Visual servoing of UAV using cuboid model with simultaneous tracking of multiple planar faces

Manlio Barajas*[1], José Pablo Dávalos-Viveros[1], Salvador García-Lumbreras[1] and J.L. Gordillo[1]

***Abstract*— Pose estimation is a key component for robot navigation and control. An Unmanned Aerial Vehicle (UAV) that is instructed to reach certain 3D poses requires a way of measuring its current pose so it knows when they are reached. In this article a method for UAV control that uses the 3D orientation and position obtained by a 3D tracking process based on a cuboid model as control feedback is presented. In particular, a simultaneous face tracking strategy where 3D pose estimations from different faces are combined is presented. Face combination was validated using a robotic arm with a cuboid at the final joint. For the UAV control, hover and path following tasks were tested. Results show that the proposed method handles changes in pose correctly, even though no face remains visible at all the times. Also, the UAV maintained a low speed in order to satisfy the small inter-frame displacement constraint imposed by tracking algorithms. This approach is an advantage for remote pose estimation algorithms, since in many situations it is not possible to rely on a single planar face.**

I. INTRODUCTION

A localization method and a control strategy are determinant elements in successful autonomous navigation tasks for unmanned vehicles. Visual 3D pose estimation of objects in real environments as a localization method has been an important topic in literature because cameras have shown to be a reliable source of environment information. Common visual approaches depend on depth information by means of stereo setups [1], [2] or laser range data [3].

With the advent of low cost drones, Unmanned Aerial Vehicle (UAV) control has become an active research topic. In particular, visual servoing is a common control approach to this problem since there are well known methods for achieving position estimation at image level. For example, Image Based Visual Servoing (IBVS) algorithms presented in [4], [5], [6], [7] are based on the tracking of a landmark placed on the floor. Others utilize IBVS for tracking a moving target and following it (physically) [8], [9]. Some approaches are on the context of Simultaneous Localization and Mapping (SLAM) where multiple sensor information is fused for more robust pose estimation [10], [11], [12]. Stereo vision is also used for pose estimation [13], [14]. These methods usually rely on the data obtained through multiple sensors on the drone for a robust control system.

3D object tracking can be used to deliver 3D pose information to the vehicle controller. This may be accomplished by using full featured models [15], [16], [14] or by using approximated models [17]. While full featured models allow having a very exact model representation, this is not well suited for all situations due to the complexity of the object or the need to render the 3D model. Approximated models can be used to overcome this constraint. This was validated at theoretical level in [18].

This research is based on the method presented in [18] for 3D pose tracking using a remote camera and tracking individual planar faces of a cuboid shaped object. For the UAV visual servoing, the control system used is the one presented in [19]. What is introduced here is an improvement that consists in simultaneously tracking where possible more than one face of the cuboid per frame and then applying a combination. Additionally, path following exercises are done using a Parrot AR.Drone quadrotor, since in [19] only results for hover are presented. The quadrotor, which has 6 DOF, is manipulated using the feedback from the visual pose estimation method in a Position Based Visual Servoing (PBVS) control system.

This article is divided as follows: section II will introduce related work to the problem of 3D tracking and UAV visual servoing. On section III, a method for 3D object tracking using multiple planar faces is reviewed. Section IV presents the control strategy used to achieve control over the UAV for then, in Section V, presenting the experimental results and a comparison with other methods used for UAV control. In section VI conclusions are remarked.

II. RELATED WORK

Since in this work a PBVS system is used, the problem can be split into two parts. First, visual 3D pose tracking, that consists in visually determining the 6 parameters that the determine the pose of a tracked object. Second, the control strategy that is used for controlling the vehicle, provided the current pose of the object is known.

For 3D pose estimation and tracking, our work relies on homography decomposition for 3D reconstruction and standard 2D plane tracking. The plane tracker used was introduced by Benhimane and Malis in [20], where it is also used for visual servoing.

