2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

Vision-Based Qualitative Path-Following Control of Quadrotor Aerial Vehicle

Trung Nguyen, George K. I. Mann and Raymond G. Gosine

Abstract— **This paper presents a vision-based qualitative 3D navigation technique as well as first results of adapting Funnel Lane theory into path-following control of quadrotor aerial vehicle. The image's Kanade-Lucas-Tomasi (KLT) corner features are detected along the reference path in order to build a funnel lane for navigation. Then a funnel-lane navigation calculation is developed to estimate the desired yaw angle and height for the next movement. The proposed algorithm uses the front camera, heading measurement and altimeter of the Ar.Drone quadrotor for navigation. The remarkable advantage of the proposed technique is independently working in GPS-denied environments without the support of the external tracking system as well as computationally efficient. As compared to other available approaches, at-least one matched feature is required during path following. The proposed navigation technique can be implemented for visual-homing, visual-servoing and visual-teach-and-repeat (VT&R) applications. The proposed method is simulated in ROS and Gazebo simulator followed by a realtime experiment with the Ar.Drone quadrotor.**

Index Terms— **micro- and mini- UAS; navigation; visual teach and repeat; visual servoing, quadrotor.**

I. INTRODUCTION

In recent years, the quadrotor type of Unmanned Aerial Vehicles (UAV) has received much scientific consideration and has become the fastest developing technology of autonomous vehicles. In order to improve the degree of autonomy, it is necessary for the quadrotor to be integrated with an autonomous path following technique. While ground vehicles have the ability to use odometry for navigation and control during path following, quadrotor faces the challenge of estimating their position reliably. To solve the navigation problem or to set reference points for path-following control, many strategies have been investigated in the past. Some researchers utilize Global Positioning Systems (GPS) to do path following. However, the accuracy of GPS becomes a considerable issue for indoor environments. For indoor applications, the work demonstrated in [1] has studied an approach for the micro quadrotor, and this work is heavily dependent on the VICON Motion Capture System for position estimation. This element prevents their solution from UAV exploration and autonomous missions. Simultaneous Localizing And Mapping (SLAM) is proposed as another solution to identify the location of the quadrotor [2][3]. However, SLAM is not considered an effective solution in the absence of a powerful sensory system, and it demands higher energy and computational power. Image processing has been studied as a viable candidate for navigation issues. For example, Bills et al [4] use Single Image Perspective Cues to control autonomous micro aerial vehicle flight in indoor environments. However, their work is limited for navigating in corridors of buildings with long, straight parallel lines. The works [5][6] have demonstrated artificial marker-based quadrotor landing. The marker technique allows the quadrotor to perform autonomous landing, but the working space is limited to a camera's field of view and the area that it can cover is limited due to the placement of the marker. In Courbon's study [7], images and landmark features of the environment from a calibrated fisheye camera have been sampled and stored in order to provide visual-guiding for path-following control. The disadvantage of their approach is computationally inefficient and it requires at least five couples of matched features to define the desired position and orientation, which narrows the reliability of the system.

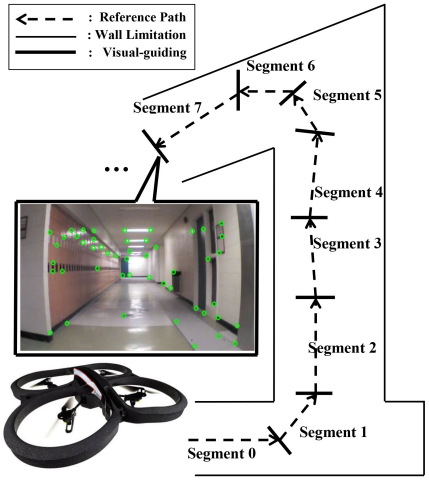


Fig. 1. Ar.Drone quadrotor ver 2.0 does path-following with visual servoing

This paper describes a vision-based 3D navigation technique for UAV in path-following control. Based on the reference images and corner features, the UAV can estimate the necessary yaw angle and height for next movement. The navigation technique derives from Funnel Lane theory, proposed in Chen and Birchfield's papers [8][9]. In their ground vehicle's application, the theory is applied to control the yaw angle based on the image's horizontal coordination of the detected landmark features. As mentioned in their

*This work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), C-CORE J. I. Clark Chair and Memorial University of Newfoundland, Canada.

