

# A Multi-Task Distributed Vision System Embedded on a Hex-Rotorcraft UAV

Nadir Younes, Boukhdar Khalid, Moutaouakkil Fouad, Hicham Medromi

Equipe Architectures des Systèmes (EAS), Laboratory of Research in Engineering (LRI)  
Université Hassan II of Casablanca, Ecole Nationale Supérieure d'Electricité et de Mécanique (ENSEM)  
Casablanca, Morocco

**Abstract**—In this paper, are presented the general architecture and implementation of a multi-task distributed vision system designed and embedded onboard a Hex-Rotorcraft UAV. The system uses multiple cheap heterogeneous cameras in order to perform various tasks such as: ground target pedestrian detection, tracking, creating panoramic images, video stabilization and streaming multiple data/video feeds over a wireless secure channel. In what follows, are discussed this multi-agent architecture designed to provide our UAV with an embedded intelligent vision system using autonomous agents entrusted with managing the previously listed functionalities. In addition to the cheap set of USB and module cameras, the presented vision system is composed of a Local Data Processing Module connected to each camera and a Central Module used to control the overall system, process the regrouped data and streams it to the ground station. The overall vision system has been tested in real flights and is still under improvements.

**Keywords**—multi-agent architecture; image processing; real-time systems; target detection; panoramic images; target following; unmanned aerial vehicles (UAVs); vision systems

## I. INTRODUCTION

In the last few decades, UNMANNED AERIAL VEHICLES (UAV) were employed and adopted in various sectors, ranging from surveillance in both military & industrial, to academic research, agriculture, etc. Specifically, UAV rotorcrafts were mainly used & upgraded by both defense and security communities [1], due to their easy maneuvering and vertical take-off/landing. More specifically, an UAV rotorcraft equipped with a vision system can be used to perform various tasks, ranging from objects investigation, patrolling, to tracking targets, etc...

Since vision is used as our native sensing mean, instead of using basic systems that are only able to acquire/send image & video feeds, researches were mostly focused on implementing embedded systems onboard UAVs able to perform multiple tasks such as: flight control using vision [1] [2], [3], [4], object detection & tracking, etc.

So far, research efforts were mainly geared toward the proposal of solutions answering to specific problems, which makes them optimized to perform the tasks they were conceived for, but they generally lack flexibility and scalability when deployed in other environments, or when they're to be upgraded to add new sensors or features. And so, it is rare to encounter in the literature documentations proposing

implementations, architectures of detailed and exhaustive vision systems designed for UAV rotorcrafts.

In order to bypass the enunciated inconveniences, the present paper introduces the hardware configuration and architecture of our embedded real-time vision system. The developed system is ought to be embedded, cognitive, scalable and based on a flexible multi-agent architecture, able to conduct a panel of image/video processing tasks using a set of low-coast credit-sized cards: Raspberry pi. The implemented software is running on a Raspbian [5] operating system featuring a real-time kernel. In order to develop a detection and tracking of moving targets feature, a variant of the CAMSHIFT algorithm was implemented [6]. This technique is a variant of the histogram based mean shift algorithm [13] altered so that it can adapt to the object's scale and rotation changes. Also, is featured a real-time panorama construction algorithm using simultaneously multiple heterogeneous camera streams. Finally, using the vision feedback, a video stabilization followed by a tracking control setup are presented in order to control a set of pan/tilt servomotors that keeps focusing objects of interest in the center of the video frame.

The rest of the paper is structured as follows. Section 2 briefly justifies the use of a multi-agent architecture as an infrastructure needed to deploy modular embedded vision systems on UAV rotorcrafts. In Section 3, the hardware characteristics are detailed. Section 4 details the architecture of the vision system. Section 5 details the deployment and configuration of the vision system blocs and their functionalities. Section 6 resumes the work and suggests future research lines.

## II. MULTI-AGENTS IN VISION SYSTEM

Monitored scenes by UAV vision systems usually are complex and dynamic environments where multiple tasks are to be performed and where the outputs are to be merged in order to display the user requested results. This is why multi-agent systems (SMA) can be seen as an intuitive solution used to devise the vision system into a set of intelligent agents that communicate and cooperate in order to perform the requested tasks.