There are other alternatives for 3D tracking in context of image alignment. The method presented by Cobzas and Sturm [21] uses standard 2D tracking over a 3D pose parameter space (6 parameters, one for each DoF). A drawback is that, when changing the parametrization to handle 3D pose changes, it results that even some movements that could be tracked without problems by tracking planes individually can produce failure. Another idea introduced by them is to apply constraints between planes to make tracking more robust.

[1]Center for Robotics and Intelligent Systems (CRIS), Tecnológico de Monterrey, Monterrey 64849, Mexico

*Corresponding author manlito@gmail.com

On the same line of parametrizing directly in euclidean space, Panin and Knoll [15] proposed a method for tracking objects that uses Mutual Information as similarity measure, instead of the common sum of squared differences (SSD), and perform a Levenberg-Marquardt optimization. Authors report performance of 2 *fps* because of the Mutual Information step. Our approach, on the other hand, works at more than 24 *fps*. Moreover, Panin's method requires a projection of a full CAD model to the image plane on each iteration and a z-test for determining visibility. We just require this projection once per registration, and can be simplified to a partial 3D reconstruction. However our method is limited to polygon meshes, but that is a desired feature towards automatic plane detection.

Visual servoing of aerial vehicles has been a thoroughly researched topic. For example, in [5], a pure image based visual servoing (IBVS) for landing and take-off process is developed. While this work has the advantage of not requiring a full 3D reconstruction, it can't be used for full object control (executing paths). In [6], a planar patch is also used for 3D pose estimation of the UAV. This is closer to our work, but it's restricted in space. In [10] a monocular SLAM system is used to eliminate odometry drift. This method proved to be very robust, but it requires the use of internal sensors like gyroscopes, accelerometers and cameras. In a similar manner, in [11] a SLAM algorithm tracks the pose of the on board camera and an on board PD controller is used for the attitude control. In [13] the full object pose is also controlled. However, this approach requires a stereo setup for 3D reconstruction and uses IMU data for rotational dynamics. A common pattern in these works is the use of on board cameras and sensors of the UAV. In our work, we only used a single remote camera without relying on any data provided by the internal sensors. An example of a quadrotor controlled by remote cameras has been presented in [14] but this method also utilizes the on board sensors and is based on a binocular system with a previously known CAD model for the pose detection.

III. 3D TRACKING OF OBJECTS USING MULTIPLE PLANAR FACES

In this research it is presented a method for Position Based Visual Servoing (PBVS) that uses as feedback the 3D pose (3D orientation and 3D position) estimation of the UAV using the 3D tracking method proposed in [18]. This method uses planar faces as visual tracking candidates, over which standard plane tracking is done [22].

In the tracking method presented in [18], only one face is tracked at a time and a face selection criteria is used to determine which face is the optimal for tracking. Once a 2D homography between the reference template and the current image is obtained, homography decomposition is used to obtain a 3D reconstruction of that face. Later on, a set of precomputed 3D transforms between the different faces of the model allow obtaining the full 3D reconstruction of the model.

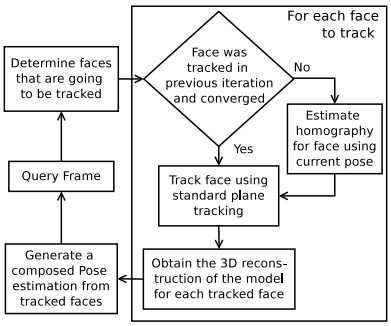

Fig. 1. Overview of of 3D tracking process using simultaneous tracking of planar faces. The number faces that are tracked simultaneously depends on the shape of the model and the number of visible faces.

Here we propose to use, when possible, more than once face at a time. For possible we mean that face is visible and that plane tracking converges to the correct solution. Then, multiple poses are obtained which most be integrated. In this article, a simple averaging rule is used, that is based on averaging.

Without the loss of generality, a cuboid model is used. But it can be extended to other polygon meshes, as far as the faces be planar, the relation between them be known and they be large enough to be tracked without sampling related issues. The full 3D pose tracking process is shown in Fig. 1.

