

Appearance-Based Visual-Teach-And-Repeat Navigation Technique for Micro Aerial Vehicle

Trung Nguyen  · George K. I. Mann ·
Raymond G. Gosine · Andrew Vardy

Received: 28 November 2014 / Accepted: 7 December 2015 / Published online: 15 February 2016
© Springer Science+Business Media Dordrecht 2016

Abstract The objective of this paper is to develop a vision-based navigation technique for micro aerial vehicles, quadrotor type, to operate in GPS-denied environment. The navigation method has been developed while using appearance-based *Visual-Teach-and-Repeat* (VT&R) technique. In a teaching phase, a quadrotor is manually navigated along a desired route to collect a set of reference images. In a repeating phase, the quadrotor is able to autonomously follow the desired route using these reference images. Self-localization is developed to determine the current segment of the desired route by a number of *Speeded-Up Robust Features* (SURF), matched between the current image and the reference images. In this paper, three methods of self-localization are proposed and

compared. After performing self-localization, the quadrotor computes appearance-based motion control commands (desired yaw and height) for the next movement in order to keep track of the desired route. This computation is developed on Funnel Lane theory, which was originally proposed in Chen and Birchfield (IEEE Trans. Robot. **25**(3), 749–754 (2009)) for 2D navigation of a ground vehicle. The paper extends this theory to 3D navigation of a quadrotor. The proposed self-localization methods are tested with several image databases. Finally, an online experiment of proposed VT&R technique is demonstrated using Ar.Drone quadrotor model.

Keywords Visual teach and repeat · Micro aerial vehicle · Self-localization · Visual homing · Visual servoing · Vision-based navigation

T. Nguyen (✉) · G. K. I. Mann · R. G. Gosine · A. Vardy
Intelligent Systems Lab, Faculty of Engineering
and Applied Science, Memorial University
of Newfoundland, A1B 3X5, St. John's, NL, Canada
e-mail: tn0432@mun.ca

G. K. I. Mann
e-mail: gmann@mun.ca

R. G. Gosine
e-mail: rgosine@mun.ca

A. Vardy
e-mail: av@mun.ca

1 Introduction

In recent years, *Micro Aerial Vehicles* (MAV), especially quadrotor type, have become one of the fastest developing technologies of unmanned vehicles. Although quadrotors have demonstrated potentials to replace humans performing difficult tasks [36], the widespread use of quadrotors in industry is still limited due to numerous navigation issues. The utilization of *Global Positioning System* (GPS) for localization

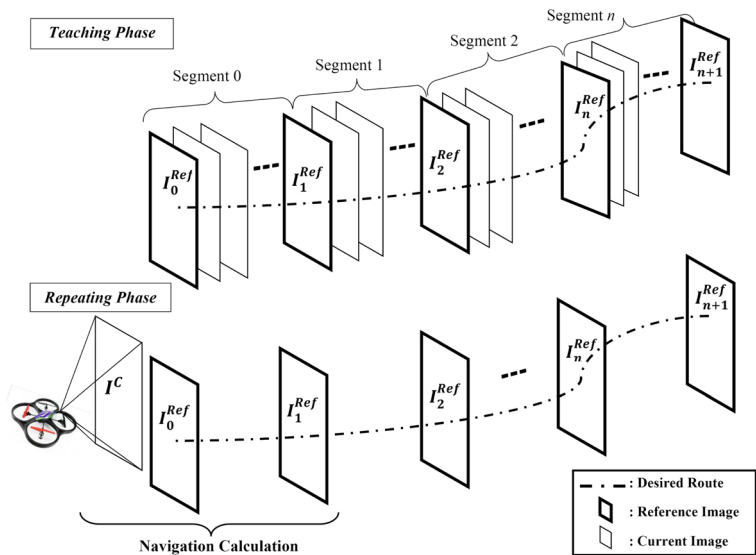
and control does not satisfy the necessary requirements of fully autonomous applications in several environments such as indoor and in urban canyons. Basically, if the initial position of the quadrotor is defined, the measurements of traveling speeds over elapsed time and course can reveal the current position of quadrotor [38, 53]. This dead-reckoning technique can produce acceptable results only for short-term operations due to experimental issues in cumulative errors and measuring drifts. Combining these measurements with other external measurements taken from a ground station [38] by an External Kalman Filter can improve the accuracy of localization. However, the ability of independent operations of quadrotor is reduced. One popular strategy is Simultaneous Localizing And Mapping (SLAM). The SLAM technique has been applied on quadrotor using either laser-based [3, 46] or vision-based [21, 22, 25]. Such implementations require considerable payload to carry exteroceptive sensors and demand higher computational cost. Running SLAM filters in computationally and hardware constrained systems is expensive and limits the applications for long term operations. Therefore, the practical applications of SLAM technique on quadrotor need the support from ground vehicle [20, 36] in order to perform the computations. Sharing computations among multiple vehicles is considered in the scheme of relative localization techniques [16–18].

