



HAL
open science

Vision Based Target Tracking Using An Unmanned Aerial Vehicle

El Houssein Chouaib Harik, François Guérin, Frédéric Guinand, Jean-François Brethé, Hervé Pelvillain, Adel Zentout

► **To cite this version:**

El Houssein Chouaib Harik, François Guérin, Frédéric Guinand, Jean-François Brethé, Hervé Pelvillain, et al.. Vision Based Target Tracking Using An Unmanned Aerial Vehicle. ARSO 2015. 2015 IEEE International Workshop on Advanced Robotics and its Social Impacts, Jul 2015, Lyon, France. 10.1109/ARSO.2015.7428194 . hal-01170117

HAL Id: hal-01170117

<https://hal.science/hal-01170117>

Submitted on 1 Jul 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

Vision Based Target Tracking Using An Unmanned Aerial Vehicle

El Houssein Chouaib Harik¹, *Student Member, IEEE*, François Guérin², Frédéric Guinand¹,
Jean-François Brethé², *Member, IEEE*, Hervé Pelvillain³, Adel Zentout⁴

Abstract—We present in this paper a backstepping controller for vision based target tracking with an Unmanned Aerial Vehicle. A down facing camera is used with a pose estimation algorithm to extract the position of the target (an Unmanned Ground Vehicle). The output is then fed into the developed controller to generate the necessary movements (pitch and roll) of the Unmanned Aerial Vehicle in order to keep the target in the coverage view of the camera (following it constantly). The developed scheme is used to help the Unmanned Ground Vehicle to navigate among obstacles, and the overall system is designed in order to help human operator to supervise the Aerial and Ground vehicles for area inspection or object transportation in industrial areas (when using multiple Unmanned Ground Vehicles).

I. INTRODUCTION

The usage of Unmanned Aerial Vehicles (or UAVs) has known an exponential growth in the last past years. Primarily introduced in military activities, UAVs, and most precisely rotorcrafts are now used in different civilian applications, from agriculture [1], 3D mapping [2], to search and rescue [3], and even cinematography, where the first drone festival has been held in New York city at *The Directors Guild of America Theater* [4].

When deployed with Unmanned Ground Vehicles (UGVs), UAVs bring complimentary skills making the team outperforms the classical single-robot systems in many aspects (speed, effectiveness, coverage, etc...). Likewise, using Air Ground Cooperation (AGC) in mobile robotics allows to obtain a complete system that include global perception and high movement velocities brought by the UAVs, and powerful computational capabilities and payload brought by the UGVs. Thus AGC opens a new wide range of applications using mobile robotics that was not an easy (or even impossible) task to be performed using a single type of robots or even a group of homogeneous robots.

Similarly as in [5], the main idea in this work consists in using the UAV as a remote eye, where the authors use the UAV as a mobile sensor that flies ahead to provide geo-referenced 3D geometry in order to enable the UGV to navigate safely avoiding obstacles and negative obstacles (holes and ditches). Another cooperative navigation scenario can be found in [6], [7], [8], where an aerial robot equipped

with Lidar collects 3D information of a specific area (Desert, Alpine, or Meadow areas), these information are processed to produce a 3D map (vegetations and traversability). An UGV navigates through that area combining the post-processed map, and its own 3D Lidar information to generate an optimal path avoiding vegetation and other static obstacles. Similar work can be found in [9] and [1].

In previous related works, a first flight of the UAV over the inspected area is always performed before the beginning of the mission. A traversability map is then processed to provide a trajectory to the UGV. A first contribution of our work consists in providing a real-time navigation scheme: the UAV flies over the area to provide a global coverage, and to assist a UGV in real-time to navigate safely avoiding obstacles. The UAV follows continuously and autonomously the UGV using the developed visual servoing algorithm and controller. The second contribution of our work is to use an off-the-shelf technology (commercial UAVs and open source libraries) to propose an efficient control law in order to develop a GPS denied tracking system. The result of this work is used to provide the UGV with an aerial coverage view of the navigation area to facilitate its navigation through obstacles. The UGV goes through waypoints provided by a human operator. For numerous reasons, monitoring, surveillance and inspection of industrial sites belonging to Seveso category (highly critical sites) cannot be performed without a constant and careful attention of human experts. Related works and the choice of the controller will be discussed further in this paper.

This work takes part of a project [10], [11], [12] developed by two research laboratories (LITIS, GREAH) at the University of Le Havre (France), and funded by Le Havre town council (CODAH).