For the proposed method it is assumed that the initial state of the target object is known. As it will be described, this can be obtained by using a single planar face, for which its initial homography is known.

A. Cuboid Model

For this work, a cuboid model is used. A cuboid can be defined in terms of the 6 planar faces that are part of it as:

$$C = \{\mathbf{P}_i \mid 1 \leq i \leq 6\} \tag{1}$$

Where $\mathbf{P}_i$ refers to the homogeneous transformation matrix for plane π_i from its current referential to camera referential. For all faces in the cuboid C there exist at least one transformation matrix ${}^{i}\mathbf{T}_k$ that maps face referential *k* to face referential *i*. This can be expressed as:

$$\forall \pi_i \in C \; \exists \; {}^{i}\mathbf{T}_k \mid \mathbf{P}_i = \mathbf{T}_k \mathbf{P}_k \tag{2}$$

In the case of the cuboid used, these transforms can be estimated by assuming certain dimensions (up to scale factor) and then applying the corresponding transform.

B. Face Visibility

From the set of faces of cuboid C, only a subset of 1 up to 3 of the faces will be visible at anytime, assuming the cuboid is always visible by the camera. Moreover, there will be some cases in which, even if the face is visible, tracking it will not be possible due to sampling (Fig. 2).

To determine the faces that are visible and suitable for tracking, a similar criteria to that used in [18] is used. That subset of faces is obtained as follows:

Fig. 2. The UAV used for visual servoing. At the pose shown in the picture of the UAV, two faces of the cuboid are visible. Anyway, only the face with 'suns' should be used, since the other has a very small visible area.

$$\nu = \{i \mid \angle \mathbf{cn}_i \geq 0 \wedge \angle \mathbf{cn}_i < \epsilon,\ \forall i\ :\ 1 \leq i \leq 6\} \quad (3)$$

Where $\mathbf{c}$ refers to the center of the tracked object relative to the camera coordinate system. $\mathbf{n}_i$ is a normal vector to plane i. Threshold ϵ sets the maximum allowed angle between vectors $\mathbf{c}$ and $\mathbf{n}_i$.

C. Plane Tracking of Planar Faces

The 3D pose tracking method employed is based on standard plane tracking. In particular, the Efficient Second Order Minimization (ESM) proposed by Malis [23] was used. This method has proven to provide higher convergence with less global error [23].

This method works by iteratively updating parameters $\mathbf{p} := \mathbf{p} \circ \mathbf{\Delta p}$ where $\mathbf{\Delta p}$ can be evaluated as:

$$\Delta \mathbf{p} \approx -2(\mathbf{J}(\mathbf{e}) + \mathbf{J}(\mathbf{p}_c))^{+}(\mathbf{s}(\mathbf{p}_c) - s(\mathbf{e})) \quad (4)$$

Where $\mathbf{p}$ are the current parameters, $\mathbf{J}$ is the Jacobian, as presented in [22], and $\mathbf{s}$ is the transformed current image. $\mathbf{e}$ is the identity parameter set, in this case $\mathbf{0}$. This method assumes the following homography parametrization:

$$\mathbf{W}(\mathbf{x};\mathbf{p}) = \begin{pmatrix} 1+p_1 & p_3 & p_5 \\ p_2 & 1+p_4 & p_6 \\ p_7 & p_8 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (5)$$

D. Per Face 3D Object Reconstruction

To obtain a 3D reconstruction of object from which the pose is needed in order to feed the control loop, the approach proposed in [18] is implemented to 3D reconstruct the object provided a single face of it and, in this case, assuming the real world dimensions of the object are known.

In the 3D reconstruction stage, two main steps are done. First, the homography that is obtained using the ESM tracker is used to calculate its decomposition. Then, using transformations matrices, the center of the object is computed.

Homography decomposition deals with reconstructing the 3D pose of a planar surface given its projection on image plane. When working with plane tracking, usually only 8 parameters are used to accomplish tracking, which are mapped to the 3×3 homography transform, as shown in equation 5.

