Nous devons avoir un système qui répond dans un temps minimum. Les différents périphériques partagent le même bus de périphériques et chacun possède un temps de latence particulier dans la récupération de ses données par le système. Nous devons donc resynchroniser chaque périphérique avec les autres pour être sur à 100% que l'image sauvegardée correspond aux coordonnées sauvegardées et dans un temps de réalisation considéré comme acceptable.

ARCHITECTURE MATERIELLE GENERALE

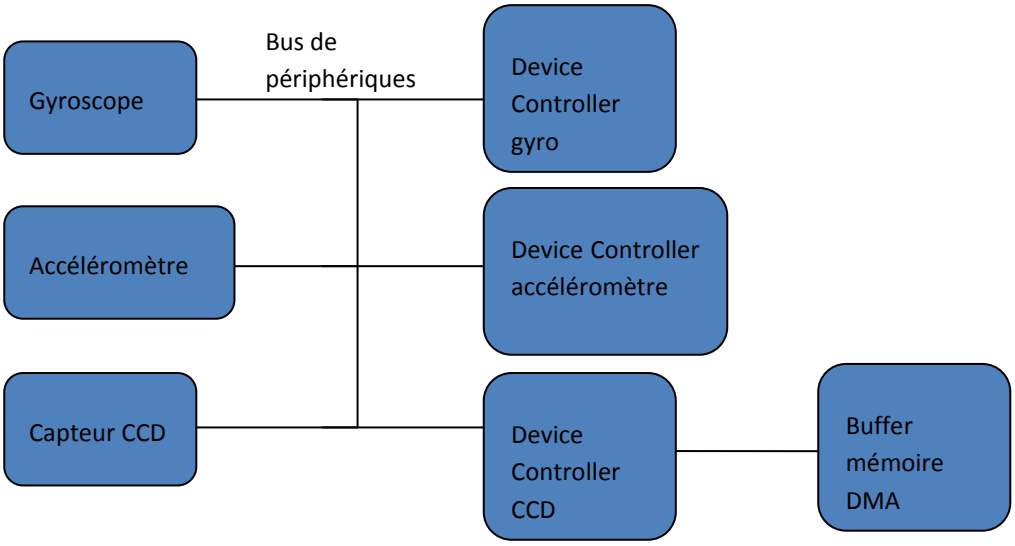
Nous considérons notre système comme un système généraliste. Nos recherches aboutissent à penser que les Smartphones actuels sont basés sur une architecture très proche voir identique à ce que je vais proposer. Je vais présenter dans cette partie les bases théoriques qui permettront d'envisager notre algorithme.
Les différents composants sont donc :

- Périphérique :
  - Le gyroscope
  - L'accéléromètre
  - Capteur CCD
- Les contrôleurs matériels :
  - Contrôleur de périphérique gyroscope
  - Contrôleur de périphérique accéléromètre
  - Contrôleur de périphérique capteur CCD
- Les mémoires :
  - La mémoire ROM contenant l'OS et les données sauvegardées
  - La mémoire RAM et les données qui transitent
  - La mémoire DMA (Direct Memory Access)
- Le processeur

MODELISATION DU PROBLEME PAR RAPPORT A L'ARCHITECTURE

Comme on peut le voir sur le schéma ci-dessous, les périphériques partagent le même bus de périphériques. Cette contrainte permet de penser qu'il existera déjà un temps delta entre chaque envoi de données par périphériques. En effet, ils ne peuvent pas tous simultanément accéder au bus de périphériques.

Un deuxième problème vient se rajouter à l'accès au bus de périphériques, c'est le temps de latence entre le moment ou un périphérique envoi une donnée et le moment ou elle est réellement en mémoire. Par exemple le temps de mise en mémoire des données de gyroscope ou d'accéléromètre est beaucoup plus rapide que celle de la mise en mémoire d'une photo prise dans une vidéo. On voit enfin apparaître la problématique de synchronisation de périphériques. J'expliquais plus haut dans ce rapport que l'on voulait faire correspondre une image de la vidéo avec des coordonnées spatiales. Il est donc nécessaire de synchroniser tous ces périphériques.

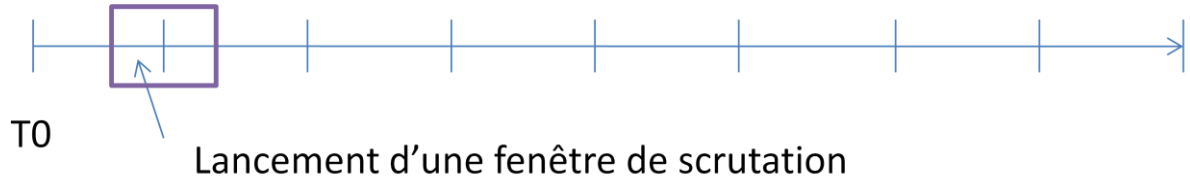


PRINCIPE DE SYNCHRONISATION

Il est nécessaire de proposer une solution pour synchroniser la récupération d'image, de coordonnées gyroscopiques et d'accéléromètre. L'algorithme suivant va permettre de répondre à cette problématique.