Although these kinds of systems have demonstrated practical applications to investigate indoor environment, the working volume of quadrotor is limited to a restricted space defined by the maximum measured distance from ground vehicles.

To overcome the above limitations, *Visual Teach and Repeat* (VT&R) technique [4, 14, 24, 43, 50] is proposed as an effective solution to enable quadrotors to autonomously follow a desired route in GPS-denied environment. As implied in the name, VT&R technique consists of two phases (Fig. 1): teaching phase and repeating phase. In the teaching phase, a quadrotor is controlled by users (using joystick, keyboard, teleoperation) along a desired route to collect numerous reference images, which become the database of the desired route. In the repeating phase, the quadrotor will compare the current image with the reference images in order to produce appropriate motions.

This paper assesses the appearance-based monocular VT&R navigation technique including two modules: self-localization and motion control. The system is described in Fig. 2. In repeating phase, self-localization (or place recognition) first defines the current segment of quadrotor in order to load appropriate reference image. Three methods of self-localization are proposed to compare and validate with four databases, collected by the lead author and

Fig. 1 Two essential phases of VT&R quadrotor aerial system: teaching phase and repeating phase



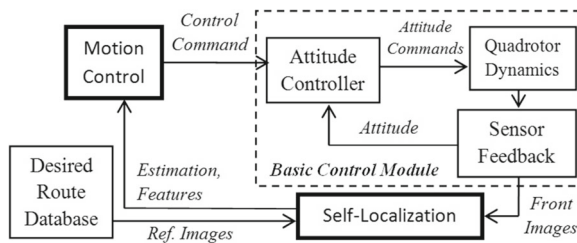


Fig. 2 Appearance-based VT&R system

other research groups (COLD database [44] and New College database [47]):

- Method I is a classical method, which uses matches of the SURF features in the current image with the SURF features in the reference images to compute the probability value of each segment in the desired route. The segment receiving the best probability value of matched features will be chosen as the current segment.
- Method II is an improvement of method I which incorporates feature-size relation with spatial distance for outlier rejection when matching SURF features.
- Method III is an adaptation of the self-localization method, proposed in [51], into monocular forward-looking camera.

Using one of these methods, the system will compare the current image with the reference image. Basing on the coordinates of matched features between these two images, motion control commands (desired yaw and height) are computed to guide the vehicle along the desired route. The computation of motion control command is developed on Funnel Lane theory, which was originally proposed in [11]. This paper extends it to 3D navigation for the quadrotor.

To summarize, this paper makes the following contributions. Firstly, this paper advances the-state-of-the-art in monocular self-localization of appearance-based VT&R aerial system by utilizing the feature-size relation with spatial distance. Secondly, Funnel Lane theory is adapted and expanded into 3D for computing the appropriate motion control commands for the next movement. Finally, the qualitative VT&R technique is firstly implemented on quadrotor aerial

vehicle and operating in *Robot Operating System* (ROS) [2].

The remainder of the paper is organized as follows. Section 2 introduces literature of VT&R systems. Section 3 describes unified notations in the paper as well as some properties of SURF feature and its matching-feature technique. Section 4 discusses Method I, Method II, Method III of self-localization. Section 5 presents the computation of motion control component built on Funnel Lane theory. Experimental results are presented in Section 6 and discussed in Section 7. Lastly, some conclusion and future work are presented in Section 8.

2 Related Works

The VT&R systems are classified by the calculating approach of navigation [51]: pose-based approach [4, 24, 43] and appearance-based approach [11, 14, 41].

The pose-based (or quantitative) approach reconstructs the positions of vehicle, detected landmarks and desired route in the same global coordinate frame basing on the current image and a set of reference images. Works, reported in [4, 14, 23, 24, 33, 42], are examples of the quantitative approach. In these works, the relative pose between the current image and the reference image is estimated based on matched features between these images. The shortest relative pose determines the appropriate reference image to be used for navigation. The initialization is performed using all reference images in order to define the initial position of the vehicle. This is a highly computationally expensive task. Hence, during route following, navigation is limited to use few reference images (2 or 3) for the transition between two successive segments [13, 14] in order to decrease the time of calculation. However, the ability to handle kidnapped-robot scenarios and recovery from considerable deviations from the desired route is reduced. Due to the computational cost, the calculation may be performed on a ground station (desktop) before sending the motion commands to quadrotor [13]. The limitations of the quadrotor hardware do not allow the onboard calculation. The pose-based approach requires calibrated cameras and scale factor estimation for metric localization within a segment. Although this approach is successful in

route-following, the quantitative approach still shows high computational cost and complicated implementation. These properties become disadvantageous for some specific systems, such as micro aerial vehicles, which possess strict constraint of computational cost or consider VT&R as secondary plan for visual homing.

