

MegaUAV 2013

A.JOURQUIN, A.HULAK, A.MARSOULAUD, C. NGUYEN

June 17, 2013

Contents

1	Introduction	3
2	Cadrage du projet	3
2.1	Besoin du client	3
2.2	Description fonctionnelle	4
2.3	Schéma matériel envisagé	4
3	Présentation du matériel	4
3.1	Matériel du Quadricoptère	4
3.2	Matériel Embarqué	5
3.3	Architecture finale	6
4	Communication	7
4.1	Architecture des communications	7
4.2	Communication série	8
4.2.1	Série inter carte	8
4.2.2	GPS	8
4.3	Réseau	9
5	Vol Stationnaire	12
5.1	Algorithme	12
5.2	Estimation du déplacement	12
5.3	Extraction des points d'intérêt	13
5.3.1	Détection des contours	13
5.3.2	Détection des coins	13
5.3.3	Détecteur de Harris	14
5.3.4	FAST	14
5.4	Matching	15
5.5	Compensation du mouvement	15
6	IHM	16
6.1	Connection	17
6.2	Etat général	17
6.2.1	Battery	17
6.2.2	Pitch	17
6.2.3	Roll	17
6.3	Control	18
6.4	Carte	19
6.5	Options	19

1 Introduction

Philippe GAUSSIER est le responsable du laboratoire ETIS, spécialiste en sciences de l'information et de la communication, il travaille sur la modélisation des mécanismes cognitifs impliqués dans la perception visuelle (vision préattentive et attentionnelle) et sur la modélisation de structures telles que l'hippocampe (pour des problèmes d'intégration d'informations spatio-temporelles). Ces modèles servent de base pour imaginer des architectures de contrôle permettant à des robots mobiles d'apprendre à survivre dans un environnement à priori inconnu. Ces robots utilisent généralement la vision comme source principale d'information ce qui met en avant l'importance du traitement de l'image. Les tâches étudiées sont : l'apprentissage de conditionnements visio-moteurs, la reconnaissance de lieux (retour au nid), la construction et l'utilisation de cartes cognitives et enfin l'apprentissage de la discrimination d'objets et/ou de lieux.

2 Cadrage du projet

2.1 Besoin du client

L'objectif du client est de concevoir un drone pouvant effectuer un voyage de façon entièrement autonome en ne connaissant que la position GPS du point de départ et du point d'arrivée. Pour cela, il est mis à notre disposition une plateforme pour effectuer nos tests, à savoir :

Un modèle réduit de quadricoptère très maniable, simple d'accès et facile à mettre en oeuvre (possibilité de tests en intérieur, en revanche sensible aux intempéries). Les cartes de développement, commandées en double, pourront nous permettre de répartir les tâches de développement en deux groupes.

Les applications possibles du drone peuvent être très diverses :

- Exploration de zones difficilement accessibles
- Entretien matériel et vérifications sur les avions, les buildings, les ponts etc.
- Surveillance d'une zone et reconnaissance des lieux et objets
- Recherche de personnes disparues (avalanches, naufrages, catastrophes naturelles)
- Surveillance de foule lors de manifestation.
- Détection d'intrusion, d'incendie de forêt

Il est essentiel et cela fait partie des objectifs finaux du projet d'avoir un vol autonome lors de la perte du signal avec le drone. Ce dernier devra effectuer sa mission avant de revenir au lieu de départ.

Le projet est évidemment complexe et présente beaucoup de possibilités.

Le client serait très satisfait d'obtenir à la fin du projet une communication

robuste et temps réelle avec un tour opérateur pour visualiser l'évolution de la mission et un système en boucle fermée sur l'hélicoptère qui lui permettrait des déplacements en toute autonomie.

2.2 Description fonctionnelle

Le drone principal est un quadri-rotor qui devra fournir les fonctionnalités suivantes :

- Communiquer avec un tour opérateur afin d'envoyer les paramètres de vol
- Maîtriser les commandes de vol sans intervention extérieure
- Etre capable d'effectuer une mission GPS et de naviguer en autonomie
- Etre capable de se repérer dans un environnement quelconque à l'aide de capteurs et d'imagerie
- Etre capable de reprendre la main à tout moment avec une radiocommande. Le tour opérateur correspond à un PC portable quelconque situé à terre et qui devra fournir les fonctionnalités suivantes :
- Afficher les informations de vol dans une interface conviviale et les stocker pour des besoins de debug
- Reconfigurer les missions à distance

2.3 Schéma matériel envisagé

L'architecture du système est composée des éléments suivants :

- Un PC portable, sur lequel est installé le logiciel de supervision et qui se charge d'enregistrer les paramètres de vol
- Une maquette de drone en modèle réduit (type hélicoptère, jusqu'à 4 rotors)
- Une certaine quantité de matériel embarqué sur le drone (Intelligence) ainsi que les différents capteurs et les moyens de transmission.