Trung Nguyen, G. Mann and R. Gosine are with Intelligent Systems Lab, Faculty of Engineering and Applied Science, Memorial University of Newfoundland, St. John's, Newfoundland and Labrador, Canada, email: {trung.nguyen,gmann,rgosine}@mun.ca

papers, the work uses a single, off-the-shelf, forward-looking camera, odometry and the method does not require any camera calibrations. In addition, the theory is also easy to implement and requires less computation. For aerial vehicles, these advantages are beneficial. For applications of ground-aerial multi-vehicle systems [10], the quadrotor is tracked by a sensor system in the ground vehicle. When the quadrotor needs to exceed its working-area limit, it requires a navigation method for unassisted navigation. Therefore, the application of the proposed navigation technique is effective for indoor exploration missions and it can work without an external tracking system. Another scenario to apply the navigation technique is visual servoing and visual-teach-and-repeat (VT&R) in indoor working environment. The use of reference images navigates the quadrotor follow the desired path (Fig. 1). Our contributions in this paper have two aspects. Firstly, we propose a solution for the indoor path-following control of the quadrotor without the support of other external tracking system by utilizing a VT&R technique on Kanade-Lucas-Tomasi (KLT) corner features. Secondly, we implement Funnel Lane theory into the Ar.Drone quadrotor and propose a novel funnel-lane navigation calculation. Section II explains the Funnel Lane theory and how we build Funnel-Lane Visual Path based on features and reference images. Section III shows the funnel-lane navigation calculation. Section IV presents initial simulation results in ROS-Gazebo simulation and realtime experiments with Ar.Drone quadrotor.

II. FUNNEL-LANE VISUAL PATH

A. Funnel Lane Theory on One Landmark Feature

Generally, the pose of the UAV includes the position $\{x,y,z\}$ and orientation $\{\phi(Roll),\theta(Pitch),\psi(Yaw)\}$. The x and y elements are not available. The optical axis of the camera is parallel to the heading direction of the UAV. Funnel Lane theory [8] is applied for navigating the UAV from the current location (point C) with the pose $R_s^C = (x_s^C, y_s^C, z_s^C, \phi_s^C, \theta_s^C, \psi_s^C)$ to reach the next destination location (point D) with the pose $R_s^D = (x_s^D, y_s^D, z_s^D, \phi_s^D, \theta_s^D, \psi_s^D)$. The number s is the segment's order in the reference path. As shown in Fig. 2, the UAV at location D sees a landmark feature (red point) at point $u^D(u_X^D, u_Y^D)$ in the destination image plane I_s^D. At location C, the UAV sees a landmark feature (red point) at point $u^C(u_X^C, u_Y^C)$ in the current image plane I_s^C. A funnel lane is created by a fixed landmark corner feature is defined:

Definition 1: The funnel lane of a fixed landmark corner feature CF and a UAV at location point D is the set of locations $\mathbb{F}_{CF,D}$ such that for each $C \in \mathbb{F}_{CF,D}$, four funnel-lane constraints are satisfied:

$$|u_X^C| < |u_X^D| \qquad \text{(Constraint 1)}$$

$$\text{sign}(u_X^C) = \text{sign}(u_X^D) \qquad \text{(Constraint 2)}$$

$$|u_Y^C| < |u_Y^D| \qquad \text{(Constraint 3)}$$

$$\text{sign}(u_Y^C) = \text{sign}(u_Y^D) \qquad \text{(Constraint 4)}$$

Definition 2: The funnel lane of a fixed landmark feature CF, a UAV at location D, and a relative angle α is the set of locations $\mathbb{F}_{CF,D,\alpha} \subset \mathbb{F}_{CF,D}$ such that $\psi_s^C - \psi_s^D = \alpha$ for each $C \in \mathbb{F}_{CF,D,\alpha}$. A relative angle α is in the plane which is parallel with the X-Y plane, as in Fig. 3.