This paper is organized as follows: We give an overview of the used vision based tracking techniques in section II. We detail in section III the pose estimation of the UGV. In section IV we explain our choice of the backstepping controller, and present the different steps in order to generate the necessary movements to follow the UGV. We present in section V simulation and experimental results. We conclude in section VI the present work and give future perspectives.

II. VISION BASED UGV TRACKING

After acquiring the video flow (Figure 1) from the UAV's down facing camera (we suppose initially that the UGV is on the coverage view), we process this flow in order to extract the UGV's pose. For the UGV visual tracking we have used two different methods: color detection, and augmented

¹LITIS, University of Le Havre. 25 Rue Philippe Lebon, 76600 Le Havre Cedex (France) - (el-houssein-chouaib.harik, frederic.guinand)@univ-lehavre.fr

²GREAH, University of Le Havre. 75, Rue Bellot, 76058 Le Havre Cedex (France) - (francois.guerin, jean-francois.brethe)@univ-lehavre.fr

³IUT GEII, University of Le Havre. Rue Boris Vian, 76610 Le Havre Cedex (France) - herve.pelvillain@univ-lehavre.fr

⁴LHS, University of Mentouri. Route Ain El Bey, Constantine 25017 (Algeria) - zentout_adel@umc.edu.dz

reality marker detection. We will explain in this section the developed algorithms for each method and discuss their advantages and drawbacks.

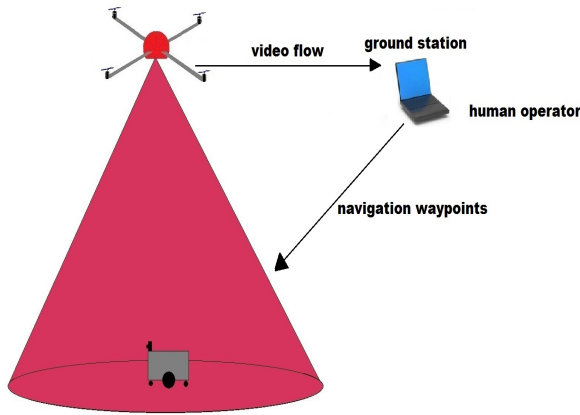


Fig. 1. Overall architecture

A. Color tracking

Color detection is the first technique used for target tracking. It consists generally in going through the image pixels until a match is found with the predefined target color (with a tolerance). The authors in [13] used a UAV to estimate the position of a Mecanum Wheel Robot (MWR) and track it. They used LabVIEW vision development toolkit for target tracking (a colored ball on the MWR). The commands are then sent to the UAV in order to change its pitch and roll angles according to the MWR movements. A similar localization approach can be found in [14] with the exception of adding a digital compass to get the orientation of the ground target. Similar works can be found in [15], [16].

Similarly to [13] and [14], we installed a red colored marker (R_c) on a flat surface on the center of the UGV (Figure 2). The first step consist in acquiring (R_c) color which is a value that contains the basic RGB colors (red, green, blue). Now that we stored the color of the marker R_c , we will look for a match in the image (we have to chose a unique color that does not belong to the environment). We go through each pixel by extracting each time its RGB components and compare it to the stored value. Once a match is found (with an allowable tolerance), we save the position of that pixel. A barycenter of the matching pixels is computed, and assumed to be the UGV's location in the image.

Figure 2 shows the results of the color tracking algorithm. The center (R_c) of the UGV is a red marker. We added the green marker located at the head of the UGV in order to estimate the orientation of the UGV in the image to the waypoint (blue circle) provided by a human operator, for the purpose to navigate through obstacles as mentioned previously.

The advantages of the color tracking method is the easiness of implementation, it doesn't require huge processing

time, it is relatively fast, and give good results in good lightning conditions.

The drawbacks of the method is the sensitivity to light changing. Going from a light spot to a dark one changes the detected color thus loosing the colored target, that is why we added the tolerance factor. Another drawback is if the environment has an object with the same tracked color, this will result in a false detection, thus the pose estimation of the UGV will be false.