## III. HARDWARE CONFIGURATION

This section describes the hardware configuration of the embedded vision system designed for small UAV. The hardware platform is mostly composed of various video sensors linked to a couple of credit-card sized computer

boards, bounded together using an implemented communication middleware in charge of the tasks distribution, computations and centralization of the outputs sent to the ground station. The hardware configuration that has been used can be justified by the following arguments:

#### A. Central processing module

First of all, this manipulation aims to implement a low-cost, low power consumption, compact, autonomous and real-time vision system onboard a small UAV rotorcraft. Since the platform ought to run real-time applications, a real-time operating system was chosen to support the implementation [5]. In addition, it is highly recommended to use a hardware platform that supports libraries of programming functions aimed at real-time computer vision such as Aforge, SimpleCv, OpenCv (Open Source Computer Vision), etc. And finally, it is suitable to work with a hardware platform that can use cheap, small and heterogeneous video sensors in addition to regular USB, RS232 or wireless cameras. According to the literature, more than a dozen of small board cards can satisfy parts of these constraints such as Arduino cards, NanoPC- T1, BeagleBone Black or ODROID-XU4, etc... . To meet up all of our expectations, we ended up hesitating between two processing boards: the Raspberry Pi and the BeagleBone Black cards. After comparing all these boards specifications [16], our mind was finally set on using RPI due to their large developing community, diversity of compatible modules that it offers and most of all because it possess a powerful integrated Video-core graphics processor able to decode up to 1080p video streams which is suited in computer vision applications. Is it important to note that this board sole purpose is to run the UAV vision system, and up till now it does not intervene in the flight control process. The separation of the vision system and the flight control system [17] into different computing boards can be justified by the following arguments:

- First, due to the fact that the computation consumption of both these tasks is way too heavy, running them both simultaneously on the same embedded computer board is not possible.

- Also, by distributing the system we can guarantee a better stability of the overall system by protecting the flight system from potential latencies caused by data overload.

#### B. Visual sensors

Various heterogeneous color video cameras were picked as visual sensors in our system. The default set of cameras that is used in our system is composed of:

- **Camera Module:** This camera plugs into the CSI connector located between the Ethernet and HDMI ports. The cost of the camera module is € 20 in Europe (9 September 2013). [7] It can produce 1080p, 720p and 640x480p video. The dimensions are 25 mm x 20 mm x 9 mm and it weighs less than 30g [7].
- **HERO3+ Black edition camera:** this device offers various functionalities such as the so called “SuperView” which increases the field of view, a panel video modes (1440p48, 1080p60, 960p100 and 720p120 as well as 4K15 and 2.7K30), can shoot 12MP

still at up to 30 frames per second and also includes a Wi-Fi Remote [8].

- **HD PRO WEBCAM C920:** simple USB webcam, able to save up to 1080p videos.

#### C. Pan/Tilt servomechanism

In order to maintain a constant visual contact with followed targets, a pan/tilt servomechanism had been used to mount the module cameras, enable them to rotate up to 180 degrees horizontally and 110 degrees vertically.

#### D. Wireless data link

In order to ensure a permanent wireless communication between the ground operators and the vision system, a 150 Mbps wireless adapter is used in order to perform a Wi-Fi communication -to transmit commands and receive visualization data- between the ground control and the embedded system

### IV. PROPOSED ARCHITECTURE

#### E. Abstract view of proposed architecture

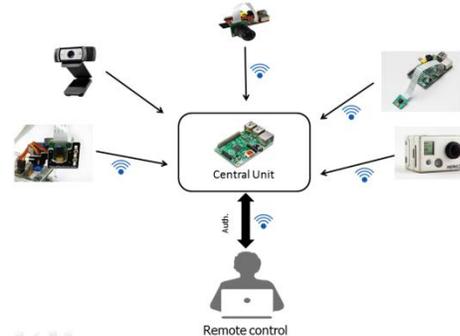


Fig. 1. Abstract view of proposed architecture

As shown in Fig.1, the architecture is mainly composed of a set of raspberry pi board and various types of cameras (USB, module, GoPro ...). Each board can be considered as an independent sub-layer where visual data can be extracted and preprocessed before being sent to the central unit board. In a concern for scalability (possibility to add or suppress boards), and also to keep the system compact as much as possible, communications between these different components have been kept essentially wireless (Wi-Fi).

