

A Vision Based Algorithm for the Guidance of a Glider

John Jairo Martínez, Jose Tiberio Hernández
Department of Computer Science
Universidad de los Andes
Bogotá, Colombia

Carlos F. Rodríguez
Department of Mechanical Engineering
Universidad de los Andes
Bogotá, Colombia

Abstract—In this paper we propose a visual estimation algorithm to aid the guidance system of an unmanned aerial vehicle (UAV). Specifically, we are considering a glider UAV trying to reach a target zone. Given that the UAV is falling fast, the scale of the images captured by the on-board camera grows. The vehicle dynamics and the control algorithm cause fast changes in the region seen by the camera and in the way it projects to the image plane. We propose an algorithm based on projective transformations to estimate the target coordinate during the flight trajectory. The proposed vision algorithm provides accurate position information allowing the control system to guide the UAV to the target zone. Simulations have shown the accuracy of the vision algorithm and its usefulness in the considered application.

Keywords—Visual Tracking; UAV; Visual Guidance; SURF

I. INTRODUCTION

A glider is an unmanned aerial vehicle (UAV) without propulsion, it is launched with given initial conditions, and then moves down due to the gravity acceleration. In our particular application, the glider has two control surfaces, the rudder and the elevator, which allow to change the yaw and pitch angles respectively.

UAVs have the ability to perform dangerous tasks avoiding putting human lives at risks [1]. Unmanned gliders can be used in many civil applications like in zone recognition of dangerous areas, and rescue tasks. In military applications, there are other uses like unmanned aircraft that deliver cargo munitions in battle zones and unpowered rockets or guided bombs [2].

Most gliders are guided by Global Positioning Systems (GPS), and are designed to have very small delivery CEPs (circular error probable). The limiting factor on the effectiveness is often the accuracy of the coordinates provided as target points [3], specially when there are emerging targets. Another technique for glider guidance is the use of a laser in a specific wavelength to designate the target coordinate. This approach has the drawback of requiring the launching aircraft to remain close to the target point, with a line of sight to the target, difficulting escaping maneuvers. Another alternative is to use a second aircraft to designate the target zone.

A. Related work

During last decades, there has been a growing interest in developing computer vision algorithms to help designate

targets and guide gliders for a number of applications. The works presented in [1] and [3] focused their attention in finding the target geo-location. In [3], the authors present an approach where multiple photos of the target zone are taken in a noncollinear large subtended angle. The algorithm uses the GPS coordinates of photos along with the camera calibration information to generate a 3D coordinate estimation of the target. The main drawback of this approach, is that it requires a reconnaissance flight to take multiple photos in a specific fashion, before being able to release the glider. In [1], the authors propose an algorithm to find the north-east-down (NED) geo-location of the target based on the pixel coordinates of the video frame and the use of a range finder. The algorithm assumes that a human specifies the target to track at a ground control station in a pixel coordinate frame. The algorithm tracks the coordinate using a Horn-Schunk optical flow method over a fusion of images taken from a regular and thermal cameras. This method is capable of providing a 3D estimation of the target coordinate in real time.

Other works have focused in vision algorithms for other types of UAVs. In [4] a revision of current algorithms and techniques for UAVs vision algorithms is presented. Image processing, feature tracking and appearance based tracking are proposed as categories for visual tracking. The article presents a complete revision of techniques that could be of interest to the readers. The authors in [5] propose a vision algorithm to estimate the movement of a UAV using optical flow. Their approach is divided in two parts, the first one estimates the location of the UAV using path integration of the optical flow measurement, and the second uses a Kalman filter to continuously estimate and correct the position errors. One of the differences between the approach presented by the authors and ours, is that the simulated conditions in their work don't consider the full six degrees of freedom of the UAV, while the simulations in our approach do, extending the applicability of the algorithm to more complex scenarios.

In the work presented in [6], the authors use an approach based on projective transformations using SIFT feature points. A reference image is previously processed and feature points are stored. Then, when the vehicle is flying, the online images are processed to look for matches with the points in the reference image. If matches are found, a RANSAC algorithm is used to find a projective transformation relating the reference image

and the online captured one. Our approach differs from this one in that we cannot know which is going to be our reference image, and because of the glider's typical release points and flying dynamics is unlikely that the initial image works as reference for the whole flight.