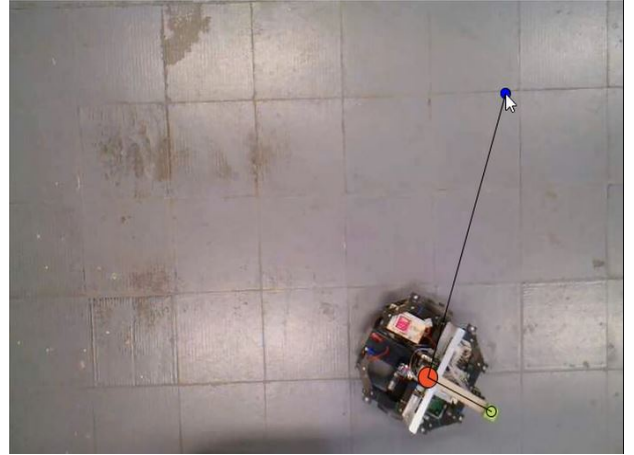


Fig. 2. Color tracking for UGV pose estimation

B. Augmented reality marker tracking

To overcome the limitations of the color tracking method, some works got inspired by an augmented reality application [17], where the authors presented a Computer Supported Collaborative Work (CSCW): A conference system where remote collaborators are represented on virtual monitors which can be freely positioned in a user space [17]. The user is wearing a Head Mounted Display (HMD), and can see the collaborators faces overlayed on a specific marker. The proposed method is publicly available and known as the ARToolKit. The library has been used in [18], where the authors included as a vision feedback for pose estimation in the control loop of several UGVs. We can find in [19] another interesting feature-based (black and white pattern) detection for multiple robots localization. The presented method allow the tracking of several features at a time with a good precision using low cost cameras, and uses the same kind of visual marker as in ARToolKit. The proposed method has been made publicly available as a library called ArUco [20]. Another AR marker tracking can be found in [21]. The authors used a down facing camera mounted on an aerial vehicle (quadrotor) in order to hover it on a ground feature (augmented reality marker). They fused the information gathered from the vision combined with internal sensors (Inertial Measurement Unit (IMU)) in order to generate the necessary movements to stay on the top of the ground feature.

We have used in our work the library presented in [17]. Figures 3 and 4 represents two examples of AR markers [22] that will be used later for the UGV's pose estimation:

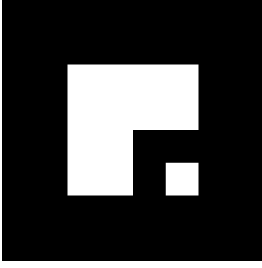


Fig. 3. Marker 1

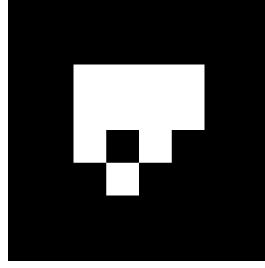


Fig. 4. Marker 2

As in color detection, we have to go through each received image to check if the marker is present. Algorithm 1 illustrates the steps in order to estimate the 2D UGV's center (R_c) coordinates.

Algorithm 1: Augmented reality marker tracking

Input: Image (video flow), AR marker (the tracked coded marker)

Output: X_{R_c}, Y_{R_c} (2D pose of the tracked marker)

begin

 Define AR marker (marker 1) ;

for Image **do**

if Image has marker 1 **then**

 Extract marker 1 corners (0, 1, 2, 3);

 Find the center of the marker 1;;

$dx1 = x.corner(2) - x.corner(0)$;

$dy1 = y.corner(2) - y.corner(0)$;

$m1 = dy1 / dx1$;

$c1 = y.corner(0) - m1 * x.corner(0)$;

$dx2 = x.corner(3) - x.corner(1)$;

$dy2 = y.corner(3) - y.corner(1)$;

$m2 = dy2 / dx2$;

$c2 = y.corner(1) - m2 * x.corner(1)$;

$X_{R_c} = (c2 - c1) / (m1 - m2)$;

$Y_{R_c} = m1 * X_{R_c} + c1$;

 draw a red circle on (X_{R_c}, Y_{R_c}) ;

else

 Continue searching;

Figure 5 shows the results of the augmented reality marker tracking algorithm. We have used marker 1 (Figure 3) as an example:

The main advantage of this method is that it is more robust to lighting conditions compared to color tracking algorithms. Another advantage is that we can use different markers with different IDs and the environment most likely doesn't include an object with the same characteristics as an AR marker, which will limit the tracking errors. The drawbacks of this method is that it is time consuming (edges detection, pattern match searching, etc...). Fortunately a lot of progress has been made in the literature to improve the used algorithms, and we are so far satisfied with the results of the tracking (results will be presented in section V).