#### F. Architecture overview

The system overall architecture (Fig.2) can be structured into three main layers: a reactive layer, deliberative layer, and a user layer. Interactions and communications between these layers are maintained using a communication middleware. Next, characteristics of each level are summarized:

- Basically, the reactive layer is composed of heterogeneous video sensors used to obtain visual information of the UAV in-flight surroundings in addition to agents charged with the unification of the data format on a reactive level.
- The deliberative layer can functionally be split into sub-layers. The first sub-layer (the panoramic editing

unit) host agents that register the exploitable collected uniformed data in order to merge them into panoramic images. While in the second sub-layer, are agents entrusted with the construction of a knowledge model of the environment, so that it can be used in various tasks (such as objects recognition, maintaining the

overall system connected to the ground station, pan/tilt object tracking using a set of servomotors).

- The user level allows users to interact and monitor the vision system via a developed web interface.

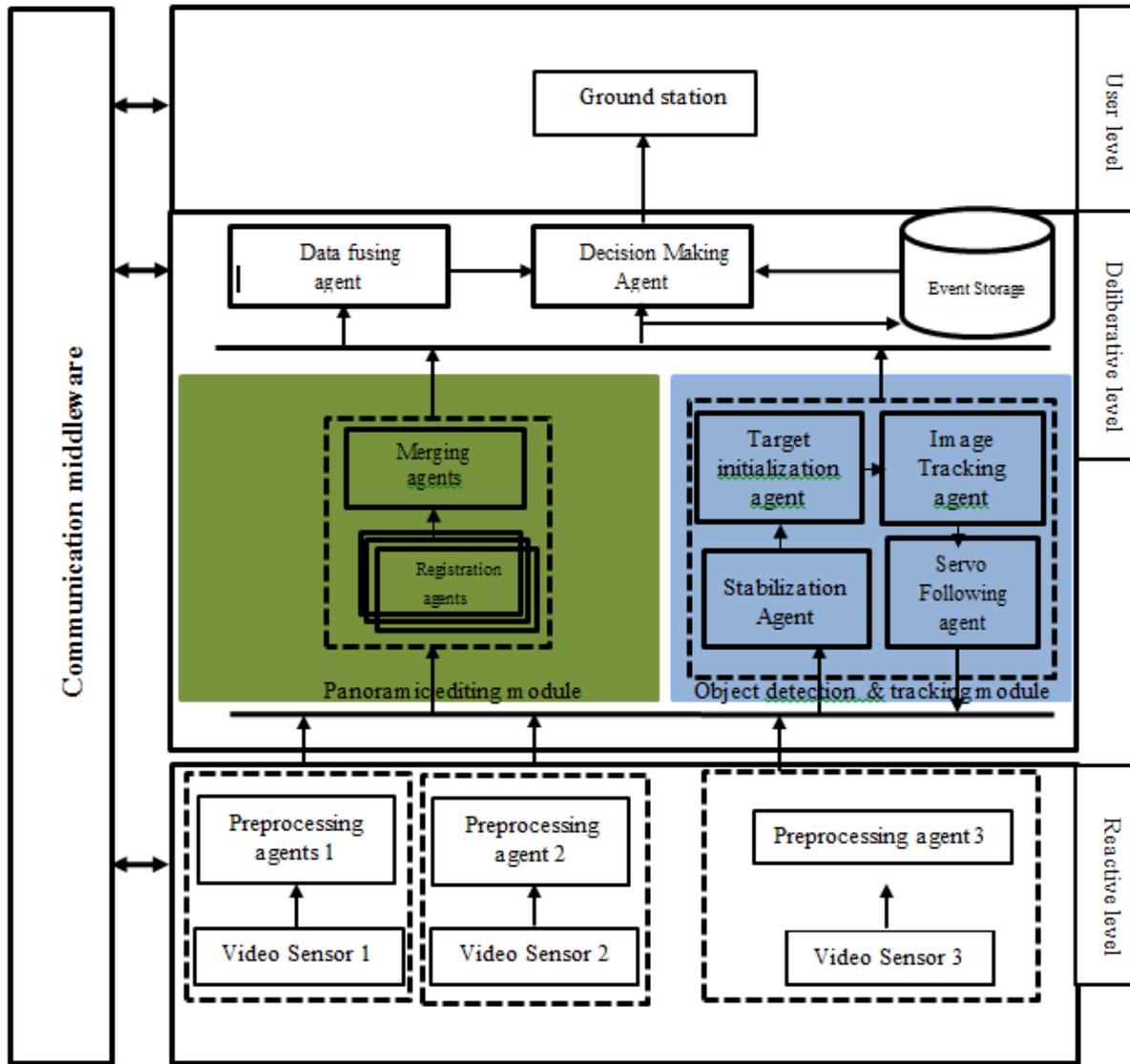


Fig. 2. Architecture of an intelligent and multi-tasked vision system for small UAV

## V. DEPLOYMENT AND CONFIGURATION OF THE VISION SYSTEM BLOCS

Each layer in this system incorporates different classes of agents. These agents deal with the problematic of performing simultaneously multiple computer vision tasks using incoming streams of heterogeneous video sensors by treating them locally first on their respective boards, followed by a unification of the retrieved formatted data. In what follow, the data flow between these different layers is detailed.

### G. Reactive level: Data Preprocessing

On the reactive level, at the start of the overall process, preprocessing agents are deployed at the end of each video sensor to process the gathered data. And so, these agents are used to unify the gathered streams from the heterogeneous cameras into uniform classes (a VideoClass for video streams and an ImageClass for images) with standard spatial/temporal resolutions. The spatial synchronization mainly consists of a resizing of all the videos based on the one with the smallest resolution while the temporal synchronization is insured by adjusting the FPS processing rate based on the lowest one. This entire process can be summarized in the following figure (3).