In this work, we attempt to solve a similar but slightly different problem. We propose a vision only estimation algorithm to track in real time a target coordinate during the flight trajectory of a glider. The proposed approach receives the target designation online. The human operator intervenes just once to select a target pixel from an image captured by the vehicle, then the vision algorithm allows the bang-bang control to take the glider to the target zone.

B. Technique overview

Our approach is based on feature points obtained by a SURF point detector [7]. The RANSAC [8] algorithm is used to fit a projective transformation between a reference image, taken at the beginning of the glider's flight trajectory, and the current image. We use the transformation obtained by the RANSAC method to calculate the position of the target coordinate in the current image. Since characteristics like scale, field of view and viewing angle of the images taken by the onboard camera change quickly, we propose an indicator to choose specific frames where the reference image should be changed by the current one. Our approach is capable of providing the input to a classical bang-bang control algorithm to reach a target point with a small distance error. The algorithm was tested using computer rendered images of quasi-planar scenes. The simulated conditions lead us to conclude that the algorithm is useful in the guidance of gliders when the target point is contained in a quasi-planar region and there are good visibility conditions from the release to the target point.

II. TRACKING ALGORITHM

In this paper, a method to track a target coordinate during the flight path of an unmanned glider is discussed. The target is defined just before the release of the glider. Given the wide range of zones where the glider can be sent to, it's difficult to segment out the landing zone, and also impractical to create a database of possible targets. Other guiding alternatives are the use of lasers to illuminate the landing zone, but this requires the use of more than one aircraft, or that the aircraft remains close to the target zone. The type of image sequences and the estimation expected for a typical glider's flight can be seen in Fig. 1.

We propose a method to overcome the above mentioned problems, using SURF [7] feature points. The method estimates the target coordinate during the glider's flight and helps with the guidance to the desired landing zone. Projective transformations are used to find the target coordinate during the flight trajectory. A block diagram of the algorithm can be seen in Fig. 2.



Fig. 1. A typical image sequence of a glider's trajectory. The orange crosses indicate the expected image estimation.

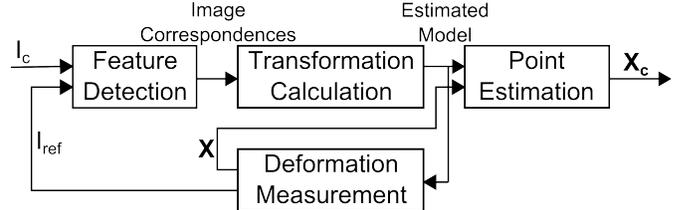


Fig. 2. Block diagram of the algorithm

A. Feature Detection

In order to estimate the location of the target coordinate in the image sequence, a SURF feature detector is used to find correspondences between the reference image I_{ref} and the current image I_c . Shortly, SURF can be described in two phases. The first one uses a Hessian matrix to find interest points (blob-like points). The second one uses a Wavelet Haar transformation to create a 64 vector that describes the region around the interest point. The descriptor contain information about the gradients' orientation in the region. SURF uses masks of different sizes to find feature points in different scales. The filtering process is done using integral images, which makes this algorithm extremely efficient. The type of points obtained after processing a test image can be seen in Fig. 3. A detailed description of the algorithm can be found in [7].

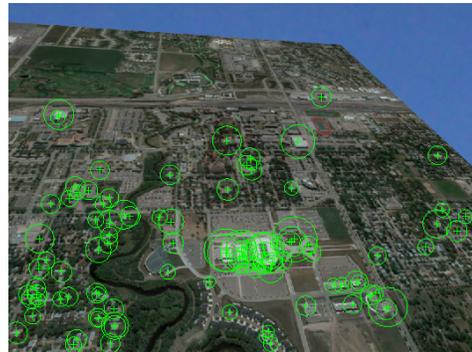


Fig. 3. Feature points obtained by SURF algorithm in a test image

After calculating the feature points and its descriptors, correspondences are found using a sum of squared differences (SSD) distance. This process is usually known as matching.

B. Transformation Calculation

To determine the transformation that relates I_c with I_{ref} , the RANSAC [8] algorithm is used. RANSAC is an algorithm created to fit a model to experimental data. RANSAC is capable of interpreting/ smoothing data containing a significant percentage of gross errors, and is suited for our application where the correspondences obtained through the feature point detector are error-prone.