3 Présentation du matériel

3.1 Matériel du Quadricoptère

Pour ce projet nous allons utiliser un quadricopter de mikrokopter.

Le modèle est : MK Basicset L4-ME.

Les principales caractéristiques de ce modèle sont les suivantes :

Ce kit contient la carte Flight-Ctrl V2.1 ME dont les caractéristiques sont les suivantes :

Caractéristique MK BasicSet L4-ME	
Autonomie	15 min
Charge utile	500g
Masse à vide	86g
Surface	45cm ²
Structure	Fibre de carbone

Figure 1: Caractéristiques L4-ME

Et la carte contrôlant les moteurs :

Flight-Ctrl V2.1 ME	
Processeur	ATMEGA 128P-AU
Contrôleur	Roxy 2827-35 - black Brushless Motors
USB	MK-USB
Radio	PPM / ACT DSL
Capteur	<ul style="list-style-type: none"> - pressure sensor MPX4115A - 3 axes gyroscope / accelerometer
Bus	I2C

Figure 2: Caractéristiques Flight-Ctrl

3.2 Matériel Embarqué

Le matériel embarqué décrit comme suit sert à exécuter l'intelligence artificielle et donc de contrôler le quadricopter :

Overo Air COM						
Processor:	Texas Instruments		OMAP	3503	Applications	Processor:
	- ARM Cortex-A8 CPU					
Clock(MHz):	600 MHz					
Performance:	Up to 1200 Dhrystone MIPS					
Memory:	512MB				RAM	
	512MB Flash					
Features:	OMAP3503 Application Processor 802.11b/g wireless communications Bluetooth communications microSD card slot TPS65950 Power Management					
Expandability:	via one 140-pin expansion board of Gumstix Overo series or custom, 140-pin expansion board via 27-pin camera board					
Connections:	- (2) 70-pin connectors with 140 signals for: - I2C, PWM lines (6), A/D (6), 1-wire - UART, SPI, Extra MMC lines - Headset, Microphone - Backup battery - High Speed USB Host and USB OTG (1) 27-pin connector with signals for camera board (2) x u.fl antenna connectors					
Power:	Powered via expansion board (Overo series or custom) connected to dual 70-pin connectors					
Size:	17mm x 58mm x 4.2mm (0.67 in. x 2.28 in. 0.16 in.)					
Weight:	GUM3503A @ 42.6g (incl. shipping case & 2 x antenna) GS3503A @ 5.6g					
Products Included:	GUM3503A includes: - 1 x GS3503T (Overo Air COM)) - 2 x ANT006 (U.FL Antenna) - 4 x retaining spacers for Overo COM (CONH0020)					
Temperatures:	Built with components rated 0C < T < 75C					

Figure 3: Caractéristiques Gumstix

3.3 Architecture finale

L'architecture finale choisie est donc la suivante :

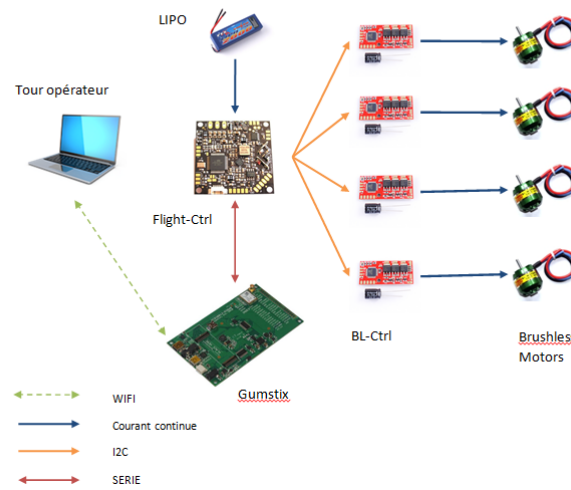


Figure 4: Architecture de la communication

4 Communication

4.1 Architecture des communications

Nous avons deux cartes : une Microkopter, la Flight-Ctrl, qui est la carte du drone et un Gumstix sur lequel notre programme fonctionne. La communication