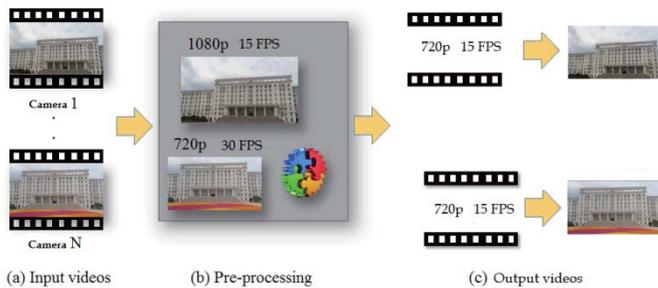


Fig. 3. Pipeline of video preprocessing: (a) input videos that are captured by heterogeneous cameras, (b) auto video synchronization based on lowest FPS/resolution, (c) Output videos generated

#### H. Deliberative level

On the deliberative level, once the acquired data is uniformed, two scenarios can take place depending on the nature of the requested task.

In the *panoramic editing module*, are input the collected preprocessed video streams in order to create real-time panoramas. By altering the Panoramic Image Stitching algorithm presented by M. Brown and D. Lowe [9], we were able to extend this technique to real-time videos at the cost of performing it only when the UAV is in stationary flight. The reason behind such a limitation will be addressed later on.

#### Registration and Merging Agents:

Component heads identify the different components of your Assuming that the UAV is performing a stationary flight in order to keep the cameras still, registration agents are used to detect a difference of Gaussian keypoints (DoG) [10] and extract local invariant descriptors SIFT [11] from the previously unified received images. Next, merging agents loop over the previously computed descriptors, compute the distances, find the smallest distance for each pair of descriptors, computes the matches for each pair of descriptors using Lowe's ratio test [9] and estimate their homography matrix by applying RANSAC algorithm [18] on the matched feature vectors. Finally, using the previously created homography matrix, a warping transformation produces a panoramic image that is sent to the user level. The reason behind insisting on having to apply this method only on still cameras resides in the fact that performing these tasks (keypoint detection and matching, SIFT descriptor detection and especially estimating the homography matrix) on successive frames can be computationally heavy. So, applying this algorithm on videos received from moving cameras would makes us estimate the homography matrix for each set of frames, making it unmanageable to run it in real-time. However, if by assuming that the cameras are still (as in the case of a stationary flight), the estimation of the homography matrix would only be computed once, resulting in the creation of a video panoramic view using multiple cameras. The overall algorithm can be reviewed in figure 4 while figure 5 gives an inflight example of the image stitching method using frames taken from three video sensors