RECUPERATION DES IMAGES PAR LA CAMERA

La caméra enregistre des images constituant le flux vidéo. Le débit est de 30 images par seconde. On va créer une fenêtre de scrutation qui va, à chaque moment où on le lui demande, scruter en mémoire la présence d'une nouvelle image. T0 correspond au lancement de la prise de vidéo sur le système. La fenêtre violette correspond à la fenêtre de scrutation que l'on cherchera à placer autour de chaque enregistrement en mémoire. Le but est que cette fenêtre soit placer au plus proche de chaque enregistrement mémoire pour soulager l'usage du processeur. Car, en effet, cette dernière se stoppera dès la détection d'une présence de nouvelle image.



Le schéma ci-dessus représente là ou se place notre fenêtre de scrutation de présence en mémoire d'une image du flux vidéo. Chaque cran du chronogramme représente une mise en mémoire d'une image du flux vidéo par la caméra.

DEFINITION D'UNE FENETRE DE SCRUTATION

La première étape de notre algorithme sera de définir une fenêtre de scrutation en mémoire de la présence ou non d'une image. Le principe est de pouvoir détecter la nouvelle mise en mémoire d'une nouvelle image de notre flux vidéo. Cette fenêtre se recalera continuellement autour des photos obtenues.

Présentation de la fenêtre :

Dans ce chronogramme, T1 représente le début de la passe de scrutation. Δt représente l'intervalle de temps du timer de vérification de la mémoire. A chaque coup de timer, on va vérifier dans la mémoire du Smartphone si une image est présente.
La précision du timer dépend de la volonté du développeur à disposer d'une seule image à chaque coup.

Il faut savoir que la majorité des périphériques de capture d'images proposent des enregistrements de vidéos à un « framerate » de 30 images par secondes. Il faut donc lancer ce timer au plus de 30 fois en une seconde.

Le processeur d'un Smartphone actuel est capable de monter jusqu'à 1Ghz ou plus. On a donc une bonne marge de manœuvre sur la plateforme embarquant l'application.

DEFINITION D'UN TIMER DE RECUPERATION DES COORDONNEES SPATIALES

La 2e étape consiste à définir un temps de récupération avec horodatage des coordonnées spatiales du téléphone. Dans cette étape on va à intervalle régulier récupérer les coordonnées de l'accéléromètre et du gyroscope. Ce timer doit être activé au lancement de la fenêtre de scrutation en mémoire. On compte donc 2 timers différents dans notre algorithme à ce moment.
A ce stade on a un choix, créé un timer par périphérique gyroscope et accéléromètre, ou alors en utiliser qu'un et récupérer les 6 données. Il est à savoir que la récupération des coordonnées spatiales est extrêmement rapide comparée à la récupération d'images.

Les coordonnées spatiales sont enregistrées dans des listes. L'une contiendra celles qui concernent le gyroscope et l'autre celles qui concernent l'accéléromètre. Elles sont enregistrées de la manière suivante :

- On les récupère en mémoire tout en connaissant la modélisation de différence de temps entre le moment où on la trouve en mémoire et le moment où elle a été réellement prise
- On est donc capable de donner un horodatage précis aux coordonnées
- On remplit le tableau de coordonnées au fur et à mesure de la récupération avec l'horodatage précis

MODELISATION DE LA DIFFERENCE DE TEMPS DE RECUPERATION DE DONNEES

Cette étape est cruciale dans la création de l'algorithme final. Il est nécessaire de représenter la différence de temps entre la récupération de données spatiales et la récupération d'images. Une fois connue cette différence va permettre de créer un lien fiable entre l'image du flux vidéo et les coordonnées spatiales. La modélisation de la différence de temps est explicitée plus loin dans ce rapport.

REACTION SUR LA DETECTION D'UNE NOUVELLE IMAGE

Dans cette étape, nous allons considérer que nous avons trouvé une nouvelle image enregistrée en mémoire dans notre fenêtre de scrutation. Le premier travail est d'enregistrer le moment auquel nous avons trouvé cette image.
Nous allons donc modélisé la différence de temps entre le moment où nous la détectons et le moment où elle a réellement été prise. Nous avons ainsi une image avec son « instant » de capture précis.

LIEN AVEC LES COORDONNEES SPATIALES

On connaît la date précise de capture de l'image, on peut donc lier cette capture à des coordonnées. On doit donc parcourir les listes des coordonnées horodatées très rapidement pour récupérer celles qui sont le plus proche (avec une marge acceptable) de la capture de l'image. On va pouvoir créer le lien entre l'image et les coordonnées.