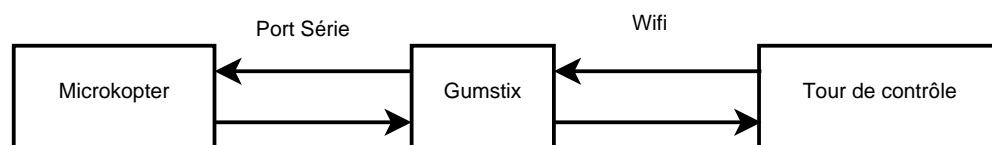


Figure 5: Architecture de la communication

entre les deux cartes se fait par port série sur

```
/dev/ttyUSB0
```

Le programme gpsd va analyser et convertir les infos du capteur gps en chaine de caractères.

les infos GPS sont récupérées directement sur la gumstix via le port série:

```
/dev/ttyS0
```

4.2 Communication série

4.2.1 Série inter carte

Afin de pouvoir faire communiquer le gumstix avec la carte Flight Control de Mikrokopter, il a fallu respecter le protocole de liaison série défini pour cette carte de vol.

Le protocole est basé sur des trames de données organisées de la façon suivante :

```
|START BYTE|ADDRESS BYTE|ID-BYTE|DATA BYTES[Base64]|CRC 1|CRC 2|STOP BYTE|
```

L'octet d'adresse permet de spécifier le type de carte, dans notre cas il est égal à 1, correspondant à la Flight Control. L'identifiant définit le type de la requête. Dans le cadre de notre projet, nous utilisons l'identifiant "-d" qui permet de récupérer une structure de données contenant toutes les informations de vol.

Le gumstix se charge de demander périodiquement les informations courantes de la carte de vol contenues dans cette structure afin de les transmettre au PC de contrôle. Le format de la structure est détaillée en annexe.

Les trames n'étant pas transmises en une fois par la liaison série, le thread de communication du gumstix se charge de détecter chaque début et fin de trame puis de la reconstruire pour pouvoir la décoder. Les données étant codées en base 64 il a fallu réaliser le décodage avant de les interpréter.

Nous n'avons trouvé que peu d'informations sur le décodage des données au sein de la documentation officielle de Mikrokopter, et la communauté présente autour de ces cartes et s'étant déjà confronté à ce problème, nous a permis de le faire rapidement et de ne pas perdre de temps sur cette partie technique.

4.2.2 GPS

la trame gps sur port série est lisible via cette commande :

```
cat /etc/ttyS0
```

En cas de problème avec la trame GPS, tapez cette commande :

```
killall gpsd; gpsd /dev/ttyS0
```

Notre programme va aller lire dans ce descripteur pour récupérer la trame GPGGA.

Détail:

Trame GGA

Nous utiliserons la trame identifiée GGA dans nos algorithmes dont la spécification est donnée ci-dessous:
\$GPGGA,hhmmss.ss,lll.ll,a,yyyyy.yy,a,x,xx,x.x,x.x,M,x.x,M,x.x,xxxx

Name	Example	Data Description
Sentence Identifier	GPGGA	Global Positioning System
Time	170834	17:08:34 Z
Latitude	4124.8963, N	41d 24.8963' N or 41d 24' 54" N
Longitude	08151.6838, W	81d 51.6838' W or 81d 51' 41" W
Fix Quality:(0,1,2)	1	Data is from a GPS fix
Number of Satellites	05	5 Satellites are in view
Précision horizontale	1.5	Accuracy of horizontal position
Altitude	280.2, M	280.2 meters above mean sea level
Height of geoid above WGS84 ellipsoid	-34.0, M	-34.0 meters
Time since last DGPS update	blank	No last update
DGPS reference station id	blank	No station id
Checksum	*75	Checksum

Figure 6: Trame GPGGA

Nous gardons la longitude, la latitude, le nombre de satellites , la précision horizontale et l'altitude. Exemple d'une trame GPGGA sur le port série:

```
GPGGA,11373.00,4902.59765,N,00205.00247,E,1,09,1.07,73.1,M,46.0,M,,*65
```

4.3 Réseau

Afin d'assurer la communication entre le drone et la tour de controle il a été nécessaire de développer un protocole de communication appelé MUAVCOM

On distingue deux canaux:

- un canal où la source d'émission est le drone, il envoie ses informations en continu à la tour de contrôle (batterie, accéléromètre, état, position gps, image de la caméra etc ...).
La tour de controle envoie un acquittement pour certaines infos critiques.
- un canal de pilotage dans lequel la tour de contrôle va envoyer des ordres : pour changer le comportement du drone, passer du mode manuel au mode

auto, choisir de l'ia à utiliser, envoyer des commande de vol, etc ...