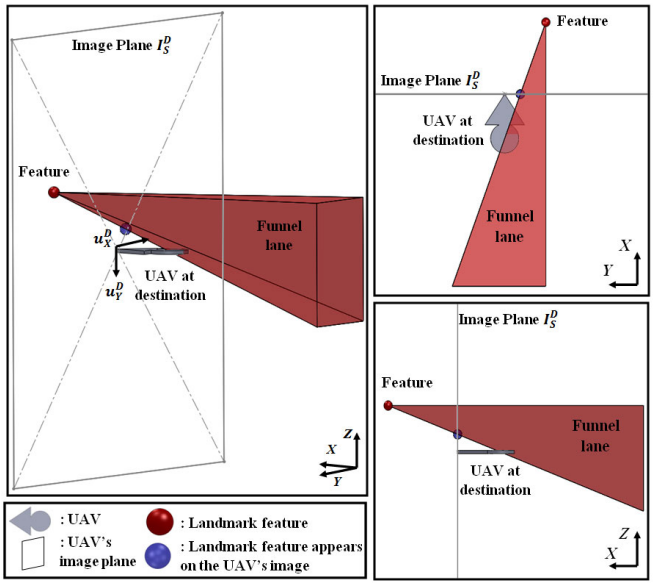


Fig. 2. Funnel lane is produced by a fixed landmark and UAV at destination location (point D) in the view of 3D (LEFT), Y-X view (RIGHT-TOP), and X-Z view (RIGHT-BOTTOM)

Funnel Lane theory defines a set of possible locations where the UAV is controlled to fly straight. In Fig. 2, the funnel lane is presented in 3D space as a red pyramid with respect to a red landmark corner feature or pyramid with a flat top with respect to the UAV at destination location (point D). A blue point in Fig. 2, Fig. 3 and Fig. 4 is the red landmark feature appears in the image plane of the UAV. The UAV at R_s^C, is presented in non-filled symbol, is navigated to reach R_s^D, presented in full-filled symbol. Notably, the origin of the coordinates axis (u^C or u^D) is assigned at the centre of the image plane as in Fig. 2. Fig. 3 presents the case of the UAV's same heading angle $(\psi_s^C - \psi_s^D = 0)$ at the current and destination location. Fig. 4 presents the case of the UAV's different heading angle $(\psi_s^C - \psi_s^D \neq 0)$. The funnel lane will rotate α angle as in *definition 2*. The UAV is equipped with a single forward-looking camera to detect the landmark corner features for navigation. In our case, the type of UAV is quadrotor and the kind of landmark corner features is KLT feature [11].

In reality, many landmark features are supposed to be detected in the UAV's image. Therefore, each feature produces its funnel lane. The intersection space between these funnel lanes satisfies both constraint conditions of all features. If the UAV is inside the intersection funnel lane, it will fly forward. Otherwise, the navigation is computed in such a way that the UAV will fly back to the intersection funnel lane. Another advantage given from this point is the reliability in navigation where in this case the UAV needs a minimum of only one landmark for navigation. The navigation depends

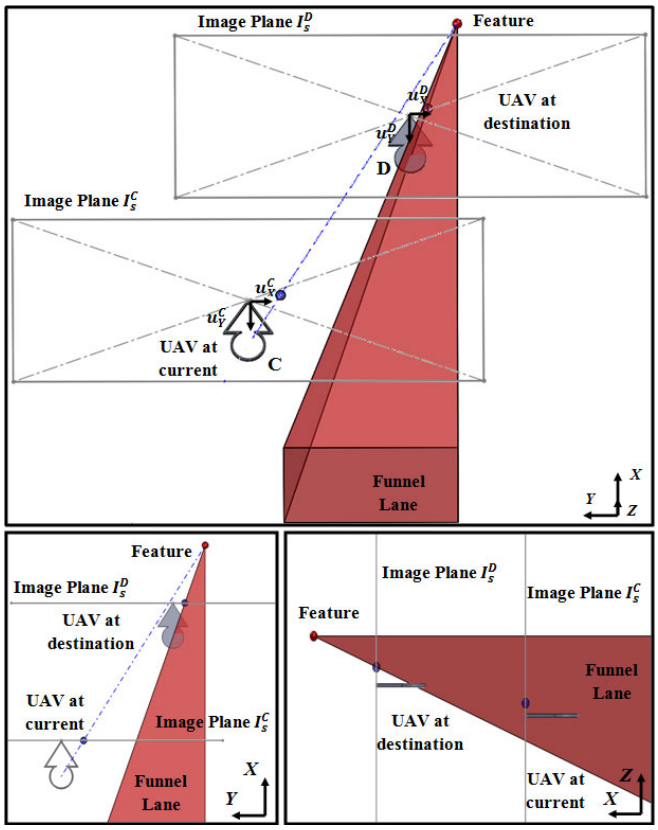


Fig. 3. The current location of UAV with respect to funnel lane. UAV at current location (point C) is navigated back to produced funnel lane in order to reach the destination location (point D)