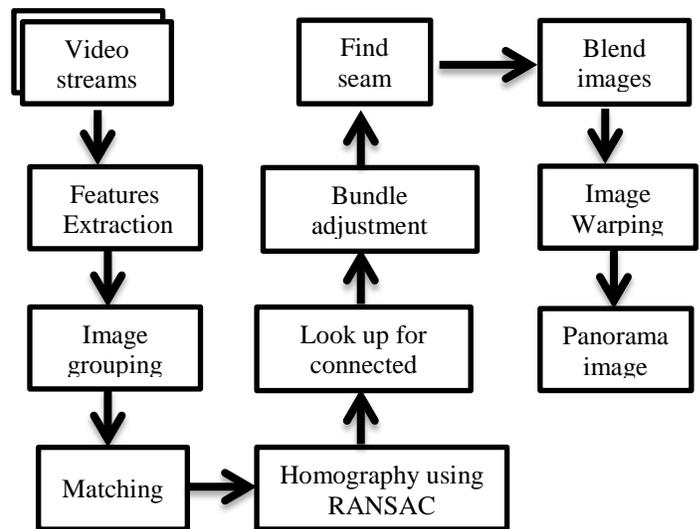


Fig. 4. Real-time panorama stitching algorithm



Fig. 5. Panorama created using three simultaneous frames

On the other hand, the Object detection & tracking module is implemented on the main board (central unit). This board serves to remotely command the overall system, and it is also used to:

- coordinate between all sub-units via a communication middleware,
- stabilize moving videos,
- perform some computer vision tasks using a set of vision algorithms such as HOG + Linear SVM detector [15] for pedestrian detection, CAMSHIFT for scale invariant objects recognition in high altitudes [6], etc.,
- track targets by centering them in the middle of the frame using a couple of pan/tilt servomotors able to perform both vertical and horizontal rotations,
- fuse all the retrieved data before it is sent to the ground station,

- take decisions (send alerts, actuate a command, ...) when a special event (listed in the event knowledge base) is encountered,
- aggregate all the generated outputs destined to the ground station using a **Data Fusing Agent**,
- Interact with the ground station via a wireless secure channel.

**Stabilization Agent:**

Once video streams have been preprocessed, in order to help performing a robust target tracking, a video **stabilization agent** is first used to reduce the effects of shakiness induced by wind perturbations, forward movements ....

The hereby **stabilization agent** functioning can be resumed into five steps:

- A look-up for a Euclidean transformation that occurred between the precedent and current frames is conducted using optical flow [20] on all frames. This transformation is only based on three parameters: dx (horizontal), dy (vertical), da (angle). Figure 6 show an example of shakiness in two consecutives frames histograms based on dx and dy.

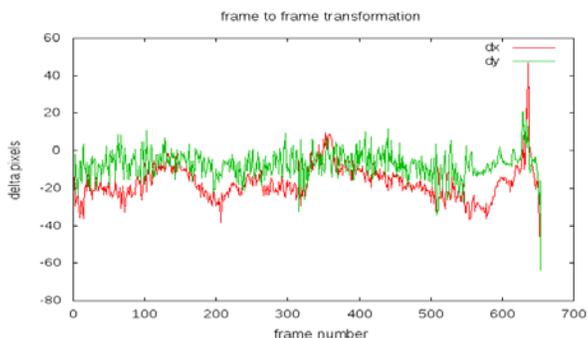


Fig. 6. Example of shakiness on dx and dy from a moving camera

- Stock the consecutives transformations to trace the “trajectories” for x, y, angle, at each frame.
- Smooth out the trajectory using a sliding average window, and defining the window radius as equal to the number of frames used for smoothing. Figure 7 shows an example of smoothing based on dx.