Le protocole est construit comme ceci:

- une entête dans laquelle sera renseigné l'id du drone, l'id de la flotte du drone
- le type de requête et un code d'erreur si besoin, ensuite vient les informations de la requete en question

Ci-dessous la liste des requetes :

- **PILOTE_REQ_MANUAL**: on demande au drone de passer en mode manuel
- **R_PILOTE_REQ_MANUAL**: acquitement du passage au mode manuel
- **SEND_INFO**: envoi des informations du drone
- **R_SEND_INFO**: acquitement de reception des infos
- **EMERGENCY**: (non utilisé) indication que le drone se trouve en situation critique (ex: plus de batterie)
- **MISSION**: non utilisé, à définir
- **PILOTE_MANUAL**: envoi des informations de pilotage manuel
- **PILOTE_REQ_AUTO**: on demande au drone de passer en mode automatique
- **R_PILOTE_REQ_AUTO**: acquitement du passage au mode auto
- **PILOTE_REQ_OFF**: on demande au drone de passer dans l'état inactif
- **R_PILOTE_REQ_OFF**: acquitement de cette état
- **SEND_IMG_SIZE**: on envoi la taille de l'image que la caméra capture
- **R_SEND_IMG_SIZE**: acquitement de la reception de cette taille
- **SEND_IMG**: envoi d'une partie d'image
- **SEND_GPS_INFO**: envoi des information GPS
- **HELLO**: message envoyer à la tour de controle pour indiquer la mise en marche
- **R_HELLO**: réponse de la tour de controle au hello
- **R_SEND_IMG**: acquitement de la reception de l'image
- **GPS_INFO_START**: on demande au drone d'envoyer les infos

- **GPS_INFO_STOP**: on demande au drone d'arreter d'envoyer les info gps
- **IMAGE_SEND_START**: on demande l'envoi des images capturés par la caméra
- **IMAGE_SEND_STOP**: on demande d'areté d'envoyer l'image.

Datagramme MUAVCOM :

```
| IP|UDP|FLOTTE_ID[INT]|DRONE_ID[INT]|TYPE_R[INT]|TIMESTAMP[INT][INT]|ERROR[INT]|DATA|
```

l'entête muavcom commence à **FLOTTE_ID**, et se termine à **ERROR**, elle fait 24 octets.

Détail:

```
-image TYPE\_R = SEND\_IMG
```

```
| ID\_PART[BYTE]|IMAGE\_PART[BYTE]|
```

A cause du MTU par défaut (maximum taille des packets accepté) limité à 1500 octets, de la plupart des cartes réseaux, nous sommes dans l'obligation de découper l'image. Les images capturées font environ 25ko que nous envoyons en 25 parties.

Dans les données nous avons le numéro de partie de l'image suivi de la partie elle même.

```
TYPE\_R = SEND\_GPS\_INFO
```

```
|latitude:longitude:nombre de satellite:précision:altitude|
```

Les données GPS sont envoyées sous forme de chaines de caractères. On envoie seulement la latitude, la longitude, le nombre de satellites captés, la précision de la position horizontale et l'altitude.

```
-info microkoptère TYPE\_R = SEND\_INFO
```

Les informations de la carte microkopter sont envoyées dans un tableau de 32 int. Ce tableau, contient de niveau de batterie, l'état du quadri, les erreurs éventuelles, etc ...

```
-pilotage TYPE\_R = PILOTE\_MANUAL
```

```
|NICK[INT]|ROLL[INT]|YAW[INT]|GAS[INT]|
```

Lorsque que l'on pilote le quadri à l'aide du joystick, on envoie les infos qui vont indiquer la puissance des moteurs.

```
-ia TYPE\_R = PILOTE\_REQ\_AUTO
```

Lorsque l'on passe en mode automatique, on indique quel type d'IA on a choisi.

5 Vol Stationnaire

Pendant le vol, le quadricoptère est soumis à diverses forces extérieures perturbant la stabilité de ses mouvements, tels que la puissance des moteurs et le poids de l'appareil non homogènement répartis, le retour de force lié au brassage d'air des hélices, etc...

Nous cherchons donc dans un premier temps à compenser les déplacements non intentionnels de notre appareil afin d'effectuer un vol stationnaire.