ENREGISTREMENT DE L'IMAGE EN FONCTION DU DEPLACEMENT

Dans cette étape, on dispose des images avec leurs propres coordonnées. En fonction des données spatiales et entre autre de l'accélération et rotation sur les axes, on va être capable d'envisager le recouvrement des images. C'est dans cette partie que va intervenir la vision par ordinateur. En effet, on est capable par les coordonnées du téléphone d'avoir les coordonnées de l'image et donc de connaître le recouvrement. On va créer une liste chainée des images pour pouvoir les parcourir très rapidement, éliminer celles qui ne nous intéressent pas et garder celles qui nous intéressent.

DEUXIEME PASSE DE SCRUTATION

Dans cette partie de l'algorithme, on s'intéresse à la passe de vérification de présence d'image dans la mémoire qui précède celle que l'on vient d'exécuter. On test une nouvelle fois le temps de récupération de l'image dans notre fenêtre de temps. Dans le cas où on aurait observé un décalage entre le lancement de la fenêtre et le moment où on trouve une image en mémoire, on peut alors recadrer notre fenêtre de scrutation autour de la dernière récupération. C'est ce qui va permettre de soulager le processeur dans la scrutation d'images.

BILAN DE LA PRESENTATION DE L'ALGORITHME

On peut donc observer un phénomène intéressant dans cette présentation d'algorithme. Nous nous intéressons à une récupération d'image qui rentre dans le cadre du temps réel. Nous allons garantir un temps de récupération le plus rapide possible grâce à nos timers paramétrable voulu mais surtout garantir une liaison avec les coordonnées le plus fiable possible.

ASPECTS MEMOIRE ET GESTION PROCESSEUR

En se plaçant dans un contexte d'algorithme de synchronisation pour un système embarqué, un autre aspect particulièrement important se présente à nous. Il est nécessaire de prendre en compte l'aspect limitations du support sur lequel on travaille. Nous allons donc chercher à limiter l'impact en mémoire et à limiter les cycles processeurs inutiles.

En ce qui concerne la gestion de mémoire et de processeur :

- Fonctionnement avec une mémoire DMA à priori donc processeur soulagé dans la récupération des photos
- On utilise des pointeurs sur la case mémoire ou est enregistrée l'image donc on soulage la mémoire
- On enregistre les coordonnées et le pointeur sur cette image, les coordonnées sont des données de petite taille donc la mémoire n'a pas de gros travail à effectuer
- On recadre la fenêtre de scrutation pour limiter l'impact processeur dans cette scrutation

On vient donc de rendre le processeur agile sur le travail du traitement d'images en limitant son utilisation.

LES METHODES DE GESTION DE PERIPHERIQUES

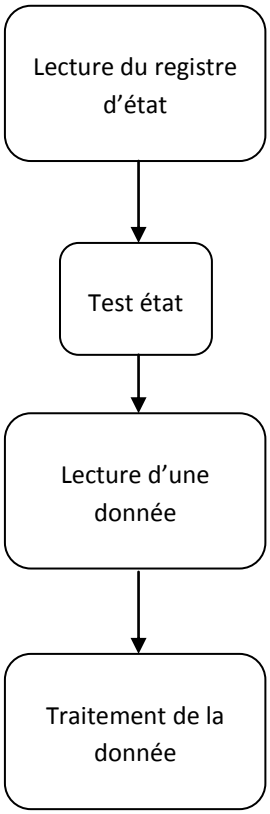
Dans cette partie du rapport, nous allons aborder les différentes méthodes de gestion des périphériques dans une architecture de système proche de celle d'un ordinateur. L'iOS est basé sur un OS BSD et bien que largement adapté à un Smartphone, le fonctionnement avec l'architecture du Smartphone est réellement très proche d'un ordinateur. Il existe plusieurs manières de faire appel à nos périphériques, chacune comporte ses avantages et inconvénients.
On peut trouver :

- Le mode d'échange par interrogation
- L'accès direct à la mémoire
- Les interruptions

LE MODE D'ECHANGE PAR INTERROGATION

Ce mode d'échange n'est à priori pas celui propre à l'iPhone pour la caméra. Mais il semble être proche de celui des périphériques gyroscope et accéléromètre.
Le processeur interroge à chaque instant les unités d'échange.
Ce dernier va venir lire les registres d'états et testé les bits concernés. Ensuite, les données reçues vont pouvoir être lues et stockées.

Le principe est schématisé ainsi :



Le réel problème avec ce type d'accès aux données est que le processeur est soumis à des cycles récurrents. Il est donc complètement mobilisé la récupération de données.

L'ACCES DIRECT A LA MEMOIRE (DMA : DIRECT MEMORY ACCESS)

Ce mode d'échange de données est celui qui va nous intéresser le plus. Dans le cas d'une caméra sur un Smartphone tel que l'iPhone, il est probable que ce type de technologie soit utilisée dans le but de soulager le processeur qui ne dépasse pas le GHz dans notre cas. De plus un téléphone comprend des services qui eux sont toujours actif et qui ne doivent pas être obstrués par un programme qui mobiliserait le processeur.