The appearance-based (or qualitative) approach has shown considerable benefits in computational cost and convenient implementation. The approach does not use camera parameters or attempt to extract exact pose of the platform, rather it estimates the current segment where the quadrotor is operating based on the similarity between the current image and the reference images. This information is sufficient to qualitatively navigate the quadrotor along the desired route.

The work shown in [15], uses mutual entropy information of the similarity to perform the transition between two successive segments. This method can overcome the problems of occlusions. Nevertheless, the appearance of unexpected obstacles and the significant changes of environment negatively effect the accuracy. The method is applicable for 2D variation of camera rotation. Most of the other reported methods such as [10, 12, 19, 32, 51] use matched features to determine the current segment of the vehicle. These methods rely on reliable feature-matching results for navigation. Therefore, the development of the feature detection and matching techniques has received much attention as means of improving the performance of VT&R systems. Work, reported in [19], aims at optimizing the combination of feature detection methods and feature descriptor methods. Although this improves the performance of feature matching, the practical applicability of the method is questionable, due to the absence of the validation of the uncommon feature-detection methods used in different environments such as indoor scenarios. Work in [32] proposes to produce a virtual view of the current image using an image database, to improve the performance of feature-matching. This work reports validation of this strategy for outdoor scenarios by using the Google-Street-View database to generate the virtual view.

The works [6, 10, 40] propose to use a switching threshold for *Mean Square Error* (MSE) of feature coordinates in the images, which tends to decrease when the vehicle moves closely to the reference image. The simple method satisfies the requirement of

the transition without consuming much computation. As improved models of the transition computation, the references use Bayes filter [48], Kalman filter [52] or Markov filter [29]. Here, the estimation of self-localization highly depends on the estimation of the previous location and the accuracy of traveling measurements. The estimating errors, which occur in previous estimation, can not be fixed in the current estimation.

In work reported in [11], the probability computation of the transition between two successive segments consists of multiple information sources (i.e. matched features, traveled distance and heading angle). Accurate performances have been demonstrated with different routes using this method. Their navigation utilizes *Kanade-Lucas-Tomasi* (KLT) features. KLT technique are known to be sensitive with the ambient lighting, rotation and scale of the viewpoint. This method cannot be directly employed into quadrotor applications because the odometric measures used to locate the vehicle in the segment are highly unreliable. However, the study suggests that the use of multiple sources to support transition significantly improves the self-localization capability of vehicles. Additionally, the works in [10, 11] propose Funnel Lane theory in order to command a ground vehicle follow the desired route. The method first qualitatively defines possible positions, where the vehicle can navigate through, by the constraints of feature coordinates between the current image and the reference image. The generation of motion control commands is based on funnel-lane guided motion. If the vehicle locates itself outside the funnel lane, it will be commanded back to the funnel lane. Robust performance and inexpensive calculation motivate the paper to apply this strategy into quadrotor navigation. However, the calculation of motion control commands needs to be extended to 3D case in order to be appropriate for quadrotors.

The work in [51] self-localizes the vehicle on matched *Scale-Invariant Feature Transform* (SIFT) features of omnidirectional camera images. The SIFT feature-size relation with spatial distance is used in the computation of the location probability values. The approach produces very good results, and is compared with the method of using average percentage of matched features so as to validate its performance. The method is a viable solution to improve image transition in VT&R systems. It should be adopted to quadrotor systems such as the Ar.Drone model which

do not have the capability of omnidirectional visual perception. The method, reported in [51], motivates this paper to apply the feature-size relation with spatial distance for improving self-localization of VT&R navigation technique on quadrotor aerial vehicle.

3 Descriptions of VT&R System

3.1 Configurations

The VT&R aerial system is equipped with monocular forward-looking camera that can be found in numerous other commercial and research quadrotor models [1, 19, 28]. Each reference image represents one segment of the desired route as in Fig. 1. Some notations used in the paper are presented as follows.

- Ψ is the desired route of VT&R system.
- $\{Seg_s | s \in \{1, 2, \dots, n\}\}$ are several continuous segments constructing the desired route Ψ .
- I_s^{Ref} is the reference image of the segment s in the desired route, $\{I_s^{Ref} | s \in \{1, 2, \dots, n + 1\}\}$.
- I_k^C is the k^{th} image feedback in repeating phase.
- I^C is the current image feedback in repeating phase.
- $F_{s,e}^{Ref}$ is e^{th} feature detected in I_s^{Ref} .
- F_f^C is f^{th} feature detected in I_k^C .
- $F_{s+1,g}^{Ref}$ is g^{th} feature detected in I_{s+1}^{Ref} .
- P_s^{Ref} is the position of MAV where taking I_s^{Ref} .
- P^C is the current position of MAV for I_k^C .
- P_{s+1}^{Ref} is the position of MAV where taking I_{s+1}^{Ref} .