5.1 Algorithme

Notre algorithme est basé sur la vision, obtenue grâce à une caméra reliée au Gumstix. Un traitement sur l'image servira à extraire des points d'intérêt qui serviront de référentiel au sol dont nous suivront les positions les coordonnées par rapport au temps. Ces coordonnées serviront à déterminer les mouvements de l'appareil par rapport à ces points au sol, sous forme de vecteurs, qui serviront ensuite à déterminer la compensation à effectuer afin de limiter les déplacements de l'appareil dans l'espace.

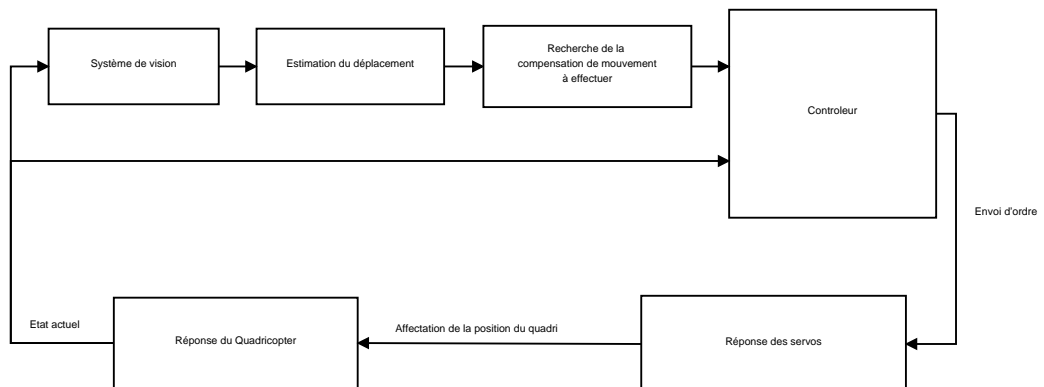


Figure 7: Algorithme

5.2 Estimation du déplacement

Pour estimer la direction de déplacement, l'algorithme se base sur la transformation affine de points d'intérêt entre deux images consécutives.

Le déplacement actuel de l'appareil est modélisé par un vecteur, déterminé par

la transformation géométrique entre 2 images. Pour $P_1(x,y)$ et $P(X,Y)$, les coordonnées d'un point d'intérêt sur l'image aux instants $t-1$ et t . La transformation T est représentée par :

$$\begin{bmatrix} X \\ Y \end{bmatrix} = T * \begin{bmatrix} x \\ y \end{bmatrix}$$

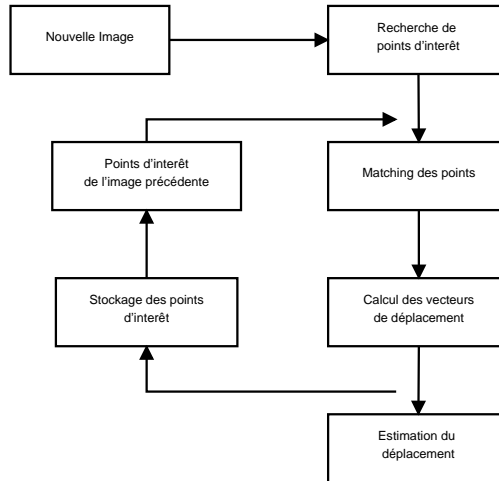


Figure 8: Traitement Image

5.3 Extraction des points d'intérêt

5.3.1 Détection des contours

Nous ne cherchons pas à proprement dit à détecter les contours mais nous avons besoin de calculer les gradients, horizontal et vertical, de l'image pour la détection des points d'intérêt. Nous utilisons pour sa fiabilité et sa vitesse d'exécution le filtre de Canny.

$$\begin{matrix} CannyH \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \\ CannyV \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \end{matrix}$$

5.3.2 Détection des coins

Nous avons choisi une détection de points d'intérêts par détections de coins (corner detection). Il existe de nombreuses méthodes de détection de points d'intérêt

telles que Harris, SIFT ou SURF. Nous avons choisi d'implémenter 2 méthodes par leur vitesse et leur robustesse :

- Le détecteur de Harris
- L'algorithme FAST

5.3.3 Détecteur de Harris

Le détecteur de Harris est un détecteur de coins basé sur les gradients orientés, en détectant des croisement de lignes. Le détecteur de Harris est robuste et fiable

$$G_x^2 * G_y^2 - G_{xy}^2 - \left(\alpha (G_x^2 + G_y^2) * (G_x^2 + G_y^2) \right)$$

Figure 9: Harris

mais est plus gourmand en ressources et en temps de calcul que l'algorithme FAST.