Le principe de la mémoire DMA est donc de permettre la récupération de données échangées directement dans la mémoire sans passer par le processeur.

Le principe de fonctionnement est le suivant :
La mémoire DMA se place entre le processeur et le contrôleur de périphérique.

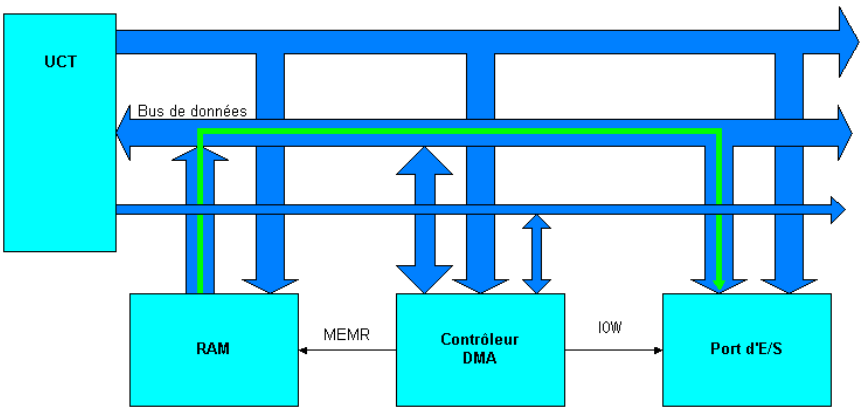


Figure 6 : Principe généraliste DMA

Comme on peut le voir sur ce schéma, le processeur n'intervient pas dans l'échange de données. Elles sont directement stockées en mémoire RAM.
L'iPhone 4 est sûrement basé sur ce principe en ce qui concerne la gestion du périphérique de capture d'images. C'est ce qui nous arrange en termes de réalisation.
Le seul problème que l'on pourrait rencontrer c'est un conflit entre le processeur et la DMA car quand cette dernière cherche à accéder à la mémoire RAM, le processeur ne doit pas le faire.

LES INTERRUPTIONS

Le denier moyen d'échange de données est le principe des interruptions. Dans les autres modes d'échange de données l'unique moyen de savoir si l'on peut rendre la main au processeur c'est de tester les registres d'états du contrôleur pour savoir si le nombre de mots transmis est passé à 0. Le temps de cette vérification monopolise inutilement le processeur. C'est pour cela que la technique des interruptions peut être particulièrement intéressante. L'interruption est très utile mais je n'ai pas trouvé le moyen de vérifier si cela était présent sur l'iPhone et dans tous les cas aucune interruption n'est utilisable pour vérifier la présence d'image dans la mémoire.

LA MODELISATION DES PERIPHERIQUES

Dans cette partie du rapport, je vais parler de la modélisation de notre architecture et du problème des deltas de temps qui interviennent dans l'algorithme. Comme vu dans l'algorithme ces temps vont avoir une importance particulière. En sachant les définir et les représenter, nous pourrons complètement implémenter l'algorithme.

MODELISATION GYROSCOPE

Le premier périphérique auquel nous allons nous intéresser est le gyroscope.

ARCHITECTURE

Voici le schéma de principe du fonctionnement de ce périphérique.