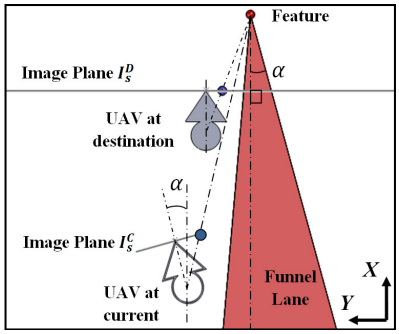


Fig. 4. Funnel lane is produced in case of different heading angles

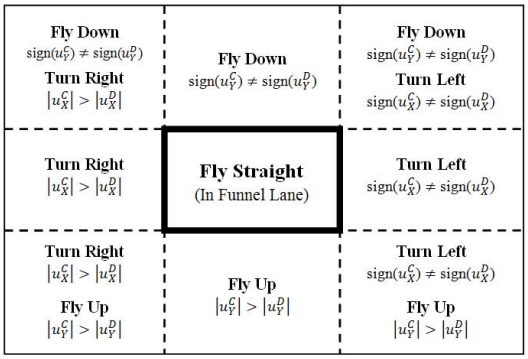


Fig. 5. Nine possible locations of UAV and the suitable performance of the UAV with respect to the intersection funnel lane

on the current location with respect to the intersection funnel lane. As a result, nine possible locations of the UAV with respect to the funnel lane are defined on the violation of four constraints and described in Fig. 5. In order to work with multiple features case, a funnel-lane navigation calculation is proposed in section III for the quadrotor type of UAV. After describing how the UAV from the current location reaches the destination location, the next section will depict a procedure to build and follow the funnel-lane visual path in experiments.

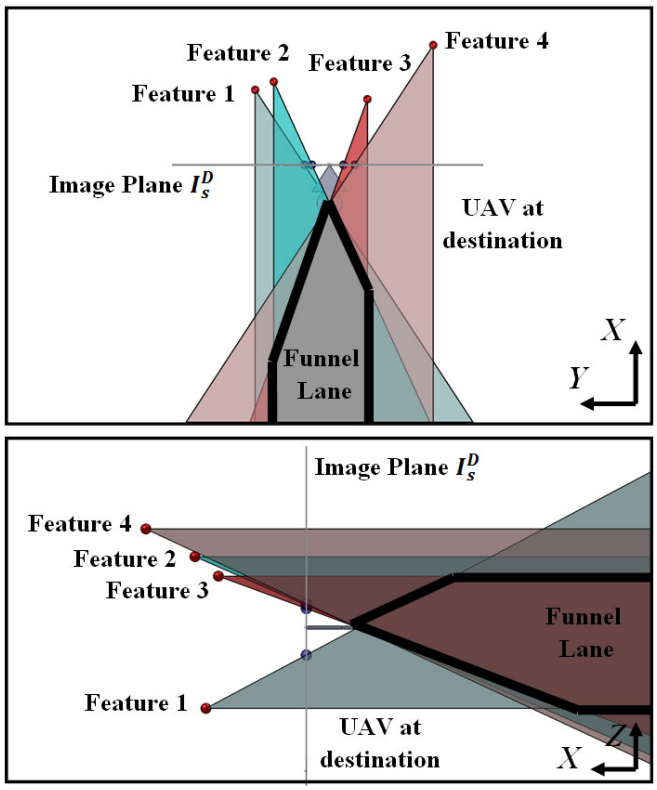


Fig. 6. Intersection funnel lane with four features in Y-X and Y-Z views

B. Reference Path Building

The reference path for the UAV is broken into several continuous segments $\{Seg_s | s \in \{1,2,...,n\}\}$. Reference images are collected at the beginning and at the end of each segment. We have a set of reference images at the start of the segment: $\{I_s^{Start} | s \in \{1,2,...,n\}\}$, a set of reference images at the end of the segment: $\{I_s^{End} | s \in \{1,2,...,n\}\}$, and a set of reference images at the current location of the segment: $\{I_s^C | s \in \{1,2,...,n\}\}$. In other words, the image at the end of the segment I_s^{End} is the image at destination location I_s^D in the Funnel Lane theory. The Funnel-lane visual path is defined by the following:

- *Hypothesis 1:* The UAV's coordinate frame is in the segment s at current location R_s^C and the task is to reach the destination location R_s^D. Two key images I_s^C and I_s^D are respectively associated with R_s^C and R_s^D. There always exists an acceptable path Ψ from R_s^C to R_s^D.