5.3.4 FAST

Pour chaque pixel p , un cercle de 7 pixels de diamètre centré sur p est extrait de l'image, nous cherchons un arc continu de N pixels (dans notre cas 9) plus clairs ou plus foncé que p (ou $p + \text{seuil}$).

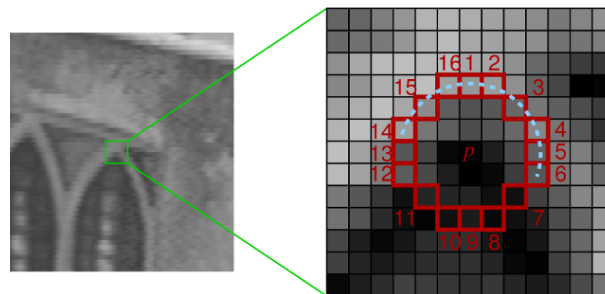


Figure 10: Fast

Cet algorithme est très rapide car il suffit de tester les points 1-5, 5-9, 9-13 et 13-1 pour déjà savoir si le point peut-être rejeté.

Cet algorithme est très rapide et les points trouvés sont fiables mais il est très sensible au bruit et dépend d'un seuil.

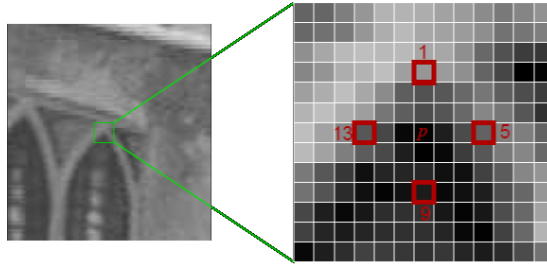


Figure 11: Fast

5.4 Matching

Une fois les points d'intérêt trouvés dans une image, il faut comparer la position qu'ils avaient dans l'image précédente. Pour cela, il faut d'abord matcher les points, c'est à dire faire la corrélation entre les points d'une image à l'autre, savoir si il s'agit du même point sur les 2 images.

Nous utiliserons les descripteurs des points fournis par les algorithmes de détection des coins que nous avons utilisé pour extraire les coins.

Si la valeur du descripteur d'un point est proche de celle d'un point sur l'image précédente et que leur différence de position est plausible, on peut considérer que ce point est le même.

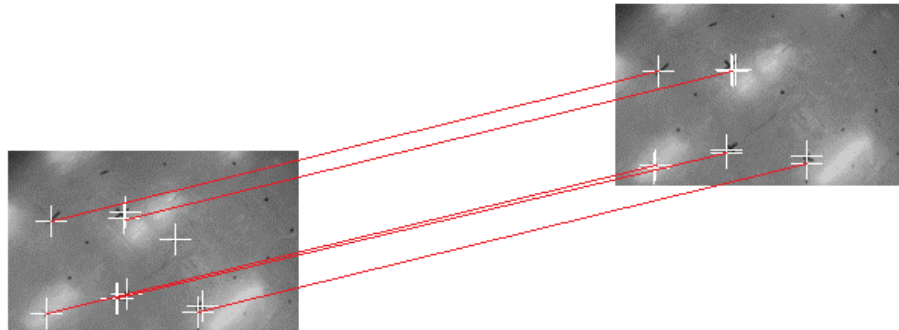


Figure 12: Traitement Image

5.5 Compensation du mouvement

Notre implémentation de la compensation du mouvement est basé sur une architecture réactive. La valeur inverse du vecteur déplacement est directement

envoyée en commande au quadricopter.