Bus de données
Bus d'Adresses
Bus de Contrôle

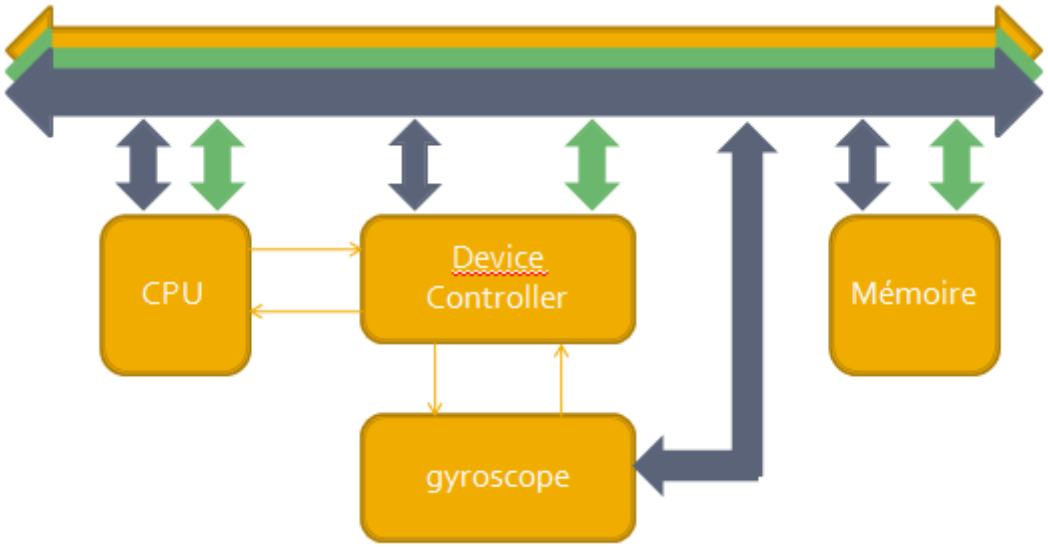


Figure 7 : Architecture gyroscope

MODELISATION

Nous allons maintenant passer à la modélisation mathématique du fonctionnement de ce périphérique. Le principe est de proposer un modèle mathématique qui sera la représentation du temps entre le moment ou l'action de récupérer les données est lancée et où les données sont en mémoire.

Nous allons prendre en compte :

- Fréquence du bus de périphériques : $fbus$
- Volume de données : $Vdata$
- Délai d'acquisition du capteur : $\Delta tacqccd$
- Délai d'acquisition du bus (si occupation) : $délai(cste)$
- Délai d'écriture dans la mémoire en fonction du volume de données à écrire : $\Delta twrite$

- La modélisation se présentera donc ainsi :

$$\Delta tacc = \Delta tacq + (fbus / Vdata) + \Delta twrite + délai$$

Le temps $\Delta tacc$ correspond donc au laps de temps qui nous servira dans la datation de la récupération des données. En toute logique ce laps de temps doit être très court car les données récupérées sont très légères.

MODELISATION ACCELEROMETRE

Nous allons maintenant nous intéresser à l'accéléromètre.

ARCHITECTURE

Voici le schéma de l'architecture de fonctionnement du périphérique.

Bus de données

Bus d'Adresses

Bus de Contrôle

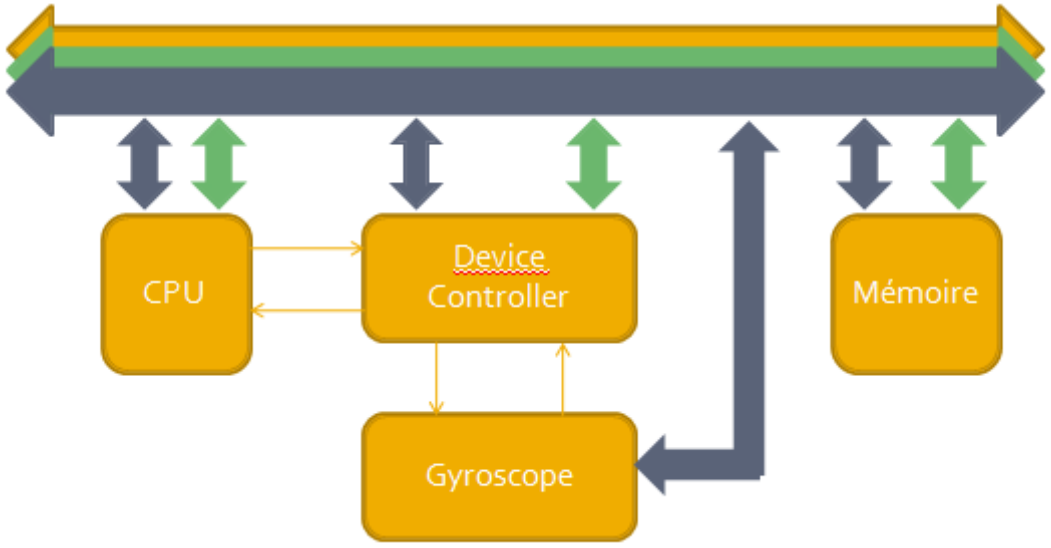


Figure 8 : Architecture accéléromètre

MODELISATION

Nous allons prendre en compte :

- Fréquence du bus de périphériques : $fbus$
- Volume de données : $Vdata$

- Délai d'acquisition du capteur : $\Delta tacqccd$
- Délai d'acquisition du bus (si occupation) : $délai(cste)$
- Délai d'écriture dans la mémoire en fonction du volume de données à écrire : $\Delta twrite$

- La modélisation se présentera donc ainsi :

$$\Delta tacc = \Delta tacq + (fbus / Vdata) + \Delta twrite + délai$$

MODELISATION CAMERA

Nous allons maintenant nous intéresser à la partie capteur CCD et donc périphérique d'acquisition d'images et flux vidéo.

ARCHITECTURE

L'architecture de ce périphérique se présente différemment dans le sens où l'on constate la présence d'un buffer de mémoire Direct Access Memory. Les images du flux vidéo sont placées dans cette mémoire afin de soulager le processeur dans la récupération des images.