- *Hypothesis 2:* In the segment s, a set of matched points P_s between two key images I_s^C and I_s^D is observed along the path Ψ from R_s^C to R_s^D and allows the funnel-lane computation of the navigation. There exist nine possible locations of the UAV with respect to the intersection funnel lane formed by tracked points P_s, $\{P_s | s \in \{1,2,...,n\}\}$.
- *Hypothesis 3:* In the segment s, the condition to apply the Funnel Lane theory is that the transformation between R_s^C frame and R_s^D frame does not include the case which contains only lateral transformation.

Our framework is designed as VT&R system, two phases of experiment are proposed: learning phase and replay phase. The learning phase is to build the visual path. We uses the camera of the UAV in order to collect reference images at this phase, or reuse available images collected in the past by other vehicles with the condition that the same camera setup is kept or some minimal changes do not have great effect to the navigation. The replay phase is to perform path-following with funnel-lane navigation calculation, mentioned in section III. These phases are depicted in Fig.7. The reference path is divided into n segments and these segments are generated during the learning phase. The first image at the beginning of the learning phase is I_0^{Start}. Using Shi and Thomasi's [12] method, corner features are generated and tracked using non-pyramidal Lucas Kanade dense optical flow method. Once the tracked features fall below a threshold value, a new segment will be defined. The image at the end of the first segment is I_0^{End}. For each segment, we record I_s^{Start}, P_s^{Start}, I_s^{End} and P_s^{End}. The poses of UAV, R_s^{Start} and R_s^{End}, are also be recorded for funnel-lane navigation calculation. This procedure is repeated until the end of the reference path.

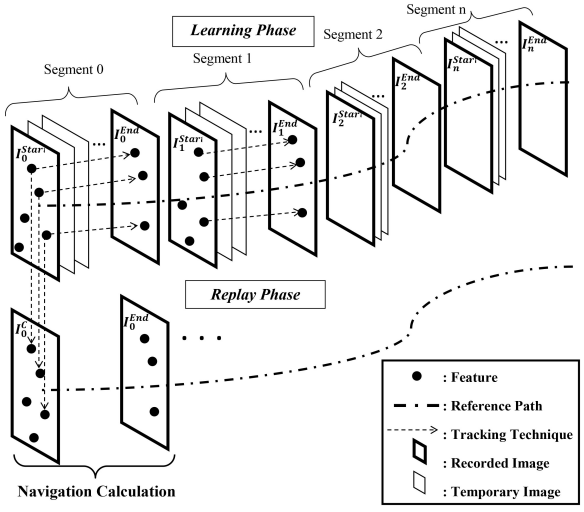


Fig. 7. Learning and replay phases of the framework

In the replay phase, the UAV uses the percent-matched-features (PM) method for initial localization. The best segment of the reference path has the highest ratio between a number of features tracked from the I_s^{Start} in the learning phase and a number of the recorded feature. If the UAV begins approximately at the same position as in the learning phase, the best segment should be the first segment. The first image of the best segment, I_0^{Start}, is used to track the feature in the current image, I_0^C, as in Fig. 7. Then these tracked features are compared with the reference features in the end image I_0^{End} to generate the control signal by the funnel-lane navigation calculation. The UAV automatically recognizes the end of the segment using the Mean Square Error (MSE) criteria. When the UAV is following the current segment, the MSE value between u^C and u^D tends to decrease and then starts to increase once it leaves the current segment. At this point, the UAV takes the next reference image for navigation.