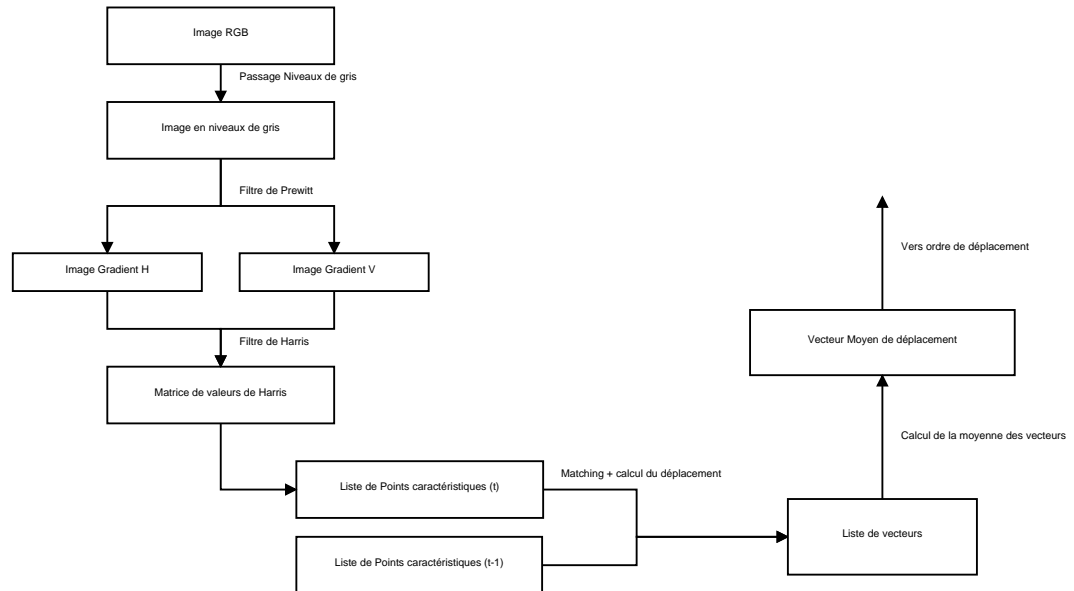


Figure 13: Traitement Image

6 IHM

L'ordinateur de contrôle permet de visualiser de nombreuses informations en provenance du drone et de lui envoyer des ordres. L'interface est constituée des panneaux suivants:

- Connection
- Etat général
- Battery
- Pitch
- Roll
- Control
- Carte
- Options

6.1 Connection

Le panneau situé en haut à gauche de l'interface permet d'afficher une liste des drones actuellement connectés et en fonctionnement. Lorsqu'un drone se met en marche, il envoie un message à l'ordinateur de contrôle en lui spécifiant son adresse IP. A sa réception par le PC de contrôle, le drone est ajouté à la liste en lui attribuant un numéro unique et son adresse IP est sauvegardée afin de pouvoir lui envoyer des ordres.

Le drone envoie périodiquement des informations de vol afin d'informer l'opérateur de son état. Un icône de statut verte permet de savoir si le drone sélectionné est toujours actif. Si le drone n'envoie plus d'information pendant un certain temps, l'icône devient rouge pour en informer l'opérateur.

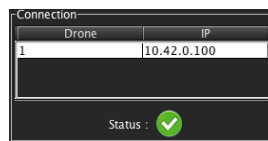


Figure 14: Panneau Connection

6.2 Etat général

Le panneau "current action" affiche à l'aide d'une image l'action courante du drone. Par défaut, le drone est immobile et les moteurs sont coupés. L'action suivante est l'allumage des moteurs. Ensuite, les actions varient entre prise d'altitude, vol à hauteur stable et perte d'altitude. Cela permet de rendre compte de l'état du drone en temps réel au cas où l'opérateur perdrait le contact visuel.

6.2.1 Battery

Le panneau affiche la charge restante dans la batterie connectée au drone. Lorsque celle-ci approche de la zone rouge, il est préférable par mesure de sécurité de faire atterrir le drone et de remplacer la batterie.

6.2.2 Pitch

Le panneau de pitch donne une information sur l'inclinaison actuelle du drone suivant l'axe avant - arrière.

6.2.3 Roll

Le roll permet de connaître l'inclinaison du drone suivant l'axe gauche - droite.



Figure 15: Panneau Etat Général

6.3 Control

Le panneau de contrôle affiche les images prises par la caméra embarquée sur le drone. Afin de recevoir les images, l'opérateur doit activer l'option "start image" dans le menu de l'application. Cette option ne peut être activée que lorsque le drone est en pilotage manuel, si ce dernier est contrôlé par l'intelligence embarquée il est impossible de récupérer les images car le Gumstix n'est pas assez performant pour gérer les deux tâches simultanément.

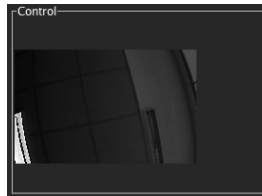


Figure 16: Panneau Control

6.4 Carte

La carte GPS permet de contrôler la position du drone et de lui transmettre les coordonnées d'une destination à atteindre.