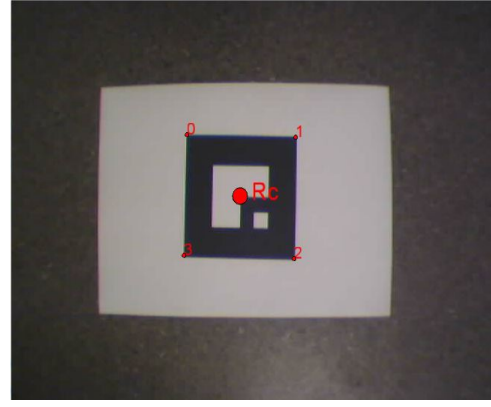


Fig. 5. Augmented reality marker tracking for UGV pose estimation

III. UGV POSE ESTIMATION

We consider X_D and Y_D the axes of the image taken from the drone (Figure 6). In order to locate the UGV, we can use one of the presented tracking algorithms to extract the UGV's position (R_c). The position (in pixels) of R_c along X_D and Y_D axes is: Y_{R_c} and X_{R_c} . Y_0 and X_0 are the center (0, 0) of the image (we assume that the center of the image is the center of the drone).

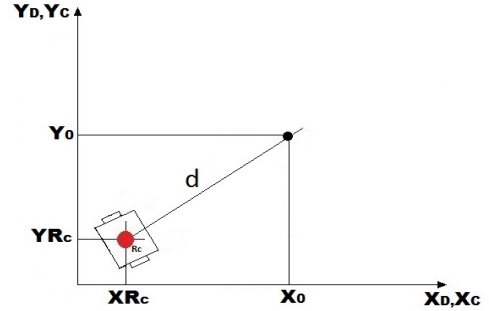


Fig. 6. The UGV in the image plan

The distance d that separates the UGV and the center of the UAV can be written as follows:

$$d = \sqrt{(X_0 - X_{R_c})^2 + (Y_0 - Y_{R_c})^2} \quad (1)$$

Where $X_0, Y_0, X_{R_c}, Y_{R_c}$ represent the coordinates of the marker R_c and the center of the aerial vehicle in its (or camera) frame previously supposed to be the same. The pose estimation of the UGV from the visual input is directly defined in the camera space, thus no need for transforming the coordinates in a world frame (using Pinhole model for example).

IV. THE UAV CONTROLLER'S DESIGN

To hover the UAV over the UGV during its navigation, we need to control the UAV movements along X_D and Y_D axis (Figure 6) since the UGV moves on a 2D plan (assumed to be flat). We need to develop a navigation controller that takes as input the location of the UGV (R_c), and generates

the necessary pitch and roll angles to keep the UGV at the center of the image plan (visual servoing).

We consider four inputs ϕ_d, θ_d, ψ_d and z_d . They represent respectively the desired roll, pitch, yaw angles, and the desired altitude. Note that the UAV attitude (roll, pitch, yaw, altitude) is controlled by the internal autopilot. We suppose that the UAV flies at a fixed altitude, and a fixed yaw: $z_d = z$ and $\psi_d = 0$. z is chosen to have a sufficient coverage of the inspected area ($z_{min} < z$), and to keep the track of the UGV ($z < z_{max}$). z_{min} and z_{max} can be defined experimentally.

To design the UAV controller, we used the dynamic model of a quadcopter [23]. Starting from this dynamic model, the movements along X_D and Y_D axis can be described as follows:

$$\begin{aligned} \ddot{x}_D &= (\cos\phi_d \sin\theta_d \cos\psi_d + \sin\phi_d \sin\psi_d) \frac{1}{m} U_1 \\ \ddot{y}_D &= (\cos\phi_d \sin\theta_d \sin\psi_d - \sin\phi_d \cos\psi_d) \frac{1}{m} U_1 \end{aligned} \quad (2)$$

\ddot{x}_D and \ddot{y}_D correspond to the acceleration along X_D and Y_D axis respectively. (2) can be as:

$$\begin{aligned} \ddot{x} &= u_{X_D} \frac{1}{m} U_1 \\ \ddot{y} &= u_{Y_D} \frac{1}{m} U_1 \end{aligned} \quad (3)$$

With:

$$\begin{aligned} u_{X_D} &= (\cos\phi_d \sin\theta_d \cos\psi_d + \sin\phi_d \sin\psi_d) \\ u_{Y_D} &= (\cos\phi_d \sin\theta_d \sin\psi_d - \sin\phi_d \cos\psi_d) \end{aligned} \quad (4)$$

where u_{X_D}, u_{Y_D} represent the orientations of the total thrust (U_1) responsible for linear motions of the quadcopter along X_D and Y_D axis. m is the UAV's weight.