The VT&R system is built on the observation of visual landmarks in the working environment. These visual landmarks appear as interest points (or features) on the image plane. These features are detected and matched between the current image and the reference images. In order to receive useful features, the landmark should contain following properties:

- **Being stationary and repeatable in the working environment for both teaching phase and repeating phase:** Landmarks on moving objects such as other vehicles and humans should not become observed features. It is complicated to navigate basing on these dynamic features. The adapted feature matching technique has ability to reject the unwanted features of moving objects.

Additionally, the features should appear in both phases of VT&R technique.

- **Being robust in case of different lighting and visual noise:** The changes of working environment between the teaching phase and the repeating phase are inevitable. When considering the changes of ambient lighting and the effects of noise, the landmark should be still robust enough for the vehicle to detect and to match its features.
- **Being distinctive in the working environment:** Similar landmarks in working environment can produce inappropriate matched features, which negatively affect the calculation of self-localization.

3.2 Feature Detection and Description

Scale- and rotation-invariant feature detectors, such as Speeded-Up Robust Features (SURF) [5] and Scale-Invariant Feature Transform (SIFT) [30] features, have recently demonstrated their useful applications in computer vision as well as in robotics. Generally, scale-invariant feature detection performs the scale-space processing through image pyramid to reach the scale-invariant property of features. The scale-space consists of many octaves, where an octave contains many levels according to the increasing or decreasing of the scale σ values. With multiple values of scaling σ parameters, Laplacian of Gaussian method (LoG), a scale-space filtering, is performed for different octaves to define local maxima across scale and space. In order to cope with the high computational cost associated with LoG, SIFT feature detection approximates LoG with Difference of Gaussians

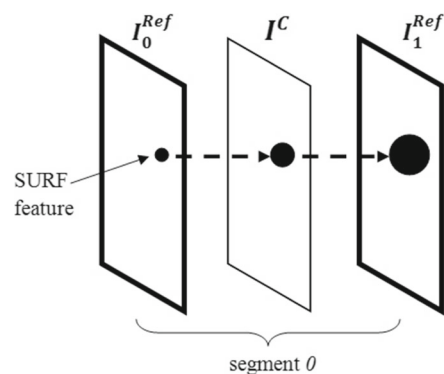


Fig. 3 The SURF feature size relation with spatial distance

Fig. 4 Test the feature-size relation on images from New College Database. *LEFT: I_s^{Ref} , RIGHT: I^C*



(DoG). And then these local maxima are defined by comparing one pixel in an image with its 8 neighbors, 9 pixels in the next level and 9 pixels in the previous level. These defined local maxima need to pass an evaluation step to become scale-invariant features. As a result, the approach of SIFT feature detection still requires large savings in memory to store the entire image pyramid. The advent of SURF features is partly inspired by the requirement of a speeded-up version for SIFT features. The work in [5] use Hessian box filter to approximate LoG in the support of integral images. More details about SURF and SIFT features can be found in the work, presented in [5] and [30] respectively. In the view of VT&R system design, the scale- and rotation-invariant properties of SURF features provide considerable benefits as quadrotor maneuvers in 3D and is subjected to image noise and viewpoint disparity.

Besides these advantages, another noticeable property of scale-invariant feature is the relation between spatial distance and the feature size (or scale). When the vehicle moves closely to the landmarks, the size of the landmark features tends to increase. For example, the position snapshot of I^C locates between the position snapshot of I_0^{Ref} and the position snapshot of I_1^{Ref} as in Figs. 3 and 4. Matched features of I^C should have larger size than the features of I_0^{Ref} but smaller size than the features of I_1^{Ref} .

The work [51] proposes to use SIFT features for the VT&R ground system. Although SIFT and SURF features apply the same principals, SURF features show faster calculation than SIFT features, and still can

satisfy the accuracy and stability for self-localization of the quadrotor. The scale-space representation of the SURF features [5, 37, 39, 49] is approximately estimated to decrease the average calculational time. Therefore, the authors have chosen SURF features for the VT&R aerial system. After detecting SURF features in the current image and the reference images, feature matching can be performed by SURF feature descriptors. However, the descriptor of SURF features is still calculated in the form of floating-point numbers forming a 64-dim descriptor vector. As a result, the memory footprint of one SURF descriptor requires at least 256 bytes. Considering computational capabilities of the embedded systems on the quadrotor, the use of SURF descriptor will take considerable memory for thousands or hundreds of features. Hence, the paper proposes to use Binary Robust Independent Elementary Feature (BRIEF) descriptor in order to improve the efficient computation of matching feature step [8, 26]. BRIEF descriptor is one of many binary descriptors, which considering the statistical properties of image region around the detected feature to perform feature matching. BRIEF descriptor has shown fast computation and compact presentation.