RANSAC is used to calculate a projective transformation that relates the point correspondences found in I_c and I_{ref} . The equation that represents the transformation is described by:

$$\bar{x}_c = \mathbf{A}\bar{x} = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} \bar{x} \quad (1)$$

where \bar{x}_c represents the homogeneous coordinates of a point in I_c , \bar{x} represents the homogeneous coordinates of a point in I_{ref} and \mathbf{A} is the projective matrix that takes coordinates in I_{ref} and converts them in coordinates in I_c .

C. Point Estimation

The point estimation step takes the estimated model \mathbf{A} and the known coordinates in the reference image $\bar{x} = [X_{ref}, Y_{ref}, 1]^T$, and calculates the output coordinate \bar{x}_c in I_c .

D. Deformation Indicator

According to [9] the uncertainty of an estimated transformation depends on many factors, including the number of points used to compute it, the accuracy of the given point matches, as well as the configuration of the points in question. In our application, when trying to use the first image as reference image during the whole flight path, the estimated coordinate was unstable due to various reasons: the low number of correct matched points that are present as input to the RANSAC algorithm; the low number of feature points, found in last images, that match points in the first image (because the viewed zone corresponds to a small region of the first image); the content of the feature vectors change because there's a significant modification in the point and angle of view. Thus, the idea was to create an indicator of change in the characteristics of the images and in the feature vectors, allowing to decide when the reference image should be replaced by a new one.

To create such measurement, the points p_1, p_2, p_3, p_4 are defined in the reference image as:

$$\begin{aligned} p_1 &= (0.25W, 0.25H) & p_2 &= (0.75W, 0.25H) \\ p_3 &= (0.75W, 0.75H) & p_4 &= (0.25W, 0.75H) \end{aligned} \quad (2)$$

where W, H are the width and the height of the image respectively. The points p'_1, p'_2, p'_3, p'_4 are calculated using the estimated transformation as $\bar{p}'_n = \mathbf{A}\bar{p}_n$. The deformation measurement is defined as:

$$def = |d'_1 \cdot d'_2| + |d'_2 \cdot d'_3| + |d'_3 \cdot d'_4| + |d'_4 \cdot d'_1| \quad (3)$$

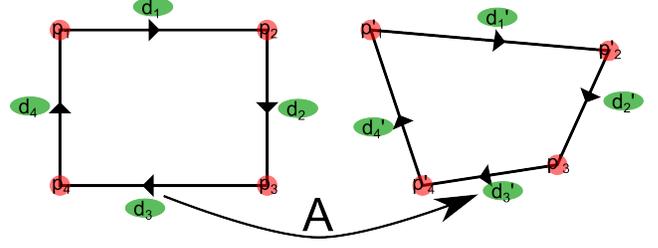


Fig. 4. Transformation of the rectangle used by the deformation measurement under a projective transformation \mathbf{A}

where d'_1, d'_2, d'_3, d'_4 are the direction vectors in the current image defined as:

$$\begin{aligned} d'_1 &= \frac{p'_2 - p'_1}{\|p'_2 - p'_1\|} \\ d'_2 &= \frac{p'_3 - p'_2}{\|p'_3 - p'_2\|} \\ d'_3 &= \frac{p'_4 - p'_3}{\|p'_4 - p'_3\|} \\ d'_4 &= \frac{p'_1 - p'_4}{\|p'_1 - p'_4\|} \end{aligned} \quad (4)$$

The indicator increases when the angle between two direction vectors is different than 90° . Since the region considered to calculate the feature vectors in SURF algorithm don't consider the changes in the viewing angle or point of view, the proposed indicator acts as a quantification of the degradation of the feature vector obtained by SURF. Also, the indicator changes with the modification of the distribution of the points in the image, which means that the uncertainty of the transformation estimation has been modified. In our algorithm, a threshold value is used to decide when the reference image should be replaced by a new one.

The work presented in [10] uses the number of matched points over the number of points found in the reference image, as an indicator of the uncertainty of the estimated transformation. But, the indicator presented in our work, better reflects the characteristics of uncertainty described in [9] and analyzed at the beginning of this section.