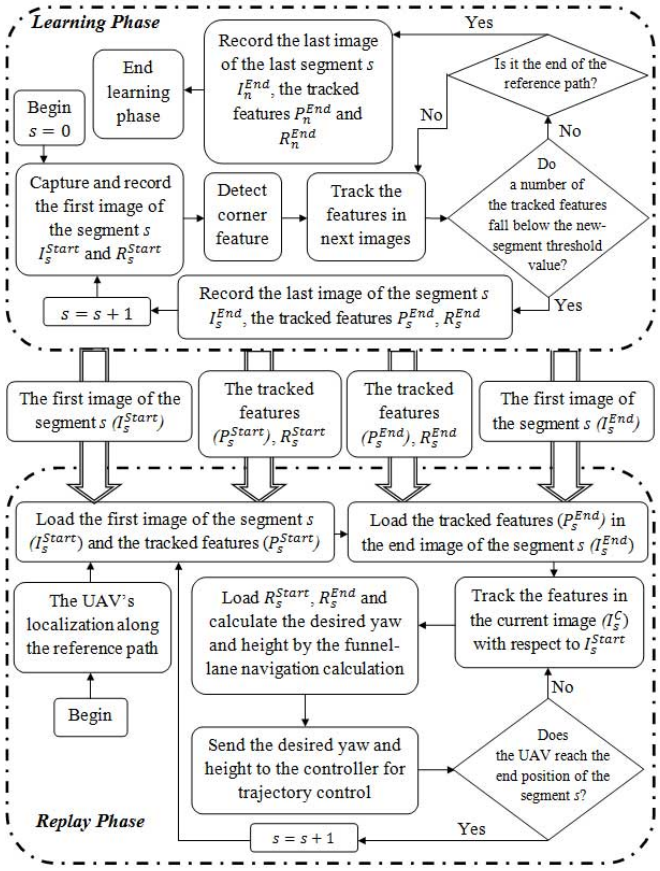


Fig. 8. The flow chart of learning and replay phases

III. FUNNEL-LANE NAVIGATION CALCULATION

A. The Estimation of Desired Heading

For each feature i, a signed distance to the line $u_X^C = u_X^D$ is calculated by the horizontal coordinate of the features:

$$f(u_X^C, u_X^D) = \frac{1}{\sqrt{2}}(u_X^C - u_X^D) \tag{1}$$

And a desired heading (yaw angle) of feature i is computed by the horizontal coordinate of the features:

$$\psi_d^{(i)} = \begin{cases} \gamma_1 \min\{u_X^C, f(u_X^C, u_X^D)\} & \text{if } u_X^C > 0 \text{ and } u_X^C > u_X^D \\ \gamma_1 \max\{u_X^C, f(u_X^C, u_X^D)\} & \text{if } u_X^C < 0 \text{ and } u_X^C < u_X^D \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Where γ_1 is the approximate conversion from pixels to degrees. Based on the desired heading computed for each feature i and the different heading of the UAV at the start and the end of the segment, the output result of desired heading is given by:

$$\psi_d = \eta_1 \frac{1}{N} \sum_{i=1}^{N} \psi_d^{(i)} + (1-\eta_1)\psi_0 \tag{3}$$

Where N is the total number of feature points used in the algorithm; ψ_0 is the desired heading obtained from the start and the end magnetometery heading measurements of the segment in the learning phase; η_1 $(0 \leq \eta_1 \leq 1)$ is the important relative between visual measurements versus magnetometery heading measurements.

B. The Estimation of Desired Height

For each feature i, a signed distance to the line $u_Y^C = u_Y^D$ is calculated by the vertical coordinate of the features:

$$f(u_Y^C, u_Y^D) = \frac{1}{\sqrt{2}}(u_Y^C - u_Y^D) \tag{4}$$

And a desired height (Z direction) is computed by the vertical coordinate of the features:

$$Z_d^{(i)} = \begin{cases} \gamma_2 \min\{u_Y^C, f(u_Y^C, u_Y^D)\} & \text{if } u_Y^C > 0 \text{ and } u_Y^C > u_Y^D \\ \gamma_2 \max\{u_Y^C, f(u_Y^C, u_Y^D)\} & \text{if } u_Y^C < 0 \text{ and } u_Y^C < u_Y^D \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Where γ_2 is the approximate conversion from pixels to meters. Based on the desired height computed for each feature i and the different height of the UAV between R_s^{Start} and R_s^{End} of the segment, the output result of desired height is given by:

$$Z_d = \eta_2 \frac{1}{N} \sum_{i=1}^{N} Z_d^{(i)} + (1-\eta_2)Z_0 \tag{6}$$

Where N is the total number of feature points used in the algorithm; Z_0 is the desired height obtained from the start and the end altimeter measurements of the segment in the learning phase; and $\eta_2(0 \leq \eta_2 \leq 1)$ is the important relative between visual measurements versus altimeter measurements.