4 Appearance-Based Self-Localization

Self-localization is performed by processing the current image feedback and the reference images to determine the current segment. Three methods of self-localization are proposed as follows:

Fig. 5 Results of matching features after matching feature descriptor step

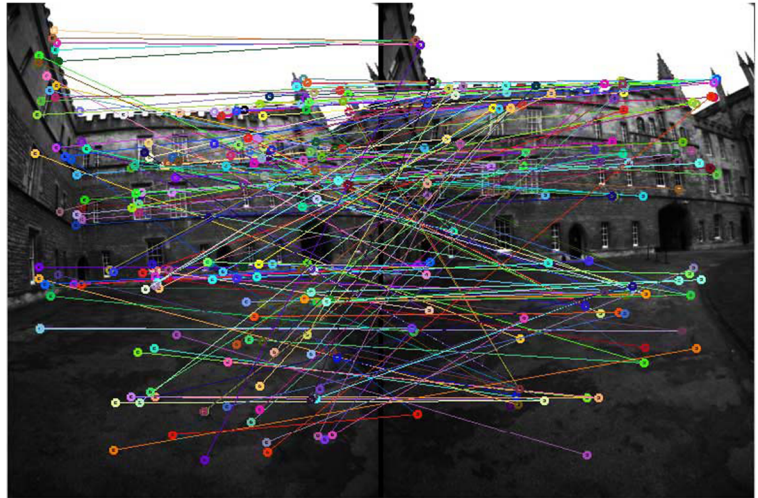


Fig. 6 Results of matching features after eliminating incorrect matched features by its distance

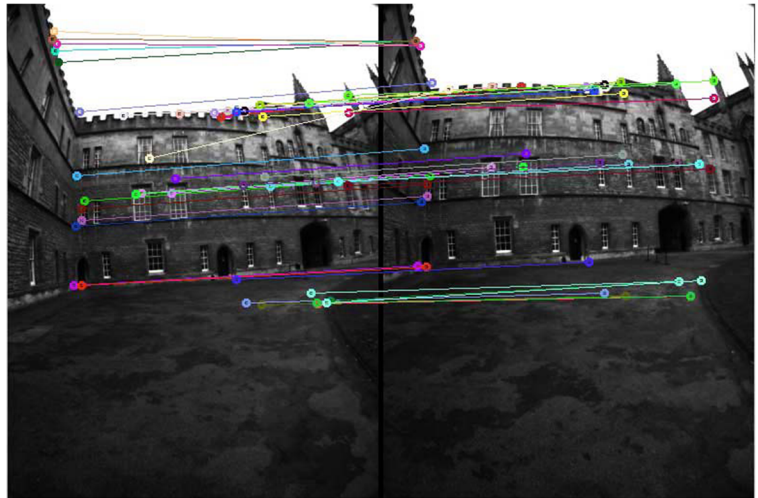


Fig. 7 Results of matching features after utilizing feature-size relation with spatial distance to filter incorrect features

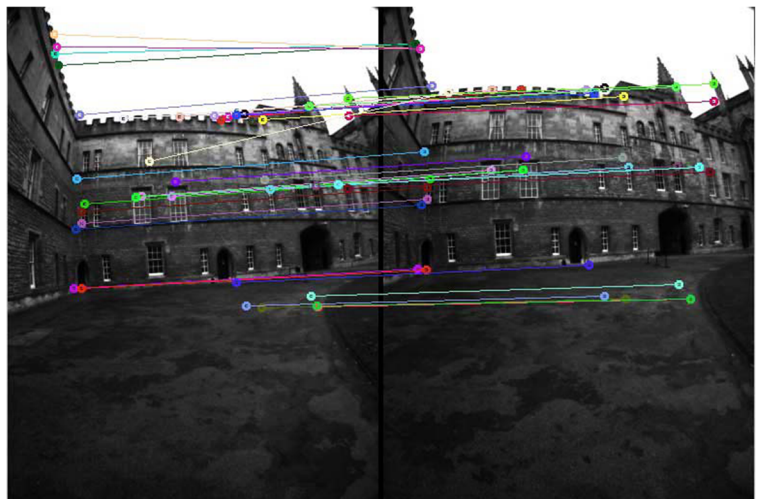
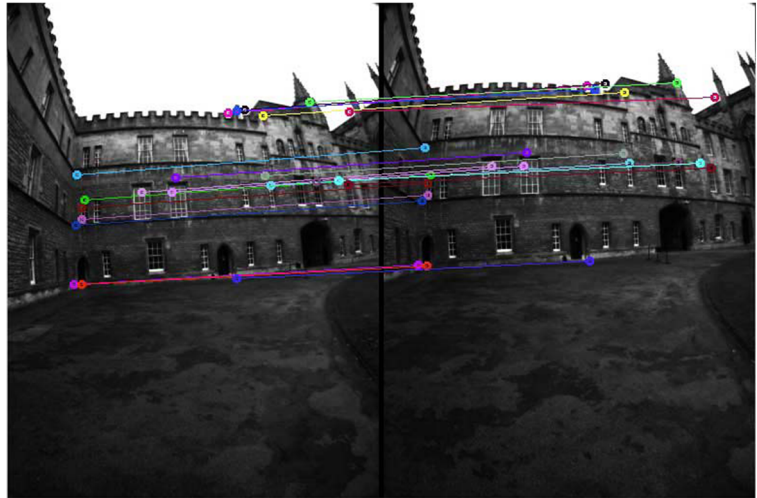