III. GLIDER'S TRAJECTORY SIMULATION

In [2], the authors propose a dynamic model of the an unmanned glider based on the description done in [11]. The equations describing the model are:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix} = \begin{bmatrix} c\psi c\theta u + (c\psi s\phi s\theta - c\phi s\psi)v + (s\phi s\psi + c\phi c\psi s\theta)w \\ c\theta s\psi u + (c\phi c\psi + s\phi s\psi s\theta)v + (c\phi s\psi s\theta - c\psi s\phi)w \\ -s\theta u + c\theta s\phi v + c\phi c\theta w \\ -gs\theta + C_x(\alpha, \beta, \delta_r, \delta_e)qS/m + Rv - Qw \\ gs\phi c\theta + C_y(\alpha, \beta, \delta_r, \delta_e)qS/m + Pw - Ru \\ gc\phi c\theta + C_z(\alpha, \beta, \delta_r, \delta_e)qS/m + Qu - Pv \\ P + Qs\phi t\theta + Rc\phi t\theta \\ c\phi Q - s\phi R \\ Qs\phi/c\theta + Rc\phi/c\theta \\ QR(I_{yy} - I_{zz})/I_{xx} \\ (C_m(\alpha, \delta_e)qSc + PR(I_{zz} - I_{xx}))/I_{yy} \\ (C_n(\beta, \delta_r)qSb + PQ(I_{xx} - I_{yy}))/I_{zz} \end{bmatrix} \quad (5)$$

Where $[x, y, z]^T$ is the position vector of the glider respect to an earth fixed coordinate system, $[u, v, w]^T$ are the body frame velocities, $[\phi, \theta, \psi]^T$ are the three Euler angles describing the glider orientation and $[P, Q, R]^T$ are the body frame angular velocities. The body frame velocities $[u, v, w]^T$ go in the direction of $[X_b, Y_b, Z_b]^T$ respectively. The control inputs are the angle of the elevator (δ_e) and the angle of the rudder (δ_r).

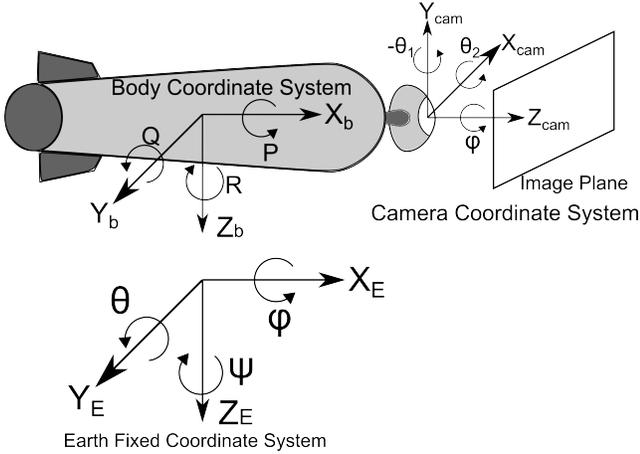


Fig. 5. Earth Fixed Coordinate System, Body Coordinate System and Camera Coordinate System

Fig. 5 shows the coordinate systems involved in the simulation. Since the camera on-board of the glider is aero-stabilized, the camera's attitude depends on the earth fixed coordinate velocities of the glider and not in the orientation itself. The following equations describe the camera's orientation angles.

$$\theta_1 = \text{atan2} \left(\frac{V_{Y_E}}{V_{X_E}} \right) \quad (6)$$

$$\theta_2 = \text{atan2} \left(\frac{V_{Z_E}}{\sqrt{V_{X_E}^2 + V_{Y_E}^2}} \right) \quad (7)$$

where $[V_{X_E}, V_{Y_E}, V_{Z_E}]^T$ is the velocity vector of the glider respect to the earth fixed coordinate system.