When using the pinhole camera model, the transformation matrix of extrinsic parameters $\mathbf{P} = [\mathbf{R} \mid \mathbf{t}]$ is a matrix composed of rotation $\mathbf{R}$ and translation $\mathbf{t}$. In order for $\mathbf{P}$ to be a valid transformation, $\mathbf{R}$ must have an orthonormal basis.

Using individual elements $h_{ij} : 1 \leq i,j \leq 3$ of $\mathbf{G}$, the following relations can be obtained:

$$\mathbf{P} = [\mathbf{r_1}, \mathbf{r_2}, \mathbf{r_3}, \mathbf{t}] \quad (6)$$

$$\mathbf{P} = \alpha \begin{bmatrix} \frac{h_{11}-c_x r_{31}}{f_x} & \frac{h_{12}-c_x r_{32}}{f_x} & \frac{\mathbf{r}_1 \times \mathbf{r}_2}{\alpha} & \frac{h_{13}-c_x}{fx} \\ \frac{h_{21}-c_y r_{31}}{f_y} & \frac{h_{22}-c_y r_{32}}{f_y} & & \frac{h_{23}-c_y}{fy} \\ h_{31} & h_{32} & & 1 \end{bmatrix} \quad (7)$$

Where h_{ij} are the elements of homography matrix. c_x, c_y are camera center offset parameters. f_x and f_y are focal distances of the camera. Normalizing factor α must be estimated such that the magnitude of $\mathbf{r_1}$ and $\mathbf{r_2}$ be unitary.

For the case of the proposed process (Fig. 1), this process must be done for each of the faces that are being tracked.

E. Pose Combination

When camera calibration process is done manually, camera parameters may not be accurate enough to produce a near to perfect homography decomposition. Also, it's well known that cameras may distort the image. In addition, as the tracked plane goes farther from the camera, sampling errors will also affect decomposition accuracy, since it's assumed that there is no camera distortion. For this reason, depending on the face that is being tracked, different pose estimations will be obtained.

The objective of pose combination is to merge the available 3D poses from tracked faces. The combination is done for rotation and translation for separate. As a preliminary step of the combination, faces are checked so that they are valid. This done by checking the average squared error of the plane tracking stage. Average squared error $\overline{e}$ is defined as:

$$\overline{e}(\mathbf{p}_c) = \frac{(\mathbf{s}(\mathbf{p}_c) - s(\mathbf{e}))^2}{l} \quad (8)$$

This equation is related to equation 4, since $s(\mathbf{p}_c) - s(\mathbf{e})$ is the *error image* at any given iteration. This error image is averaged using the number of pixels l of the template.

The combination of the rotation part of the pose is done by converting the available rotations to quaternions and then averaging. Since in quaternions $q_i = -q_i$ and poses should be close one to the other the following assignment rule is used:

$$q_i = \begin{cases} f(i) & q(i) \cdot q_0 > 0 \\ -f(i) & \text{otherwise} \end{cases}, \forall i \in \nu \quad (9)$$

Where f is a function that converts rotation matrix estimated using face i to quaternions. From the two possible quaternion representations, this expression selects the one that has the signs that correctly average them. Finally, the

averaged quaternion is normalized and returned to the matrix representation.

For the case of translation, an average is done per each translation dimension.

IV. UAV VISUAL SERVOING

The objective of this work is having an aerial vehicle to execute a given trajectory, provided a set of 3D poses that are part of the path. In the case of the implementation platform, possible manipulations are: *roll*, *pitch*, *yaw* and *vertical speed*. Roll and pitch control allow the UAV to move horizontally, while yaw can be used to have the vehicle's camera to look at certain places. Height control is used to increase or decrease the height of the drone.

The position control relative to the floor (xy plane) was achieved using a cascade controller. At the first level, a PD controller manipulated the speed in x and y at which the vehicle should move. In the nested loop, a PID controller, manipulated the roll and pitch variables of the drone. Fig. 3 shows this controller.