Fig. 8 Results of matching features after applying RANSAC for feature filtering



4.1 Method I

Normally, self-localization is performed by comparing the number of matched features between the current image and the reference images [19, 32, 51]. After matching SURF features between images by its descriptors, the results contain considerable incorrect matches. Some methods of outlier rejection are employed to provide reliable matched features for self-localization. One filtering method is RANdom SAMple Consensus (RANSAC), which is an iterative method that fits a model to filtered data [9]. The model in this case is a homography between two images. Incorrect features, which will not fit the model, are eliminated. Our first feature matching algorithm (method I) is presented below:

Algorithm 1 Matching features of method I between $\{F_f^C\}$ of I^C and $\{F_{s,e}^{Ref}\}$ of I_s^{Ref}

- Input:** $\{F_f^C\}, \{F_{s,e}^{Ref}\}, I^C$ and I_s^{Ref}
 - Output:** Matching SURF feature result: $\{F_s^{M,3*}\}$
 - 1 Calculate feature descriptors: $\{F_f^C.descriptor\}$ and $\{F_{s,e}^{Ref}.descriptor\}$
 - 2 Match feature descriptors between $\{F_f^C.descriptor\}$ and $\{F_{s,e}^{Ref}.descriptor\}$ in order to produce $\{F_s^{M,1}\}$
 - 3 Eliminate incorrect matched results by its distance (in feature space) to have $\{F_s^{M,2}\}$: $\{F_s^{M,1}.distance\} > 0.5 * Mean(\{F_s^{M,1}.distance\})$
 - 4 RANSAC feature filtering of $\{F_s^{M,2}\}$ to have $\{F_s^{M,3*}\}$
-

$\{F_f^C\}$ SURF features of I^C are matched with $\{F_{s,e}^{Ref}\}$ of I_s^{Ref} to have $\{F_s^{M,3*}\}$. $\{F_f^C\}$ SURF features of I^C are matched with $\{F_{s+1,g}^{Ref}\}$ of I_{s+1}^{Ref} to have $\{F_{s+1}^{M,3*}\}$ also by the Algorithm 1. The probability calculation of self-localization is performed by taking the average of percentage of matched features (Eq. 1). This information is used to estimate the segment where the vehicle is locating. If the quadrotor is currently at segment s , it will have more matched features $\{F_s^{M,3*}\}$ and $\{F_{s+1}^{M,3*}\}$. The segment receiving the maximum value of $MethodI\{Seg_s\}$ is taken as the vehicle’s true location.

$$MethodI\{Seg_s\} = \frac{1}{2} \left(\frac{|\{F_s^{M,3*}\}|}{|\{F_{s,e}^{Ref}\}|} + \frac{|\{F_{s+1}^{M,3*}\}|}{|\{F_{s+1,g}^{Ref}\}|} \right) \tag{1}$$

4.2 Method II

Method II intends to improve method I while using additional condition of the relation between spatial distance and the size of feature. Some incorrect features are eliminated by using feature-size relation. It means that the $\{F_f^C\}$ has larger size than $\{F_{s,e}^{Ref}\}$ but smaller size than $\{F_{s+1,g}^{Ref}\}$ as in Fig. 3. Reasonably, $\{F_{s,e}^{Ref}\}$ and $\{F_{s+1,g}^{Ref}\}$ are detected at the start position and the end position of the segment. Any features of $\{F_f^C\}$ should have the value of size between these

constraints. Matching SURF feature of method II is presented as in Algorithm 2.

Algorithm 2 Matching features of method II

Input: $\{F_f^C\}$, $\{F_{s,e}^{Ref}\}$, $\{F_{s+1,g}^{Ref}\}$, I^C , I_s^{Ref} and I_{s+1}^{Ref}
Output: Matching SURF feature result: $\{F_s^{M,4}\}$, $\{F_{s+1}^{M,4}\}$