A. Bang-Bang Control

To guide the glider to a target coordinate $P_t = [X_t, Y_t, Z_t]^T$ a bang-bang control was used. When the camera is looking at the target point, the controller moves the elevator and rudder to align the projection of P_t to the center of the image. Fig. 6 shows the control signals generated by the controller.

| | | |
|--------------|--------------|---|
| $\delta_r -$ | $\delta_r +$ | $\delta_r +$ yaw right $\delta_r -$ yaw left |
| $\delta_e +$ | $\delta_e +$ | |
| $\delta_r -$ | $\delta_r +$ | $\delta_e +$ nose up $\delta_e -$ nose down |
| $\delta_e -$ | $\delta_e -$ | |

Fig. 6. Control signals generated by the bang-bang algorithm

The dynamic model of the glider and the vision estimation algorithm were implemented using MATLAB. Since our algorithm always took less than 100ms, we set the time step of the MATLAB simulation to that value. The simulation reflects the behavior of a system running the algorithm online.

IV. SIMULATION RESULTS

To test the behavior of the control system using our vision based estimation, many simulations in MATLAB environment were performed. We used quasi-planar scenes. High resolution images were used as textures for the scenes. In this section, two different type of results are presented. First, the results of releasing the glider with given initial conditions to a specific target are presented, then an evaluation of the improvement generated by the deformation indicator is shown. Below, the results of releasing the glider from $[X_{E_0} = 0m, Y_{E_0} = 0m, -Z_{E_0} = 1524m]$ are presented. We set $[u_0 = 102.8889m/s, v_0 = 0m/s, w_0 = 0m/s]$, $[\phi_0 = 0^\circ, \theta_0 = -10^\circ, \psi_0 = 0^\circ]$ and $[P_0 = 0^\circ/s, Q_0 = 0^\circ/s, R_0 = 0^\circ/s]$ as initial conditions. The target zone was located in $[X_t = 1930m, Y_t = 300m, Z_t = 0m]$. The simulation showed that the glider landed in $[X_f = 1928.07m, Y_f = 302.76, Z_f = 0]$, which is just 3.34m away from the desired location.

Fig. 7 shows the flown trajectory of the glider. Fig. 8 shows the attitude of the glider. The yaw and pitch angles change with time allowing the vehicle moving toward the target. Since the camera is aero-stabilized, its orientation depends on the earth fixed frame velocities of the glider, which makes the changes in the camera's attitude smoother than the ones in the glider's orientation. However, camera's attitude changes are not smooth during the whole trajectory, this is due to the stronger moments generated by the controls signals when the vehicle is moving faster. Fig. 10 shows the control signals generated using the bang-bang