The choice of the backstepping controller was based on the study proposed in [24], where the authors analyzed the performance of a quaternion-based feedback controller, a backstepping controller and a sliding mode controller, showing the robust behavior of the backstepping controller for both waypoint tracking and attitude stabilization. In our case the waypoint is not a static entity, but a moving target that we estimated its position in section III.

Let us describe now the design of the backstepping controller:

1) *Positioning errors along the UAV X axis:* We consider the following system:

$$\begin{cases} x_1 = x \\ x_2 = \dot{x} \end{cases} \implies \begin{cases} \dot{x}_1 = \dot{x} = x_2 \\ \dot{x}_2 = \ddot{x} = \frac{1}{m} u_{X_D} U_1 \end{cases} \quad (5)$$

We define the first error, where x_{1d} is the desired position (the UGV coordinate along X axis: X_{R_c}):

$$\epsilon_1 = x_{1d} - x_1 \quad (6)$$

We consider the Lyapunov candidate function:

$$V_1(\epsilon_1) = \frac{1}{2} \epsilon_1^2 \quad (7)$$

Its time derivative is given by:

$$\dot{V}_1(\epsilon_1) = \epsilon_1 \dot{\epsilon}_1 = \epsilon_1 (\dot{x}_{1d} - \dot{x}_2) \quad (8)$$

Lyapunov condition to consider the system as asymptotically stable, and convergent:

$V_1(\epsilon_1) > 0 \forall x_1$, and ($\dot{V}_1 < 0 \forall x_1$). This leads to:

$$(\dot{x}_{1d} - \dot{x}_2) = -K_1 \epsilon_1 \quad (9)$$

$$\dot{V}_1 = \epsilon_1 (\dot{x}_{1d} - \dot{x}_2) = -K_1 \epsilon_1^2 \quad (10)$$

Where K_1 is a positive constant. To stabilize ϵ_1 we introduce the virtual control input ($\alpha_1 = x_2$).

$$\dot{x}_1 = \alpha_1 = K_1 \epsilon_1 + \dot{x}_{1d} = -K_1 x_1 + K_1 x_{1d} + \dot{x}_{1d} \quad (11)$$

We notice that if α_1 is equal to $K_1 \epsilon_1 + \dot{x}_{1d}$ then x_1 converge to x_{1d} . In the meanwhile, x_2 is not equal to α_1 , thus a new error is considered between x_2 and its desired value.

We consider the following errors:

$$\epsilon_1 = x_{1d} - x_1 \quad \epsilon_2 = \alpha_1 - x_2 \quad (12)$$

Their time derivatives are:

$$\dot{\epsilon}_1 = \dot{x}_{1d} - \dot{x}_1 = \dot{x}_{1d} + \epsilon_2 - \alpha_1 \quad (13)$$

$$\dot{\epsilon}_2 = \dot{\alpha}_1 - \dot{x}_2 = K_1 \dot{\epsilon}_1 + \ddot{x}_{1d} + u_x \frac{1}{m} U_1 \quad (14)$$

We define the second Lyapunov candidate function taking into account the two errors and their time derivatives:

$$V_2 = \frac{1}{2} (\epsilon_1^2 + \epsilon_2^2), \quad \dot{V}_2 = \epsilon_1 \dot{\epsilon}_1 + \epsilon_2 \dot{\epsilon}_2 \quad (15)$$

$$\dot{V}_2 = \epsilon_1 (x_2 - (K_1 \epsilon_1 + \dot{x}_{1d}) + \epsilon_2) + \epsilon_2 (K_1 \dot{\epsilon}_1 + \frac{1}{m} u_{X_D} U_1) \quad (16)$$

$$\dot{V}_2 = -K_1 \epsilon_1^2 + \epsilon_2 (\epsilon_1 + K_1 \dot{\epsilon}_1 + \ddot{x}_{1d} + \frac{1}{m} u_{X_D} U_1) \quad (17)$$

Similarly to (10) we can then get:

$$\epsilon_1 + K_1 \dot{\epsilon}_1 + \ddot{x}_{1d} + \frac{1}{m} u_{X_D} U_1 = -K_2 \epsilon_2 \quad (18)$$

u_{X_D} can be written:

$$u_{X_D} = \frac{m}{U_1} (-\epsilon_1 - K_1 \dot{\epsilon}_1 - \ddot{x}_{1d} - K_2 \epsilon_2) \quad U_1 \neq 0 \quad (19)$$

2) *Positioning errors along the aerial vehicle Y axis:* Following the same steps as in the positioning errors along X axis, u_{Y_D} can be written:

$$u_{Y_D} = \frac{m}{U_1} (-\epsilon_3 - K_3 \dot{\epsilon}_3 - \ddot{y}_{1d} - K_4 \epsilon_4) \quad U_1 \neq 0 \quad (20)$$

The UGV's velocity and acceleration can be estimated thanks to a first order approximation.