Controller design is closely related to the visual method used to determine the pose of the UAV. This method relies on standard plane tracking which assumes small inter-frame displacements, so the drone should not move faster than what the algorithm supports. This is important in the case of perturbations, where it's desired to return to the setpoint softly.

For the roll and pitch controller, the integral component of the speed controller plays a key role when perturbations are present, since in case of wind gusts, it will help to reduce the error by increasing the manipulation when the vehicle is having difficulty to move regardless of the pitch and roll manipulation.

For the yaw and vertical speed, simple proportional controllers were used, since built-in orientation and height control is stable for tracking. Fig. 4 shows the proposed controller for yaw. A similar controller is also used for height control.

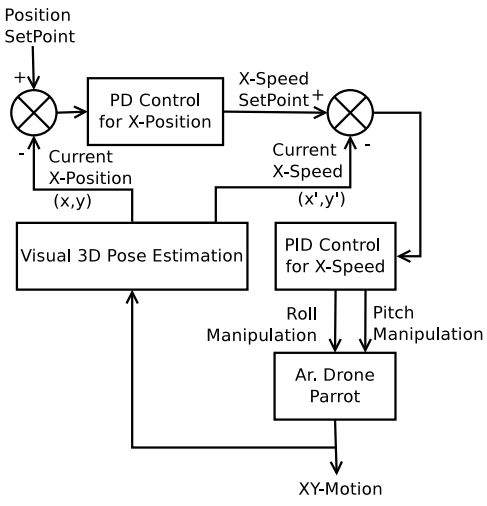

Fig. 3. Cascade controller controller for xy position.

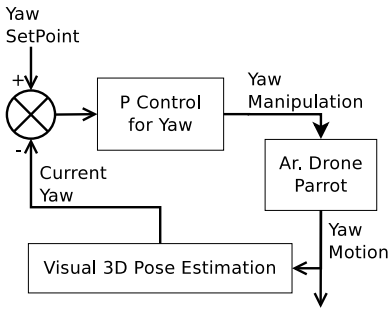

Fig. 4. Controller for yaw. A similar proportional controller is used for height control.

V. EXPERIMENTAL RESULTS

For experimentation, an *Ar.Drone Parrot* quadrotor unit was used (Fig. 2). At the top of it, a cuboid shaped surface was installed. This cuboid was used for 3D tracking and pose estimation. Control algorithms run from a desktop computer, which was connected to the drone through WiFi network.

Two different kind of experiments were performed. First, a validation exercise was carried to verify how the combination of faces was applied. Then, two experiments were done using the drone to test the visual servoing method.

A. Controlled Setup for Cuboid Tracking with Simultaneous Tracking of Faces

For this exercise, a replica of the cuboid installed over the Parrot was used. This cuboid was installed at the end joint of a robotic arm. Then, it was instructed to rotate 360° over that joint. Results from this experiment are shown in Fig. 5.

As part of this experiment, the robot was instructed to stop at a critical orientations, in order to evaluate how the combination was averaging the results. Table I shows a sample for one of these critical poses.

B. UAV Visual Servoing Experiments

The proposed method for control and pose tracking using simultaneous tracking of planar faces was tested using the drone. First, the stability of the control was verified by hovering the UAV. Some wind gusts were applied to verify its response to perturbations and the speed control (Fig. 6).

The control could accomplish the objective of responding softly to perturbations, regardless it was too far apart of the setpoint in some cases.

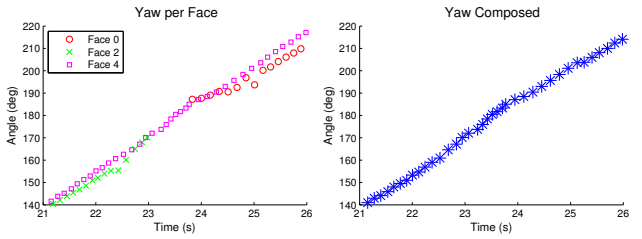

Fig. 5. Results for validation exercise using robotic arm. The Robot was instructed to apply a rotation in the yaw axis of the cuboid. As figure on left shows, not all faces can be tracked all the time, and each face may have a different estimation. Plot on the right shows the combined result.