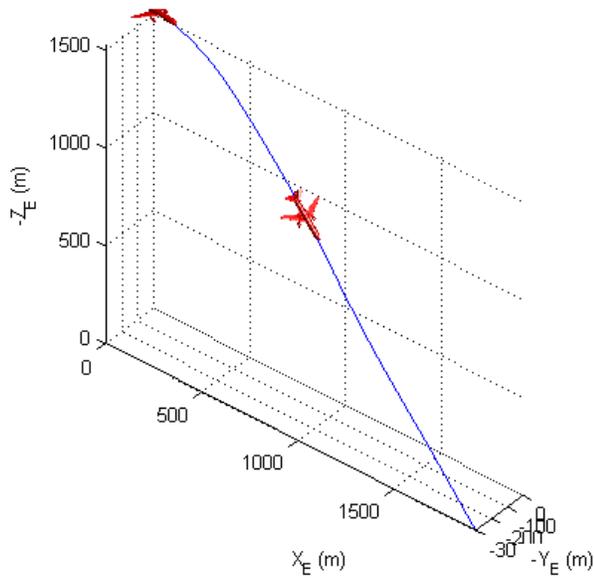


Fig. 7. Glider's flown trajectory

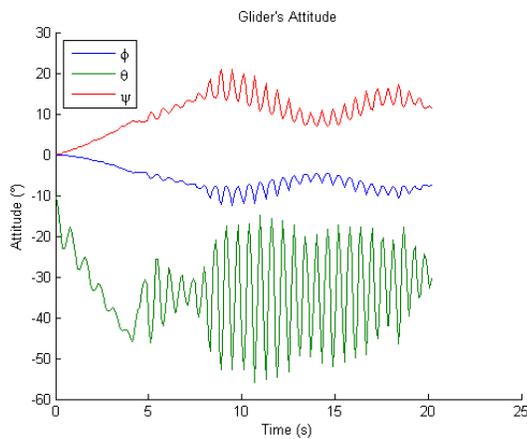


Fig. 8. Glider's attitude during the flown trajectory

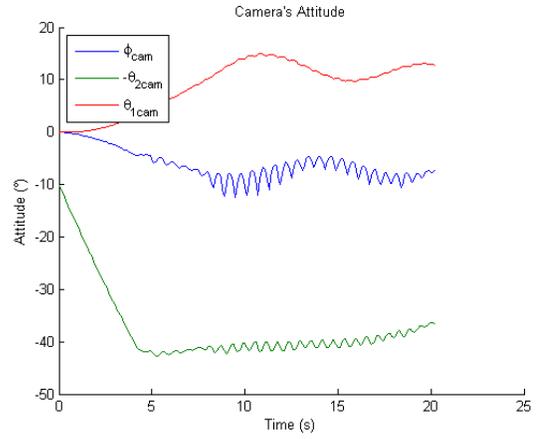


Fig. 9. Camera's attitude during the flown trajectory

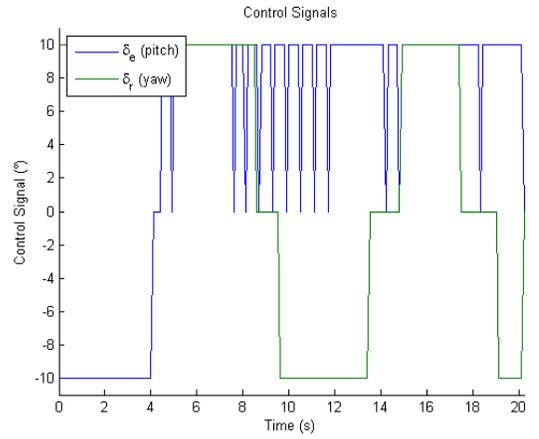


Fig. 10. Control signals generated during the flown trajectory

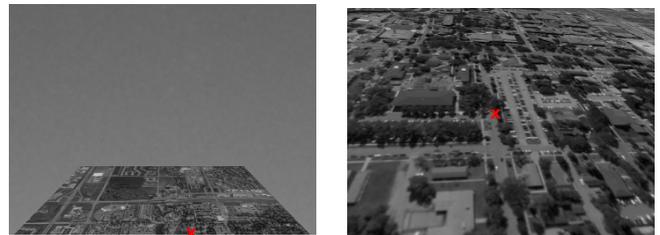
control and the estimation algorithm, the δ_e and δ_r angles change to align the projection of P_t to the center of the image.

A. Vision Estimation Results

In this section, the results of the estimation algorithm are presented, considering the flown trajectory described above. The evaluation was done in various ways. First a qualitative evaluation of the estimation was performed. Videos were created showing the estimation during the whole flight. In this article, the first and last images of the video are shown. Second, the error of the estimation in the two horizontal axes in the earth fixed coordinate frame is calculated. Finally, the error in pixels of the estimated coordinate compared to the ideal location of the projection of P_t is calculated.

Fig. 11 shows the qualitative results of the estimation algorithm. The qualitative result shows that after processing the images captured by the glider during the flown trajectory, the target zone remains the same.

Fig. 12 shows the estimation errors in the earth fixed frame.



(a) First image, the red cross indicates the user input coordinate

(b) Last image, the red cross indicates the estimated coordinate at the end of the trajectory

Fig. 11. Qualitative Result of the Estimation Algorithm

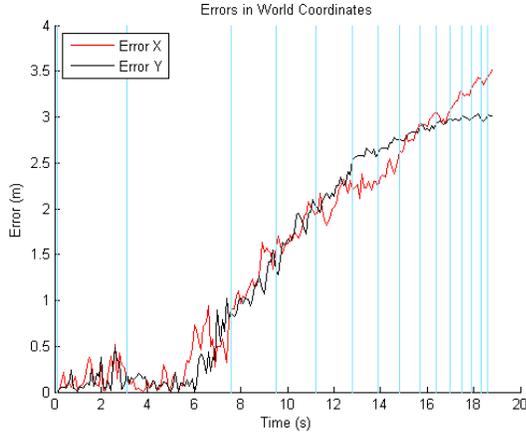


Fig. 12. Estimation error in the two horizontal axes $[X_E, Y_E]$ in the earth fixed frame