Bus de données
Bus d'Adresses
Bus de Contrôle

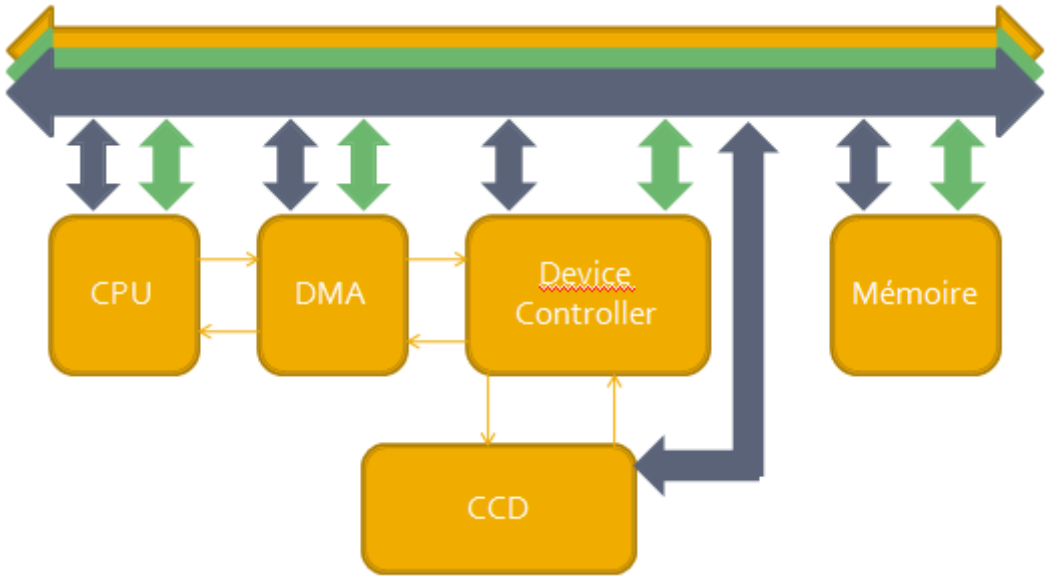


Figure 9 : Architecture caméra et DMA

MODELISATION

Nous allons prendre en compte :

- Fréquence du bus de périphériques : $fbus$
- Volume de données : $Vdata$
- Délai d'acquisition du capteur : $\Delta tacqccd$
- Délai d'acquisition du bus (si occupation) : $délai(cste)$
- Délai d'écriture dans la mémoire en fonction du volume de données à écrire : $\Delta twrite$

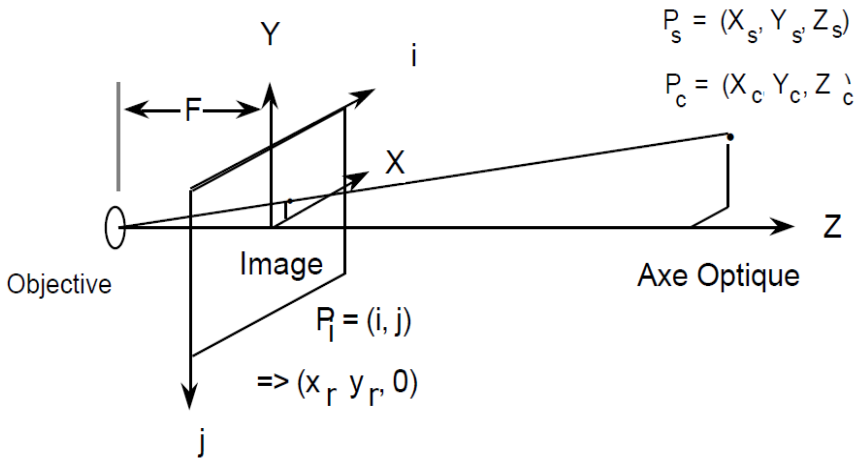
- La modélisation se présentera donc ainsi :

$$\Delta tacc = \Delta tacq + (fbus / Vdata) + \Delta twrite + délai$$

LA VISION PAR ORDINATEUR

Le dernier point que je tenais à aborder dans ce rapport, concerne la vision par ordinateur. Ce principe permet de créer un lien entre les coordonnées de l'objectif du système de capture d'images et les coordonnées du plan image capturé.
Le principe est de créer une projection entre l'objectif et son positionnement spatial avec l'image qu'il est en train de capturer.

Le principe est le schéma suivant :



Le terme « objective » désigne l'objectif du système de capture. Il possède ses propres coordonnées spatiales. Grâce à la connaissance de la focale de l'objectif du Smartphone, nous sommes en état de faire une projection des coordonnées de l'objectif sur le plan image. C'est la manière d'obtenir les coordonnées exactes du plan image.
Dans l'algorithme je n'aborde pas cette partie car je n'ai pas pu me plonger complètement dans la vision par ordinateur, mais à chaque lien créé entre une image et les coordonnées du Smartphone, il faut en fait créer les coordonnées du plan image par le biais de cette projection.