- 1 Calculate descriptors: $\{F_f^C.descriptor\}$ and $\{F_{s,e}^{Ref}.descriptor\}$;
- 2 Match feature descriptors between $\{F_f^C.descriptor\}$ and $\{F_{s,e}^{Ref}.descriptor\}$ in order to produce $\{F_s^{M,1}\}$;
- 3 Match feature descriptors between $\{F_f^C.descriptor\}$ and $\{F_{s+1,g}^{Ref}.descriptor\}$ in order to produce $\{F_{s+1}^{M,1}\}$;
- 4 Eliminate incorrect matched results by its distance (in feature space) to have $\{F_s^{M,2}\}$ and $\{F_{s+1}^{M,2}\}$:
 $\{F_s^{M,1}.distance\} > 0.5 * Mean(\{F_s^{M,1}.distance\})$
 $\{F_{s+1}^{M,1}.distance\} > 0.5 * Mean(\{F_{s+1}^{M,1}.distance\})$;
- 5 Check the feature-size relation with spatial distance to have $\{F_s^{M,3}\}$ and $\{F_{s+1}^{M,3}\}$;
- 6 **if** matching between $\{F_f^C\}$ and $\{F_{s,e}^{Ref}\}$ **then**
- 7 | $\{F_{s,j}^{M,2}.size\} > \{F_{s,e}^{Ref}.size\}$;
- 8 **end**
- 9 **if** matching between $\{F_f^C\}$ and $\{F_{s+1,g}^{Ref}\}$ **then**
- 10 | $\{F_{s+1,j}^{M,2}.size\} < \{F_{s+1,g}^{Ref}.size\}$;
- 11 **end**
- 12 RANSAC feature filtering of $\{F_s^{M,3}\}$ to have $\{F_s^{M,4}\}$;
- 13 RANSAC feature filtering of $\{F_{s+1}^{M,3}\}$ to have $\{F_{s+1}^{M,4}\}$;

$\{F_f^C\}$ SURF features of I^C are matched with $\{F_{s,e}^{Ref}\}$ of I_s^{Ref} to have $\{F_s^{M,4}\}$. $\{F_f^C\}$ SURF features of I^C are matched with $\{F_{s+1,g}^{Ref}\}$ of I_{s+1}^{Ref} to have $\{F_{s+1}^{M,4}\}$. Then, the probability calculation of self-localization is performed by Eq. 2 in order to estimate the segment of vehicle location. The segment receiving the maximum value is taken as the vehicle’s true location.

$$MethodII\{Seg_s\} = \frac{1}{2} \left(\frac{|\{F_s^{M,4}\}|}{|\{F_{s,e}^{Ref}\}|} + \frac{|\{F_{s+1}^{M,4}\}|}{|\{F_{s+1,g}^{Ref}\}|} \right) \quad (2)$$

Results of matching SURF feature step-by-step is presented in Figs. 5, 6, 7, 8. The left image is I^C while the right image is I_{s+1}^{Ref} .

4.3 Method III

Method III is an adaptation of self-localization method, proposed in the work [51], into monocular camera. Method III reuses the SURF feature matching technique in method II (Algorithm 2). Segment estimation is calculated by Bayes’s rule. Basically, in order to perform self-localization, two dependent

events are considered: number of matched features, $\{F^M\}$, and segment estimation, $\{Seg_s\}$. The conditional probability of their relationship obeys Bayes’s rule [45]:

$$p\{Seg_s|\{F^M\}\} = \frac{p(\{F^M\}|\{Seg_s\})p\{Seg_s\}}{p(\{F^M\})} \quad (3)$$

$$p\{Seg_s|\{F^M\}\} \propto p(\{F^M\}|\{Seg_s\})p\{Seg_s\} \quad (4)$$

$p\{Seg_s\}$ presents the belief of specific segment s . This information is available if the quadrotor global position during localization is provided. As the VT&R system’s configuration excludes global positioning system, $p\{Seg_s\}$ is set to be the same for every segment. Additionally, $p(\{F^M\})$ is eliminated since the maximum value of $p\{Seg_s|\{F^M\}\}$ is considered. $p(\{F^M\}|\{Seg_s\})$ is calculated by assuming the quadrotor is at segment s of the desired route between two reference images I_s^{Ref} and I_{s+1}^{Ref} in order to define $p\{Seg_s|\{F^M\}\}$. The estimations basing on matched features, $p(\{F^M\}|\{Seg_{s>i-1}\})$ and $p(\{F^M\}|\{Seg_{s<i+1}\})$ are two independent events.

$$p(\{F^M\}|\{Seg_{s=i}\}) = \frac{p(\{F^M\}|\{Seg_{s>i-1}\})}{p(\{F^M\}|\{Seg_{s<i+1}\})} \quad (5)$$

$\{F_f^C\}$ are matched with $\{F_{s,e}^{Ref}\}$ to produce matched features $\{F_s^{M,4}\}$. $\{F_f^C\}$ are matched with $\{F_{s+1,g}^{Ref}\}$ to produce matched features $\{F_{s+1}^{M,4}\}$. When matching SURF features, the feature-size relation is added in considering the $\{I_s^{Ref}\}$ or $\{I_{s+1}^{Ref}\}$. Probability calculation of $p(\{F_s^M\}|\{Seg_{s>i-1}\})$ and $p(\{F_{s+1}^M\}|\{Seg_{s<i+1}\})$ are computed:

$$p(\{F^M\}|\{Seg_{s>i-1}\}) = \frac{|\{F_s^{M,4}\}|}{|\{F_{s,e}^{Ref}\}|} \quad (6)$$

$$p(\{F^M\}|\{Seg_{s<i+1}\}) = \frac{|\{F_{s+1}^{M,4}\}|}{|\{F_{s+1,g}^{Ref}\}|} \quad (7)$$