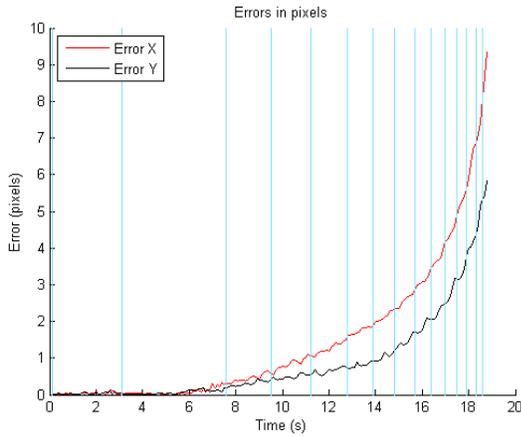


Fig. 13. Estimation error in the image plane

The light blue lines show the frames where the reference image was changed, which means that the threshold of the deformation indicator was exceeded. Fig. 13 shows the estimation errors in the image plane, the processed images are 800×600 pixels. Due to the smooth movements of the glider during the first seconds of the flight, the estimation error increases slowly as shown in Fig. 12 and Fig. 13. At the end of the trajectory, the moments generated by the changes in δ_e and δ_r are stronger, and that causes stronger changes in the glider's and camera's attitude. Something similar happens with the increase in scale. Since the viewing area reduces with time and the velocity increases, the scale increases faster at the end of the trajectory. Another factor that makes the estimation harder toward the end is the oscillation of the roll angle ϕ . To analyze Fig. 12 and Fig. 13, is important to consider that the area represented by a pixel at the beginning of the trajectory is bigger than the area represented at the end. This means, that even when the error in the fixed coordinate system doesn't increase with time, the error in pixels could increase. For this

application, it's more important to look at the error in the fixed coordinate system, because it represents the measurement error to which the control system is subjected to.

The figures above also show the behavior of the deformation indicator. Since at the beginning of the trajectory the changes in the image are smooth, the reference image is kept for longer periods of time than at the end of the trajectory. The performed simulations showed that using the indicator reduces the estimation error compared to changing the reference image in every frame. Also, the possibility of keeping the first image as reference image for the whole sequence was explored, but results showed that after a certain number of frames the RANSAC algorithm could not find a valid transformation, or found a wrong one.

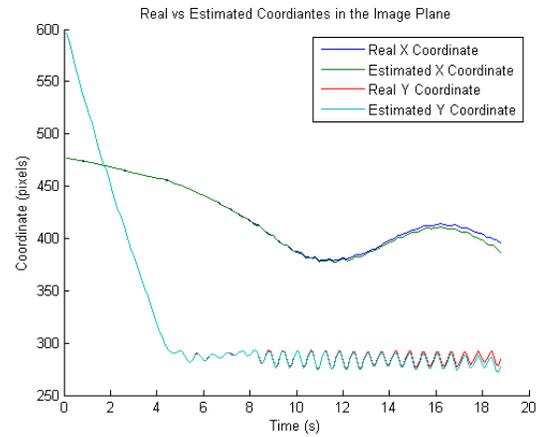


Fig. 14. Real vs Estimated Image Coordinates

Fig. 14 shows the comparison between the ideal projection of P_t versus the estimated coordinate using our algorithm. The bang-bang controller attempts to keep the projection of P_t in the center of the image $[X = 400, Y = 300]$. During most part of the trajectory, the vision algorithm is highly accurate, but toward the end of the trajectory differences between the ideal and estimated coordinate appear. For the application in question, it's more important to have a good estimation during the first part of the flight, because at the end, the changes in the control signals δ_e and δ_r may cause instability to the system.

B. Deformation Indicator Evaluation

In order to evaluate the performance of the proposed deformation indicator as a method to define when to change the reference image, a set of 160 simulations was performed. 80 simulations considered the indicator, and the other 80 changed the reference image every iteration. Fig. 15 shows a box-plot of the obtained results. The box plot shows that the median estimation error obtained when using the deformation indicator is smaller than the median error obtained when changing the reference image every iteration. This means, that in most simulations the use of the indicator helps reducing the

estimation error at the end of the trajectory.