LES CODES REALISES SOUS XCODE POUR L'IPHONE

Malgré que l'iOS oblige à de grosses limitations en termes de récupération de données et de capacités sur l'OS. Par exemple le fait que la gestion d'interruption semble à priori impossible avec le SDK de l'iPhone.
Je me suis permis de réaliser des briques de code qui réalisent certaines parties du projet. Je vais donc les présenter et les adjoindre en annexe à ce rapport.

RECUPERATION DES COORDONNEES GYROSCOPE ET ACCELEROMETRE

Le premier code qui a été effectué est un code qui affiche à l'écran en temps réel les informations de coordonnées spatiales. De plus il les place dans des variables. Ce code présente un défaut, il ne sauvegarde pas chaque coordonnées mais écrase la précédente. La modification qui permettrait de stocker ces coordonnées ne serait pas complexe. On pourrait envisager plusieurs modifications en rapport avec ce projet.

- L'application sauvegarde les coordonnées tant que l'on n'a pas l'information qu'une nouvelle image vient d'être trouvée en mémoire
- Les coordonnées ne seront enregistrées que sous l'effet d'un timer prédéfini

AFFICHAGE DU FLUX DE LA CAMERA

Ce code n'est vraiment pas complexe et ne fait qu'afficher ce que nous renvoie la caméra. L'intérêt est qu'il peut facilement être placé avec le code qui enregistre les coordonnées. On aurait donc la partie caméra active avec en plus l'enregistrement des coordonnées spatiales.

DEBUT D'IMPLEMENTATION DE L'ALGORITHME

Encore une fois, malgré les limitations de l'iOS, j'ai débuté la programmation de l'algorithme de synchronisation des périphériques.

GESTION DE PROJET

Dans cette partie du rapport, je vais parler de l'aspect gestion de projet pour cette réalisation. Ce projet a été soumis à divers ralentissements dû à des problèmes de limitation de l'iOS. Je vais donc reprendre chacune des tâches et présenter la différence entre le prévisionnel et le réel.

PRECONCEPTION

ETUDE LA PROBLEMATIQUE

Cette partie a été réalisée dans le temps imparti. Malgré cela les découvertes de limitations et de passage à une réalisation plus théorique de ce projet l'ont rendu plus complexe et ont nécessite de reprendre l'étude du problème plusieurs fois.

PRISE EN MAIN DU SDK

Cette partie a aussi été faite dans les délais prévus. Elle a juste nécessité de prendre en main tous les outils de développement disponible sur iMac pour réaliser des applications iPhone.
Il a été nécessaire de prendre en main les outils suivants :

- XCode
- Interface Builder

REALISATION DU CAHIER DE SPECIFICATIONS

Cette partie du projet a aussi été réalisée dans les temps. Le cahier des spécifications a été envoyé à l'encadrant du projet pour que celui-ci le valide dans les temps. Le cahier des spécifications n'a jamais changé au cours du projet. Le projet a dérivé vers des aspects plus théoriques à cause de soucis de réalisations mais l'objectif est resté le même : réaliser un principe de récupération d'images en fonction du déplacement d'un iPhone (ou autre Smartphone car l'étude est devenue plus générale et théorique).

REALISATION DE L'ETUDE DE L'OS

Cette partie n'a pas été réalisé dans les temps à cause du fait que les sources manquent pour appréhender complètement le fonctionnement de l'iOS. C'est donc une étude partielle qui a été rélalisée. Le principe aurait été de clairement comprendre et assimilé des notions sur l'iOS. Je me suis donc limité à une forme d'étude de faisabilité par rapport au projet.

DEVELOPPEMENT

C'est dans cette phase que le plus grand retard a été pris. Ce n'a pas été dû fait que je n'étais pas dans les jalons mais du fait que les limitations de l'iOS nécessitait des recherches constantes pour se rendre compte que le résultat voulu n'était pas possible de la façon dont on l'avait convenu avec Mathieu Delalandre sur un OS aussi fermé que celui du Smartphone de Apple.

DEVELOPPEMENT DE LA PARTIE FLUX VIDEO

Cette partie a été partiellement réalisée dans le sens ou la récupération du flux vidéo à l'écran a été réalisée. En ce qui concerne l'enregistrement d'images du flux vidéo, il s'est avéré que la méthode envisagée était impossible. De plus, un réel manque de temps pour lé réalisation de ce projet s'est fait sentir.

DEVELOPPEMENT DE LA PARTIE GYROSCOPE ET ACCELEROMETRE

Cette partie a aussi été faite partiellement du fait que la partie gestion du flux vidéo n'était pas complète et fonctionnelle. La réalisation de cette partie ne permet que de récupérer les coordonnées spatiales du Smartphone.

DEVELOPPEMENT DE LA PARTIE GESTION DE PILE

Cette partie concernait la récupération des images dans la pile d'images du flux vidéo pour ne sauvegarder que celles qui nous intéresse. Encore une fois par manque de temps, les limitations de l'iOS, cette partie s'est transformée en une étude théorique de la méthodologie pour permettre de récupérer des images du flux vidéo selon les coordonnées des autres périphériques du système.