TABLE I

INDIVIDUAL AND COMBINED ESTIMATION ON CRITICAL POSE

Pose	Measured	Face 0	Face 5	Combined
roll (deg)	45.0	48.5	42.4	45.4
pitch (deg)	0.0	-1.23	-0.5	-1.2
yaw (deg)	0.0	-5.6	-5.9	-5.6
x (mm)	25.0	27.4	26.9	27.2
y (mm)	700.0	693.9	695.3	694.6
z (mm)	0	-13.3	-13.5	-13.4

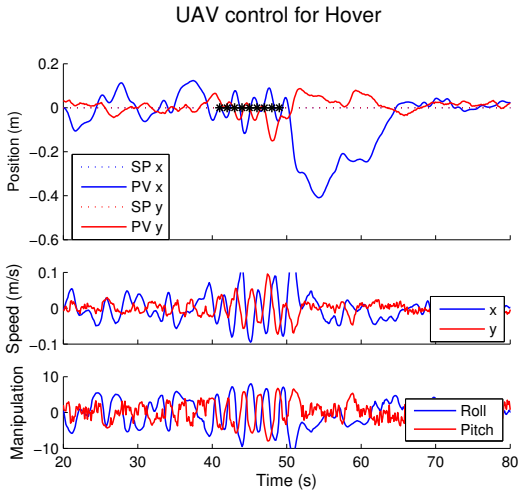

Fig. 6. Response of UAV control in hover position. Some wind gusts were applied (denoted as * in top plot). It must be noted that the speed (middle plot) was kept low. This is to avoid divergence problems related to the plane tracker. Last plot shows the manipulation output for both roll and pitch.

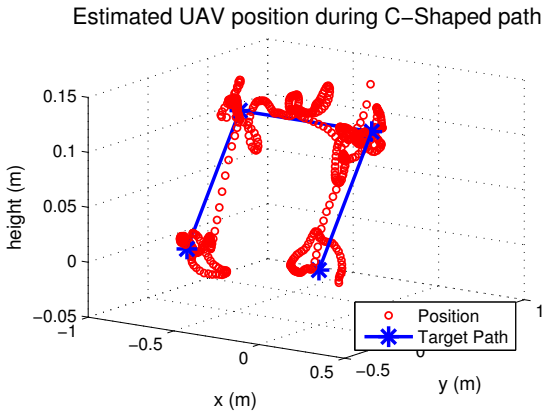

Fig. 7. Path executed by the drone in the C-Shaped experiment.

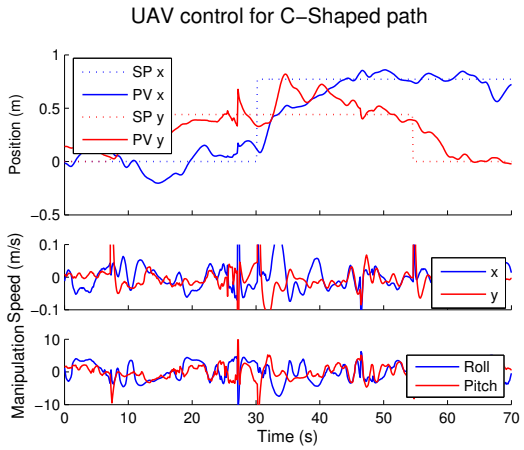

Fig. 8. UAV xy position control during the execution of C-Shaped path. Top shows how set points were applied and response. Some strong changes in pose estimation produced spikes in speed (middle). Manipulation is shown in bottom plot.

The second control experiment consisted in the execution of a path with the shape of a 'C' with a certain slope, for a total of 4 setpoints. The target path and the executed one, as measured by the 3D pose tracking process, are shown in Fig. 7.