IV. EXPERIMENTAL RESULTS

A simulation is conducted using Robot Operating System (ROS) Fuerte with Gazebo simulator running on Linux Ubuntu 12.04. Ar.Drone quadrotor model is produced in Gazebo simulator. The system architecture of simulation and realtime experiment are presented in Fig. 9 and Fig. 12. In the learning phase, the quadrotor is manually controlled along the reference path (blue dash-line) in order to perform learning phase. After that, the quadrotor will be placed approximately at the same start position of the learning phase. During replay phase, the funnel-lane navigation calculation is applied to perform path-following control. Realtime experiments are conducted in the Intelligent Systems Lab, Faculty of Engineering and Applied Science, Memorial University of Newfoundland. The front camera of the Ar.Drone quadrotor is HD camera 1280x720 pixels (720p) with 92^0 wide-angle. The calculation of navigation is performed off-board on a laptop then sent to the Ar.Drone through wireless communication. We do not consider the appearance of moving obstacles or obstacle avoidance.

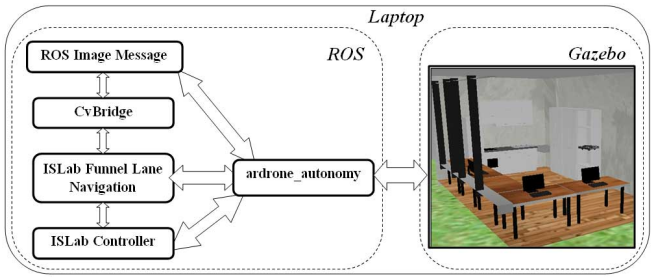
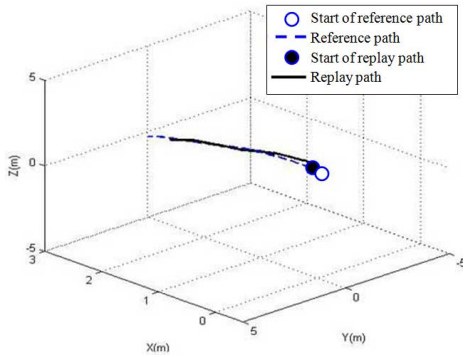
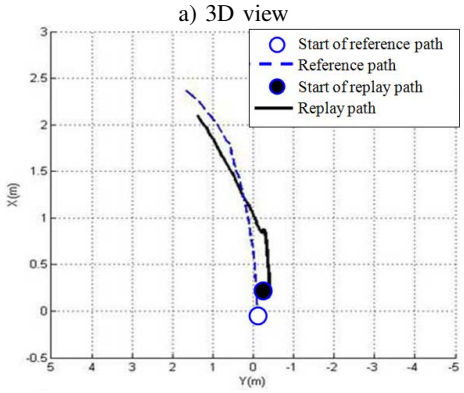


Fig. 9. The system architecture of simulation



a) 3D view

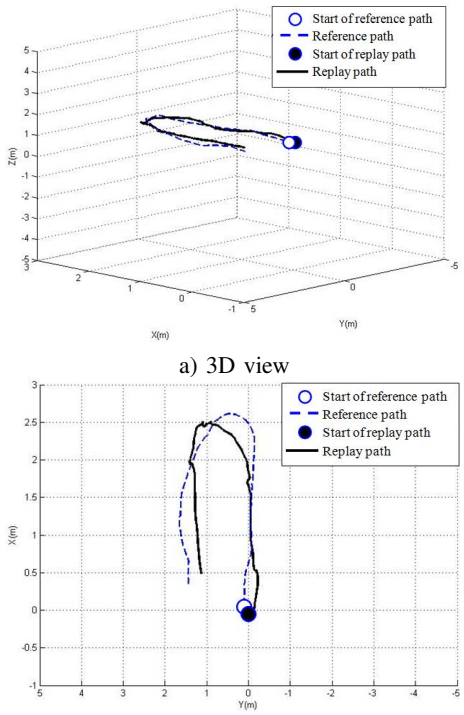


b) X-Y view

Fig. 10. The simulation result of tracking reference path 1

The simulation results (Fig.11 and Fig. 10) show the quadrotor performing path-following control with the reference path 1 (curve) and the reference path 2 (open-loop). In both two cases, the quadrotor is able to follow the trained reference path in the replay phase. During path following, the quadrotor compares the difference features' coordinates as Funnel Lane theory and performs navigation by the funnel-lane navigation calculation. Obviously, the quadrotor system works in the condition without the GPS information or any external tracking system (VICON). The position errors and the possibilities of failure are inevitable and generally caused by some following reasons. Firstly, the start position should

be as close to the original position in the learning phase. Secondly, the transit between segments is overestimated by MSE criteria. Thirdly, the control is based on PID control, which produces considerable overshoot and sensitive with noises from surrounding environment. The video of simulation can be located at *youtu.be/WVq9IttJx0g*. In the video, different types of the reference path are proposed to test the proposed navigation technique. The end of the video presents some initial results of realtime experiments. Especially, our proposed technique of navigation can handle the problem of accidental collision as showing in the video.