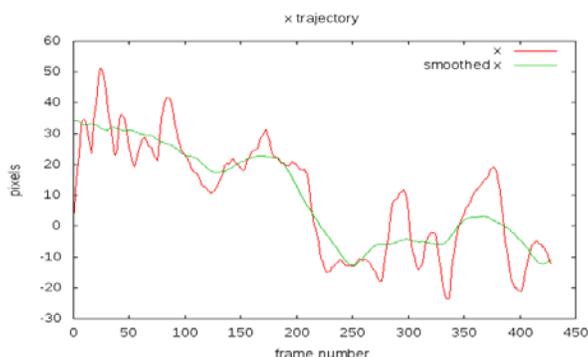


Fig. 7. Example of smoothing based on dx alone

- Create a new transformation defined as:  
 $N\_transform = transformation + (S\_trajectory - trajectory)$ .
- Apply the new transformation to the video (fig 8).

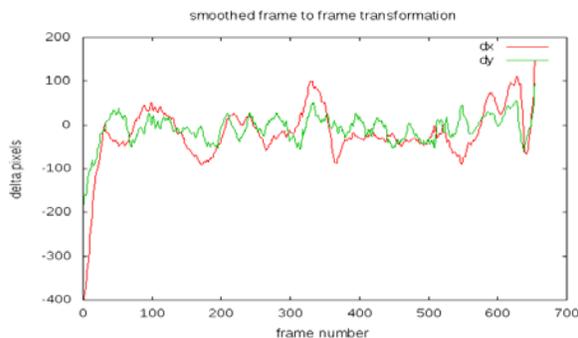


Fig. 8. Final transformation applied to the video

**Target initialization agent:**

In this section, a particular type of targets was chosen to work on: pedestrian targets. In order to detect this class of targets, an implementation based on HOG + Linear SVM [15] model had been used. Once the stream video is handed, the process begins by initializing the Histogram of Oriented Gradients descriptor.

Then, the Support Vector Machine is set to be pre-trained pedestrian detector. From then, once the pedestrian detector is fully loaded, it is looped on the stream frames.

**Image tracking agent:**

Figure 9 shows the proposed tracking agent mode of functioning.

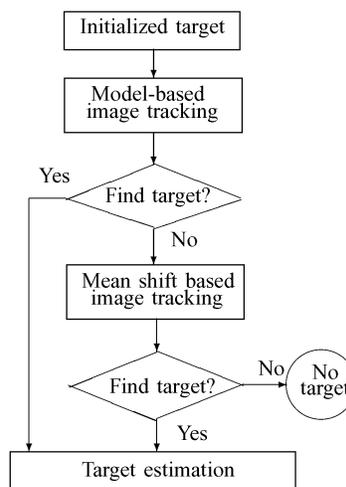


Fig. 9. Tracking image agent

- Once the target is initialized, in the model-based image tracking, a Kalman filtering technique tries to predict the position and velocity of the target in the subsequent frames and then perform data association based on an updated likelihood function [19].

- Once the target gets out of range from the model-based tracker, a changing mechanism check whether the target is still in the image.
- If yes, the mean-shift tracker will be activated. The loss of the target can be attributed to the poor match of features due to noise, distortion, or occlusion in the image. An alternative reason may be the maneuvering motion of the target, and the target is out of the image.
- If the target is still in the image, continuously adaptive mean-shift (CAMSHIFT) algorithm [6] is used to efficiently obtain the optimal location of the target in the search window.

#### Servo following agent:

In this section is described a target-following system based on a pan/tilt servomechanism. This servomechanism is used to control the orientation of the camera to keep the target in an optimal location in the image plane. The mechanism is operating via i2c and can be fully automated to track a designed target of commanded manually via keyboard.

#### Data fusing agent:

This agent sole purpose is to aggregate all the data gathered from all the devices and boards, in order to send them to the ground station after to formatting as they were requested.

#### I. User level:

From this interface, any user can access to the vision system web interface to perform the previously cited functionalities.

## VI. CONCLUSION

In this paper, are presented the architecture and functionalities of a remote control wireless vision system embedded onboard a hex-rotorcraft and formed using a distributed multi-agent system architecture. This architecture is validated by an implementation realized onboard a set composed of two raspberry pi, 2 cameras modules, 1 go pro hero + black and one USB camera. The overall functionalities provided by the system can be summarized by the following:

- At the reactive level, all the images and videos gathered from the sensors are preprocessed. This step can be summarized in two big steps: at first, the FPS of all sensors are increased, followed by a unification process that handles all the gathered streams from the heterogeneous cameras in order to uniform them into classes (a VideoClass for video streams and an ImageClass for images) with standard spatial/temporal resolutions.