Regarding the control variables for the C-Shaped path, in Fig. 8 position control is shown. Yaw and height control results are presented in Fig. 9.

C. Discussion

Results shown in Fig. 5 confirm that the proposed approach correctly handles the transition between faces, by smoothing the pose estimation which ultimately benefits the control stability. This allowed rotating without significantly affecting the control.

Another observation is that the UAV maintains a very low speed, regardless of perturbations. This is important since when perturbations are retired, the vehicle still moves at constant speed. Certainly this imposes a trade off since, if speed is increased, paths are executed better but the probability of divergence of the plane tracking algorithm increases, specially during critical poses, when two or more faces are visible according to Equation 3.

The speed is a notorious difference between this work and others, where different pose estimations methods are used. In both [10], [11] a hover and a path following exercise were presented using a visual SLAM which used the on board IMU data for attitude control and localization. This allowed them to react faster to perturbations.

It must be noted that our method relies solely on the visual feedback provided by a remote camera and because of the small inter-frame displacement assumption.

Other methods, such as those presented in [4], [5], [6], [7] have the limitation that they assume a certain mark or template is always visible. This clearly limits the possible navigation space, or requires multiple marks to be places along the way.

VI. CONCLUSIONS

In this work, a visual only method for UAV control that uses the faces of a cuboid as reference was presented. This approach differ from others on the fact that a remote camera is used to track the 3D pose of the UAV. The choose of the cuboid is related to the fact that, if using only a plane, there will be configurations in which it won't be visible to the camera.

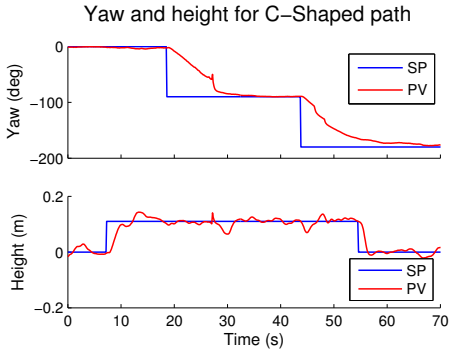

Fig. 9. Yaw (top) and height (bottom) control of UAV during the execution of C-Shaped path. Parrot drone has a stable built in control of these variables, which allowed to simplify the control over them by using only proportional control.

A validation experiment confirmed that there is the need for a way of transitioning smoothly between faces because poses are not reconstructed exactly. Also, since plane trackers may diverge, it was useful to define a criteria for determining if any given face should be tracked.

Regarding UAV control, we have validated this approach by executing paths that are in the three dimensional space. Since we required to have a slow speed, a cascade controller was implemented to have a fine speed control. Unfortunately it had the side effect of being slow to reach steady state.

At the current state, and similar to other works, navigation is limited to small areas. This is because sampling errors increase significantly when the vehicle is far from the camera.

An advantage of having a remote camera is that it doesn't have image stabilization issues and that it can track multiple vehicles at the same time. In posterior work, multiple vehicle tracking and additional face combination criteria will be evaluated. Another area of opportunity for this research is the fusion of data from drone IMU and other sensors with that of the vision. This could help increase the pose estimation during face combination stage.

ACKNOWLEDGMENT

Authors want to acknowledge Consejo Nacional de Ciencia y Tecnología (CONACyT) and e-Robots Research Chair from Tecnológico de Monterrey, Monterrey for supporting this research.

REFERENCES

[1] Tonko, M. and H.-H. Nagel, "Model-Based Stereo-Tracking of Non-Polyhedral Objects for Automatic Disassembly Experiments," *International Journal of Computer Vision*, vol. 37, no. 1, pp. 99–118, 2000.

[2] R. Munoz-Salinas, E. Aguirre, and M. Garcia-Silvente, "People detection and tracking using stereo vision and color," *Image and Vision Computing*, vol. 25, no. 6, pp. 995–1007, 2007.

[3] X. Shao, H. Zhao, and K. Nakamura, "Detection and tracking of multiple pedestrians by using laser range scanners," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, no. 1, 2007, pp. 2174–2179.