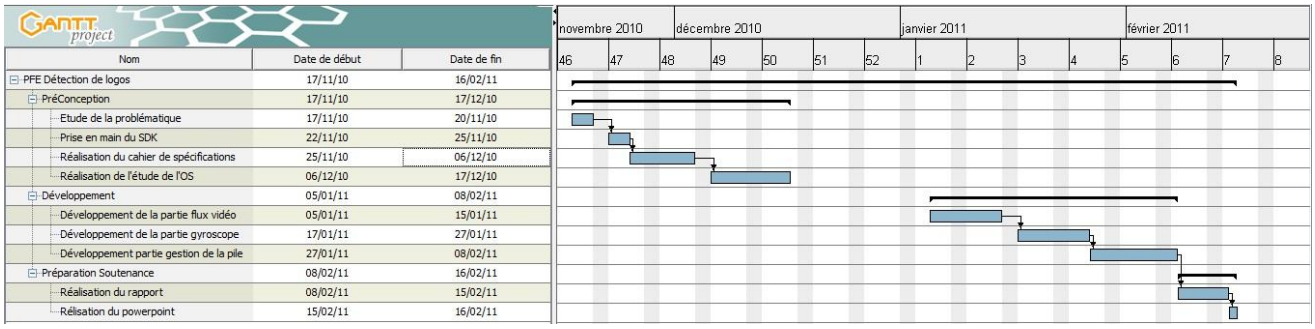
RAPPORT ET SOUTENANCE

REALISATION DU RAPPORT

Cette partie a été réalisée dans les temps.

PLANNING PREVISIONNEL

Dans cette partie, nous retrouvons le planning prévisionnel avec les dates butoirs fixées en accord avec Mathieu Delalandre.



Nom	Date de début	Date de fin
PFE Détection de logos	17/11/10	16/02/11
PréConception	17/11/10	17/12/10
Etude de la problématique	17/11/10	20/11/10
Prise en main du SDK	22/11/10	25/11/10
Réalisation du cahier de spécifications	25/11/10	06/12/10
Réalisation de l'étude de l'OS	06/12/10	17/12/10
Développement	05/01/11	08/02/11
Développement de la partie flux vidéo	05/01/11	15/01/11
Développement de la partie gyroscope	17/01/11	27/01/11
Développement partie gestion de la pile	27/01/11	08/02/11
Préparation Soutenance	08/02/11	16/02/11
Réalisation du rapport	08/02/11	15/02/11
Rélisation du powerpoint	15/02/11	16/02/11

Figure 10 : Planning prévisionnel

PLANNING REEL

Dans cette partie, nous pouvons retrouver le planning réel du projet. Celui-ci a été très dérangé du fait que le projet a rencontré de gros freins dû au développement sur iPhone.
Voici les dérives constatées :

- Préconception
  - Etude de l'iOS plus complexe du fait d'un manque d'informations
    - Débordement sur le développement de la partie flux vidéo
- Développement
  - Cette partie n'a pas été respectée dans le sens ou elle s'est transformée en une étude théorique et générale de la manière de réaliser une primitive de synchronisation de périphériques adaptable à tous les systèmes du type Smartphone

CONCLUSION

Ce projet m'a permis de prendre en main des technologies innovante. Mais surtout d'aborder un thème très en vogue et intéressant, la réalité augmentée. De plus, j'ai pu encore une fois prendre en main des outils de gestion de projet qui m'ont permis de bien me rendre compte de l'avancement.

Ce projet a aussi été motivant du fait que c'est un thème de recherches et de développement assez poussée avec des notions nouvelles en ce qui me concerne. J'ai pu, en effet, rencontrer de l'aspect temps réel et ordonnancement, de l'aspect vision par ordinateur, de l'aspect architecture des systèmes…

Ce projet a donc été enrichissant du point de vue des connaissances acquises.

TABLE DES FIGURES

BIBLIOGRAPHIE

- Programmation iPhone OS 3 – Thomas Sarlandie – Editions Eyrolles
- Architecture des Ordinateurs - Gérard Blanchet/Bertrand Dupouy – Telecom ParisTech
- Vision par ordinateur – Radu Horaud/Olivier Monga – Edition Hermes
- Cocoa and Objective-C – Scott Stevenson – Edition O'Reilly

ABSTRACT

This project is based on the augmented reality. Actually, AR applications use the GPS location on a Smartphone to display informations about the environment over the human vision with the video flow of a camera. The aim of this project is to propose a technology of AR which permits to use the gyro, the accelerometer and the camera. These 3 devices allow enhancing object recognition for displaying useful informations. So, it needs to develop a new form of devices synchronization in a mobile system. This report explains the devices synchronization algorithm and the devices modeling.