3) *Desired roll and pitch to follow the UGV*: To find the desired roll and pitch angles needed to track the UGV, we combine (19) and (20) in (3) taking a fixed yaw angle $\psi = 0$. The desired angles are then:

$$\begin{aligned}\phi_d &= \text{asin}(-u_{Y_D}) \\ \theta_d &= \text{asin}\left(\frac{u_{X_D}}{\cos\phi_d}\right)\end{aligned}\quad (21)$$

We note that the steady state error will be null (when the UGV stops) since it is a position control.

V. RESULTS

We present in this section some simulation results concerning the UAV following the UGV in its waypoints tracking. Unfortunately we don't yet have access to the UAV's position measurements in the inertial frame (for example VICON MX system). That is why quantitative results are presented for the simulation trials only.

A. Simulation result

We used for simulation the parameters of an open-source quadcopter developed by [25]. They have been obtained thanks to experimentations fully described in [26].

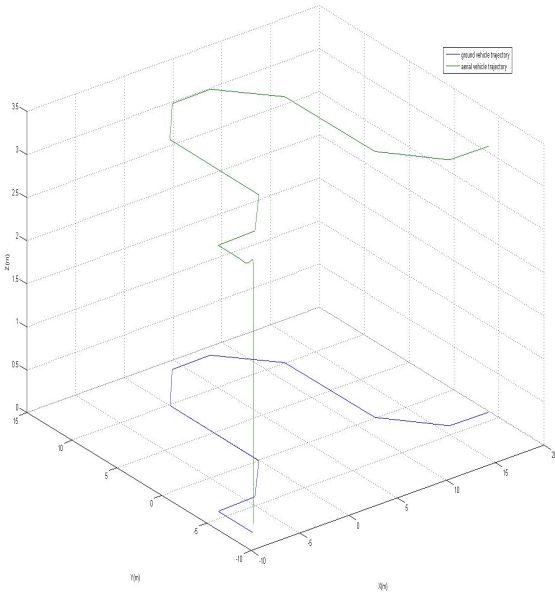


Fig. 7. Inspection scheme simulation

Figure 7 shows that the UAV takes-off from an initial position (-7,-7) and flies to hover at 3m altitude. We suppose that at this height the UGV with an initial position (-8, -8) is in the coverage view of the aerial vehicle. We can see that the UGV moves followed by the UAV. Figure 8 shows that the UAV changes its pitch and roll angles according to the motion of the UGV.

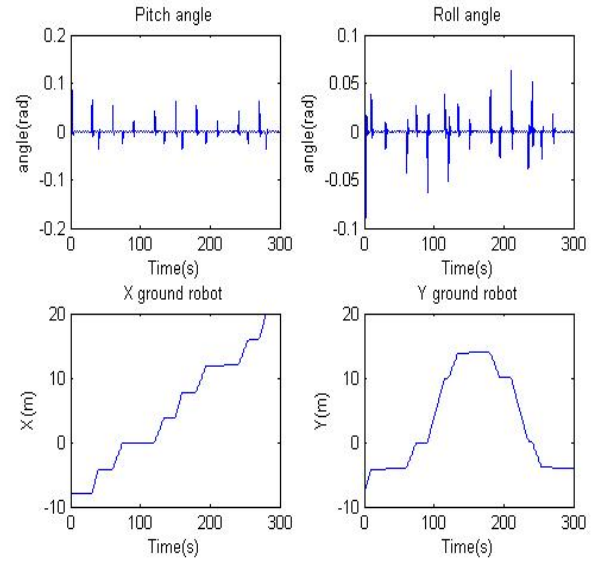


Fig. 8. UAV angles and UGV movement

B. Experimental result

We carried out several experimentations first using color tracking, then the Augmented Reality (AR) marker tracking. For the UAV we used an AR Drone 2.0 of Parrot [27]. The video flow of the UAV is sent continuously to the ground station, to be processed in our developed User Interface (UI) and send the motion controls back to the UAV through wifi link. We developed the UI with processing [28], and used several publicly available libraries that can be added directly to the processing sketch. The developed controller can be directly implemented on an UAV flight controllers. In our case, such implementation is more difficult because of the computation capabilities of the AR Drone. Obstacles avoidance of the UGV is done through on-the-go waypoints provided by the human operator, they can be selected by clicking on the UI screen and the UGV navigates to that point, further details can be found in [12].