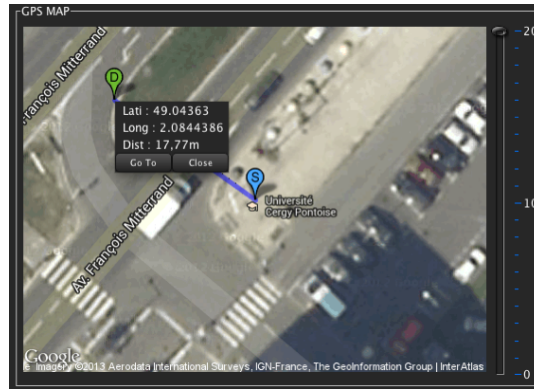


Figure 17: Panneau Carte

6.5 Options

Le menu options de l'application permet de changer l'algorithme d'intelligence artificielle tournant sur le drone, ou de passer en mode manuel afin de diriger le drone à l'aide du joystick. Lorsque le drone est en mode manuel il est possible de démarrer et d'arrêter l'affichage des images prises par la caméra embarquée et de recevoir les données GPS transmises.

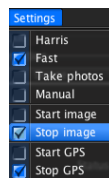


Figure 18: Panneau Options

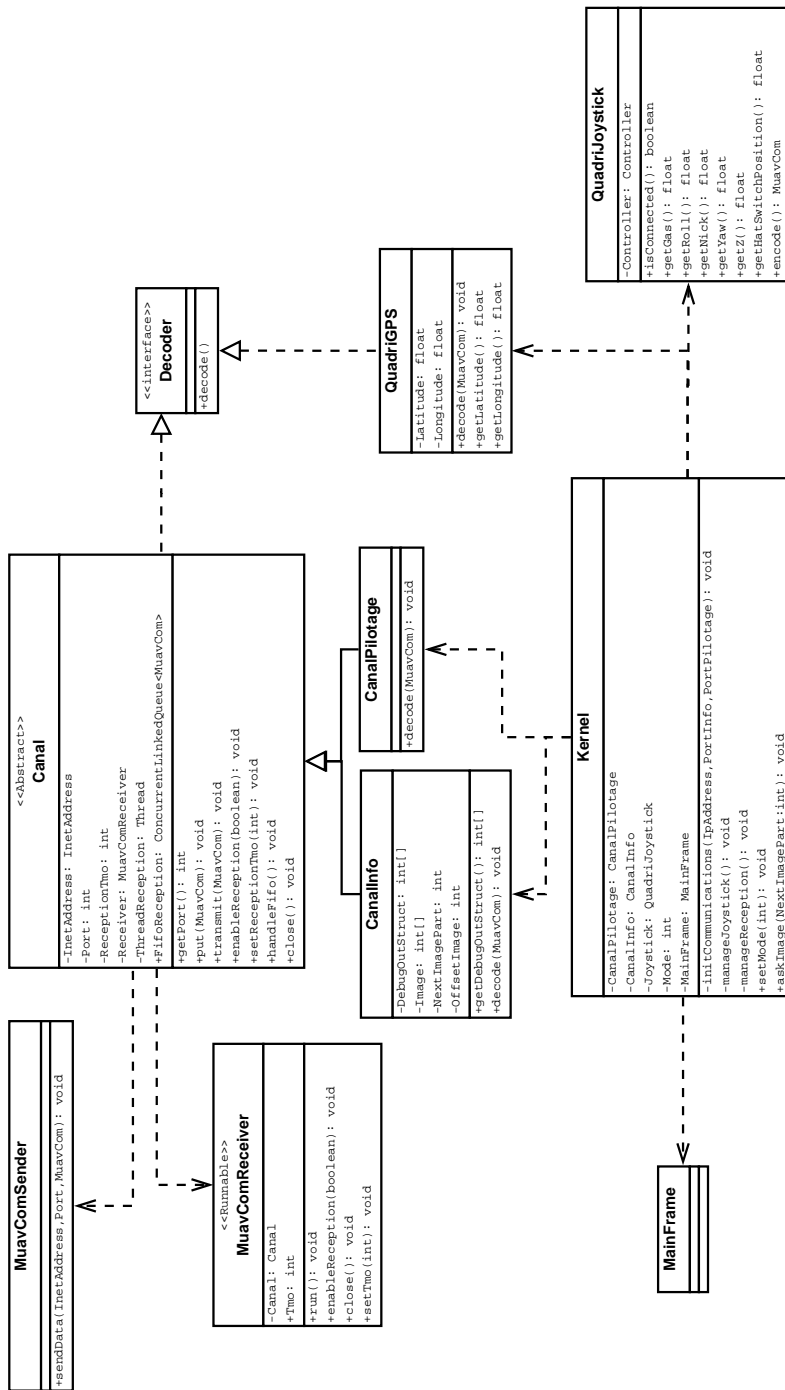


Figure 19: Diagramme des classes