- On the deliberative level, two units are implemented in order to offer either a live panoramic view using incoming data feeds from heterogeneous cameras, or a pedestrian detection and tracking system using a servomechanism system.

As a further work, the system ought to be improved by:

- Readapting the stitching algorithm so that it can withstand moving cameras,

- Supplying the decision making agent with rules so that respond autonomously to a set of unexpected events,
- implementing a backup wireless communication (radio) adapted to long range data exchange,
- Making the stabilization algorithm more robust and less CPU consuming,
- Extending the target following mechanism so that it can cooperate with UAV following control mechanism.

#### REFERENCES

- [1] M. J. Adriaans, R. C. Cribbs, B. T. Ingram, and B. P. Gupta, "A160 Hummingbird unmanned air vehicle development program," in Proc. AHS Int. Specialists' Meeting—Unmanned Rotorcraft: Des., Control Test., Chandler, AZ, 2007.
- [2] O. Amidi, T. Kanade, and R. Miller, "Vision-based autonomous helicopter research at Carnegie Mellon Robotics Institute 1991–1997," in Proc. Amer. Helicopter Soc. Int. Conf., Gifu, Japan, 1998, pp. 1–12.
- [3] N. Guenard, T. Hamel, and R. Mahony, "A practical visual servo control for an unmanned aerial vehicle," IEEE Trans. Robot., vol. 24, no. 2, pp. 331–340, Apr. 2008.
- [4] E. N. Johnson, A. J. Calise, Y. Watanabe, J. Ha, and J. C. Neidhoefer, "Real-time vision-based relative aircraft navigation," J. Aerosp. Comput., Inf. Commun., vol. 4, no. 4, pp. 707–738, 2007.
- [5] <http://docs.emlid.com/navio/Downloads/Real-time-Linux-RPi2/>
- [6] G. R. Bradski and S. Clara, "Computer vision face tracking for use in a perceptual user interface," Intel Technology Journal Q2 '98, pp. 1–15, 1998.
- [7] "RPI Camera board - Raspberry-Pi - Raspberry Pi Kamera-Board, 5MP | Farnell Deutschland". *de.farnell.com*. Retrieved 9 June 2013.
- [8] HERO3+ Black Edition, gopro.com
- [9] Brown and D. Lowe. Automatic Panoramic Image Stitching using Invariant Features. International Journal of Computer Vision, 74(1), pages 59-73, 2007.
- [10] Lowe, David G. (1999). "Object recognition from local scale-invariant features" (PDF). Proceedings of the International Conference on Computer Vision pp. 1150–1157. doi:10.1109/ICCV.1999.790410
- [11] U.S. Patent 6,711,293, "Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image", David Lowe's patent for the SIFT algorithm, March 23, 2004
- [12] "Deformable Medical Image Registration: Setting the State of the Art with Discrete Methods" Annual Review of Biomedical Engineering, Vol. 12, 2011, pp. 219-244
- [13] Dorin Comaniciu and Visvanathan Ramesh. Mean shift and optimal prediction for efficient object tracking. In Tracking, International Conference on Image Processing, pages 70–73, 2000.
- [14] Unmanned Rotorcraft Systems Par Guowei Cai, Ben M. Chen, Tong Heng Lee
- [15] Histograms of Oriented Gradients for Human Detection, Navneet Dalal and Bill Triggs,
- [16] <http://makezine.com/2014/02/25/how-to-choose-the-right-platform-raspberry-pi-or-beaglebone-black/>
- [17] K. Boukhdir, A. Boualam, S. Tallal, H. Medromi, S. Benhadou "Conception, Design and Implementation of Secured UAV Combining Multi-Agent Systems and Ubiquitous Lightweight IDPS (Intrusion Detection and Prevention System)" IREA Vol 3, No 1 (2015)
- [18] M. A. Fischler, R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. of the ACM, Vol 24, pp 381-395, 1981.
- [19] Feng Lin, Kai-Yew, A Robust Real-Time Embedded Vision System on an Unmanned Rotorcraft for Ground Target Following. IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 59, NO. 2, FEBRUARY 2012
- [20] Gibson, J.J. (1950). The Perception of the Visual World. Houghton Mifflin.