a) 3D view

b) X-Y view

Fig. 11. The simulation result of tracking reference path 2

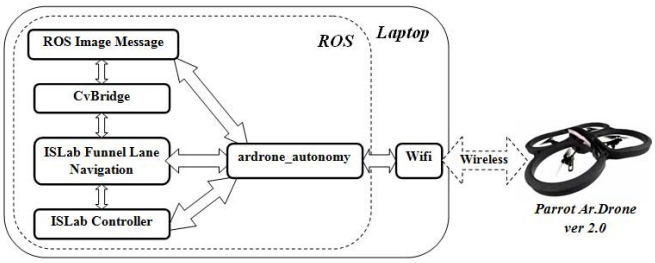


Fig. 12. The system architecture of realtime experiment

Besides the mentioned advantages, the navigation algorithm also shows some disadvantages, partly presented in [8]. The dynamic of UAV is more sensitive than that of the ground vehicle. Advanced control techniques will be studied to deal with the dynamic problem, which will be better than the simple control used in [8]. KLT feature used for navigation has some disadvantages of lighting conditions. We can assume that the lighting conditions are constant with indoor environment. Additionally, the tracking feature step between images can provide few incorrect results, which negatively affects to the navigation. However, the fusion between vision, heading measurement and altimeter measurement can keep the quadrotor at suitable direction despite the errors of tracking feature step.

V. CONCLUSION AND FUTURE WORK

This paper presents the implement of Funnel Lane theory into an Ar.Drone quadrotor aerial vehicle and extend the navigation technique into 3D case. With the funnel-lane navigation calculation, the quadrotor can work independently in indoor applications. Initial experimental results of simulation and realtime experiments are presented in the paper as well as the video. The application of the proposed technique is visual-homing in cooperation with the ground robot and visual-servoing in the path following mission. In future work, the paper will consider the problem of visual obstacle avoidance during path following and develop the segment localization. Other kind of feature will also be studied in order to improve the reliability of the system.

ACKNOWLEDGMENT

The authors thank Prof. Stan Birchfield, Department of Electrical and Computer Engineering, Clemson University, South Carolina, USA, for his time spent answering questions about Funnel Lane theory.

REFERENCES

[1] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The GRASP Multiple Micro-UAV Testbed," *IEEE Robotics Automation Magazine*, vol. 17, pp. 56–65, September 2010.

[2] F. Fraundorfer and L. Heng, "Vision-based autonomous mapping and exploration using a quadrotor MAV," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 4557–4564.

[3] C. Forster, M. Pizzoli, and D. Scaramuzza, "Air-ground localization and map augmentation using monocular dense reconstruction," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, November 2013, pp. 3971–3978.

[4] C. Bills, J. Chen, and A. Saxena, "Autonomous MAV flight in indoor environments using single image perspective cues," in *Proc. of IEEE International Conference on Robotics and Automation*, 2011.

[5] D. Lee, T. Ryan, and H. Kim, "Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing," *Proc. of IEEE International Conference on Robotics and Automation*, pp. 971–976, 2012.

[6] W. Li, T. Zhang, and K. Kuhnlenz, "A vision-guided autonomous quadrotor in an air-ground multi-robot system," in *Proc. of IEEE International Conference on Robotics and Automation*, 2011.

[7] J. Courbon, Y. Mezouar, N. Guenard, and P. Martinet, "Visual navigation of a quadrotor aerial vehicle," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 5315–5320.

[8] Z. Chen and S. Birchfield, "Qualitative vision-based path following," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 749–754, 2009.

[9] ——, "Qualitative vision-based mobile robot navigation," in *Proc. of IEEE International Conference on Robotics and Automation*, May 2006, pp. 2686–2692.

[10] O. De Silva, G. K. I. Mann, and R. G. Gosine, "Development of a relative localization scheme for ground-aerial multi-robot systems," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 870–875.

[11] J. Shi and C. Tomasi, "Good Features to Track," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

[12] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the openCV library*. O'Reilly Media, September 2008.