1) *color tracking*: First experiments were carried out using colored markers on the UGV as shown in Figure 9. The video of this experimentation can be found in [29].

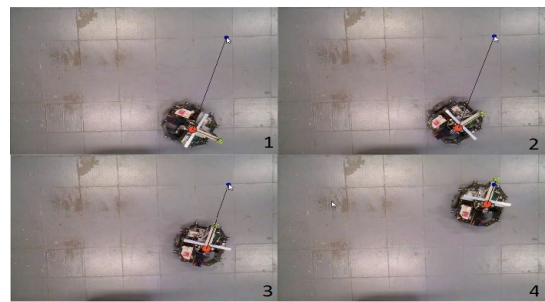


Fig. 9. Using the colored markers on the UGV

2) *AR marker tracking*: Due to sensibility of color tracking to light condition, we used the AR marker instead (figure 10)

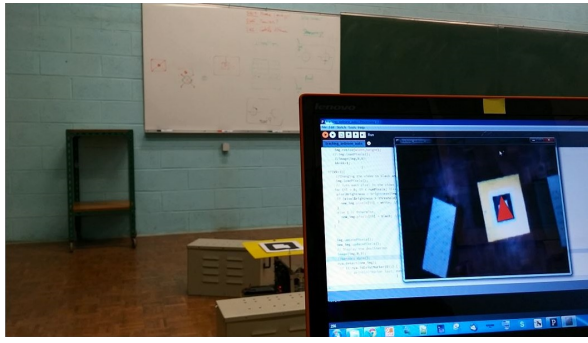


Fig. 10. Using the AR marker on the UGV

We draw a virtual triangle on the marker to illustrate the orientation and the position of the UGV in the image. This method gave much better results than the color tracking one, the video of an experimentation can be found in [30].

VI. CONCLUSION AND PERSPECTIVES

We presented in this paper a visual servoing technique using a backstepping controller to allow an UAV to follow autonomously an UGV. We have used two different tracking methods to estimate the position of the UGV and gave the advantages and drawbacks of each method. Simulations and experiments had been carried out to show the efficiency of the proposed controller and tracking algorithms giving an efficient GPS denied tracking system. A future work is to implement the developed controller and target tracking directly in an UAV autopilot.

VII. ACKNOWLEDGMENTS

The authors would like to thank Le Havre Town Council CODAH for their support under research grants. They would like as well to thank Upper Normandie Region through their support to PCMAI project.