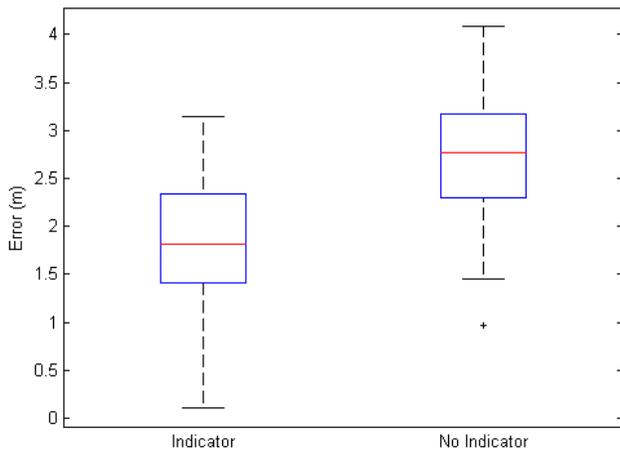


Fig. 15. Comparison of the estimation error between the vision algorithm using the deformation indicator and the vision algorithm that changes the reference image every iteration. For each case 80 simulations were performed.

V. CONCLUSION

This paper presented a method to track a target coordinate for an unmanned glider. The tracked coordinate was used in a bang-bang control system to guide the glider. The proposed algorithm consists of a feature point detector followed by a RANSAC algorithm to estimate a projective transformation. A deformation indicator was added to the system to decide when was the right time to replace the reference image. The simulation results showed that the algorithm is able to provide inputs to the guidance system with high accuracy.

One of the advantages of the proposed algorithm is that it only requires the target coordinate specified in one image to work. This extends the applications of gliders to situations where specifying the GPS target coordinate is difficult. It also facilitates escaping maneuvers for the aircraft in charge of releasing the glider.

Until now, the algorithm has been tested under high visibility situations. This leads us to conclude that this algorithm works when there is visibility from the release to the target point. To avoid interference caused by weather conditions like rain or clouds, near-infrared or infrared images could be considered. This would extent the applicability of the approach.

As future work, we intend to focus on the problem of considering scenes that are not quasi-planar. Now, the algorithm can work in geographical zones that are almost even. To consider rugged regions we need to rethink the transformation between images. Other lines of work are considering applications like the generation of trajectories based on the vision algorithm for unmanned quadrotors or unmanned helicopters.

ACKNOWLEDGMENT

The authors thank *Industria Militar de Colombia* (INDU-MIL) for financing this research.

REFERENCES

- [1] K. S. Kumar, G. Kavitha, R. Subramanian, M. Munna *et al.*, "Image fusion of video images and geo-localization for uav applications," *ACEEE International Journal on Information Technology*, vol. 1, no. 2, 2012.
- [2] A. Perez and C. Rodriguez, "Guidance of an autonomous glider to a target zone," in *5th Annual Dynamic Systems and Control Conference and 11th Motion and Vibration Conference*. ASME, 2012.
- [3] T. B. Criss, M. M. South, and L. J. Levy, "Multiple image coordinate extraction (mice) technique for rapid targeting of precision guided munitions," *Johns Hopkins APL technical digest*, vol. 19, no. 4, pp. 493–500, 1998.
- [4] P. Campoy, J. F. Correa, I. Mondragón, C. Martínez, M. Olivares, L. Mejías, and J. Artieda, "Computer vision onboard uavs for civilian tasks," in *Unmanned Aircraft Systems*. Springer, 2009, pp. 105–135.
- [5] C. Pan, H. Deng, X. F. Yin, and J. G. Liu, "An optical flow-based integrated navigation system inspired by insect vision," *Biological cybernetics*, vol. 105, no. 3-4, pp. 239–252, 2011.
- [6] I. F. Mondragon, P. Campoy, J. F. Correa, and L. Mejias, "Visual model feature tracking for uav control," in *Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on*. IEEE, 2007, pp. 1–6.
- [7] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [8] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [9] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge Univ Press, 2000, vol. 2.
- [10] M. Burke and W. Brink, "Estimating target orientation with a single camera for use in a human-following robot," 2010.
- [11] G. M. Siouris, *Missile guidance and control systems*. Springer, 2004.
- [12] R. Szeliski, *Computer vision: algorithms and applications*. Springer, 2011.