[4] O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel, and L. Eck, "Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle," vol. 25, no. 3, 2009, pp. 743–749.

[5] D. Lee, T. Ryan, and H. Kim, "Autonomous landing of a vtol uav on a moving platform using image-based visual servoing," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May, pp. 971–976.

[6] I. Mondragon, P. Campoy, C. Martinez, and M. Olivares-Mendez, "3d pose estimation based on planar object tracking for uavs control," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 35–41.

[7] R. Ozawa and F. Chaumette, "Dynamic visual servoing with image moments for a quadrotor using a virtual spring approach," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 5670–5676.

[8] V. Chitrakaran, D. Dawson, H. Kannan, and M. Feemster, "Vision assisted autonomous path following for unmanned aerial vehicles," in *Decision and Control, 2006 45th IEEE Conference on*, 2006, pp. 63–68.

[9] I. Mondragon, P. Campoy, M. Olivares-Mendez, and C. Martinez, "3d object following based on visual information for unmanned aerial vehicles," in *Robotics Symposium, 2011 IEEE IX Latin American and IEEE Colombian Conference on Automatic Control and Industry Applications (LARC)*, 2011, pp. 1–7.

[10] J. Engel, J. Sturm, and D. Cremers, "Camera-based navigation of a low-cost quadrocopter," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 2815–2821.

[11] M. Blosch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision based mav navigation in unknown and unstructured environments," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 21–28.

[12] M. Achtelik, M. Achtelik, S. Weiss, and R. Siegwart, "Onboard imu and monocular vision based control for mavs in unknown in- and outdoor environments," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 3056–3063.

[13] S. Salazar, H. Romero, J. Gomez, and R. Lozano, "Real-time stereo visual servoing control of an uav having eight-rotors," in *Electrical Engineering, Computing Science and Automatic Control,CCE,2009 6th International Conference on*, Jan., pp. 1–11.

[14] S. Klose, J. Wang, M. Achtelik, G. Panin, F. Holzapfel, and A. Knoll, "Markerless, vision-assisted flight control of a quadrocopter," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 5712–5717.

[15] G. Panin and A. Knoll, "Mutual Information-Based 3D Object Tracking," *International Journal of Computer Vision*, vol. 78, no. 1, pp. 107–118, 2007.

[16] J. A. Brown and D. W. Capson, "A Framework for 3D Model-Based Visual Tracking Using a GPU-Accelerated Particle Filter." *IEEE transactions on visualization and computer graphics*, vol. 18, no. 1, pp. 68–80, Feb. 2011.

[17] M. Manz and T. Luettel, "Monocular model-based 3D vehicle tracking for autonomous vehicles in unstructured environment," *(ICRA), 2011 IEEE*, pp. 2465–2471, 2011.

[18] M. Barajas, J. Esparza, and J. Gordillo, "Towards automatic 3d pose tracking through polygon mesh approximation," in *Advances in Artificial Intelligence IBERAMIA 2012*, ser. Lecture Notes in Computer Science, J. Pavn, N. Duque-Mndez, and R. Fuentes-Fernndez, Eds. Springer Berlin Heidelberg, 2012, vol. 7637, pp. 531–540.

[19] M. Barajas, J. Dávalos-Viveros, and J. Gordillo, "3d tracking and control of uav using planar faces and monocular camera," in *5th Mexican Conference on Pattern Recognition MCPR 2013*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, in-press.

[20] S. Benhimane and E. Malis, "Homography-based 2D Visual Tracking and Servoing," *The International Journal of Robotics Research*, vol. 26, no. 7, pp. 661–676, July 2007.

[21] D. Cobzas and P. Sturm, "3D SSD Tracking with Estimated 3D Planes," *Computer and Robot Vision, 2005.*, 2005.

[22] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221 – 255, March 2004.

[23] E. Malis, "Improving vision-based control using efficient second-order minimization techniques," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 2, 26-May 1,, pp. 1843–1848 Vol.2.