REFERENCES

- [1] Lars Kuhnert., Klaus Müller., Markus Ax., and Klaus-Dieter Kuhnert. "object localization on agricultural areas using an autonomous team of cooperation ground and air robots". In *International Conference of Agricultural Engineering, CIGR-AGENG2012, Valencia, Spain*, July 8-12 2012.
- [2] Nathan Michael, Shaojie Shen, Kartik Mohta, Yash Mulgaonkar, Vijay Kumar, Keiji Nagatani, Yoshito Okada, Seiga Kiribayashi, Kazuki Otake, Kazuya Yoshida, et al. Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *Journal of Field Robotics*, 29(5):832–841, 2012.
- [3] Cai Luo, Andre Possani Espinosa, Danu Pranantha, and Alessandro De Gloria. Multi-robot search and rescue team. In *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pages 296–301. IEEE, 2011.
- [4] NEW YORK CITY DRONE FILM FESTIVAL, 2015. URL: <http://www.nycdronefilmfestival.com/>.
- [5] Anthony Stentz, Alonzo Kelly, Herman Herman, Peter Rander, Omead Amidi, and Robert Mandelbaum. Integrated air/ground vehicle system for semi-autonomous off-road navigation. *Robotics Institute*, page 18, 2002.
- [6] Nicolas Vandapel, Raghavendra Rao Donamukkala, and Martial Hebert. Unmanned ground vehicle navigation using aerial lidar data. *The International Journal of Robotics Research*, 25(1):31–51, 2006.
- [7] Nicolas Vandapel, Raghavendra R Donamukkala, and Martial Hebert. Quality assessment of traversability maps from aerial lidar data for an unmanned ground vehicle. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 305–310. IEEE, 2003.
- [8] Nicolas Vandapel, Raghavendra Donamukkala, and Martial Hebert. Experimental results in using aerial lidar data for mobile robot navigation. In *Field and Service Robotics*, pages 103–112. Springer, 2006.
- [9] Boris Sofman, J Andrew Bagnell, Anthony Stentz, and Nicolas Vandapel. Terrain classification from aerial data to support ground vehicle navigation. Carnegie Mellon University, 2006.
- [10] Harik El Houssein Chouaib, Guérin François, Guinand Frédéric, Brethé Jean-François, and Pelvillain Hervé. Uav-ugv cooperation for objects transportation in an industrial area. In *IEEE International Conference on Industrial Technology (ICIT)*, pages 547–552. IEEE, 2015.
- [11] Harik El Houssein Chouaib, Guérin François, Guinand Frédéric, Brethé Jean-François, and Pelvillain Hervé. Decentralized control architecture for uav-ugv collaboration. In *2nd AETOS international conference on "Research challenges for future RPAS/UAV systems"*, 2014.
- [12] Harik El Houssein Chouaib, Guérin François, Guinand Frédéric, Brethé Jean-François, and Pelvillain Hervé. A decentralized interactive architecture for aerial and ground mobile robots cooperation. In *International Conference on Control, Automation and Robotics (ICCAR)*, pages 37–43. IEEE, 2015.
- [13] Jae-Young Choi and Sung-Gaun Kim. *Collaborative Tracking Control of UAV-UGV*. World Academy of Science, Engineering and Technology, 71, 2012.
- [14] Luiz Chaimowicz, Ben Grocholsky, James F Keller, Vijay Kumar, and Camillo J Taylor. "Experiments in multirobot air-ground coordination". In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 4, pages 4053–4058. IEEE, 2004.
- [15] René Vidal, Omid Shakernia, H Jin Kim, David Hyunchul Shim, and Shankar Sastry. Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation. *Robotics and Automation, IEEE Transactions on*, 18(5):662–669, 2002.
- [16] H Jin Kim, Rene Vidal, David H Shim, Omid Shakernia, and Shankar Sastry. A hierarchical approach to probabilistic pursuit-evasion games with unmanned ground and aerial vehicles. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 1, pages 634–639. IEEE, 2001.
- [17] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on*, pages 85–94. IEEE, 1999.
- [18] Mark Fiala. Vision guided control of multiple robots. In *Computer and Robot Vision, 2004. Proceedings. First Canadian Conference on*, pages 241–246. IEEE, 2004.
- [19] Tomáš Krajník, Matías Nitsche, Jan Faigl, Petr Vaněk, Martin Saska, Libor Přeučil, Tom Duckett, and Marta Mejail. A practical multirobot localization system. *Journal of Intelligent & Robotic Systems*, 76(3-4):539–562, 2014.
- [20] ArUco: a minimal library for augmented reality applications based on opencv., 2015. URL: <http://www.uco.es/investiga/grupos/ava/node/26>.
- [21] Matevž Bošnjak, Drago Matko, and Sašo Blažič. Quadcopter hovering using position-estimation information from inertial sensors and a high-delay video system. *Journal of Intelligent & Robotic Systems*, 67(1):43–60, 2012.
- [22] Augmented Reality ARToolkit Patternmaker, 2015. URL: <http://www.cs.utah.edu/gdc/projects/augmentedreality/>.
- [23] Samir Bouabdallah and Roland Siegwart. Full control of a quadrotor. In *Intelligent robots and systems, 2007. IROS 2007. IEEE/RSJ international conference on*, pages 153–158. IEEE, 2007.
- [24] Patrick Adigbli, Christophe Gr, Jean-Baptiste Mouret, and Stephane Doncieux. Nonlinear attitude and position control of a micro quadrotor using sliding mode and backstepping techniques. 2007.
- [25] Lez Concept, 2014. URL: <http://www.lezconcept.fr/>.
- [26] El Houssein Chouaib Harik. "stabilisation par backstepping d'un quadrirotor", mémoire de master recherche de l'université de montpellier 2. 2013.
- [27] Ar Drone 2.0, 2015. URL: <http://ardrone2.parrot.com/>.
- [28] Processing, 2015. URL: <https://processing.org/>.
- [29] Experimentation 1: Using colored marker on the ground vehicle, 2015. URL: https://www.youtube.com/watch?v=D110_hObeww.
- [30] Experimentation 2: Using AR marker on the ground vehicle, 2015. URL: https://www.youtube.com/watch?v=xZ76Z-q_wmg.