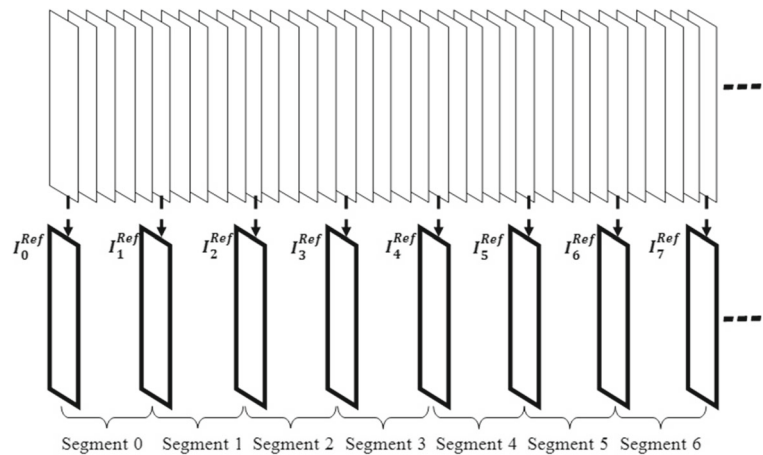
As a result, $p\{Seg_s|\{F^M\}\}$ is defined as Eq. 8.

$$p(\{F^M\}|\{Seg_{s=i}\}) = \frac{|\{F_s^{M,4}\}|}{|\{F_{s,e}^{Ref}\}|} \frac{|\{F_{s+1}^{M,4}\}|}{|\{F_{s+1,g}^{Ref}\}|} \quad (8)$$

$$MethodIII\{Seg_s\} = p(\{F^M\}|\{Seg_{s=i}\}) \quad (9)$$

The segment receiving maximum percent value of estimation $MethodIII\{Seg_s\}$ will provide data as reference image for navigation.

Fig. 9 Processing database to choose reference images and current images for testing



4.4 Experimental Validations of Self-Localization

4.4.1 Databases for Validations

Three proposed methods of self-localization are validated and analyzed with four databases. Two databases are collected by the authors while other two databases are collected by other research groups.

- **Route A** (Fig. 25): Second Floor of Faculty of Engineering and Applied Science including, Memorial University of Newfoundland (MUN) includes Engineering Lobby, Cafeteria and Engineering Lounge, 61 images are collected at 11:00 am May 17, 2014, cloudy weather. Image dimensions: 2592x1936, environment: indoor.
- **Route B** (Fig. 26): Outside Engineering Building, MUN, along Kerwin PI road, 72 images are collected at 3:00 pm May 17, 2014, in sunny weather. Image dimensions: 2592x1936, environment: outdoor.
- **CoSy Localization Database** [44]: Images are collected by ActivMedia PeopleBot platform. Image details are listed, dimensions: 640x480, environment: indoor office, year: 2008, version: COLD-Saarbrucken - Part B - night condition. Robot travels with the speed 0.220 m/s.
- **New College Database** [47]: The images of New College Database are collected by Segway robotic platform in Epoch A Campus at New College, Oxford, United Kingdom. The working

environment is outdoor with sunny weather in May 2009. Image details are listed, dimensions: 384x512 pixels, environment: outdoor.

4.4.2 Results of Testing Self-Localization Technique

Tested databases are collected when the vehicle is moving. Therefore, the images of databases show the motion blur in their content. When processing each database, some images are chosen as reference images, and the other images between chosen reference images are considered as current images and used to test the performance of self-localization technique (Fig. 9). The reference images are chosen so that the different rate measurement between the current image and the reference image is sufficient to achieve self-localization, where the distance of the segment is at-least 1m. The number of current images between two reference images is not similar depending on the different rate measurement. More current images provide less reference images, more advantages of storing reference images, and faster self-localizing operation. Three methods will be tested in using the same databases and the set of m tested images $\{I_k^C\}$ in order to have similarly operating conditions.

The results of estimating location are compared with the employed ground-truth data in order to define the percentage of success in each of the method. Additionally, entropy measurement is calculated as in Eq. 10 to measure the uniformity of the probability distribution in three proposed methods [51]. In

comparison, the method performing better has lower entropy value.

$$Entropy(\{I_k^C\}) = - \sum_{k=0}^m \text{sim}(I_k^C) \log_2 \text{sim}(I_k^C) \quad (10)$$

where sim is an output function calculated by method I or method II or method III.

In addition, the paper also observes the different rate between the current image and the reference images. The difference rate is computed in the form of correlation norm matching as Eq. 11 [7]. A value of D , closer to 1, represents a small difference or a good match, while a small D , closer to 0, is significantly different or a poor match. R , being reference image, and C , being current image, have the same size $w \times h$; $x' = 0 \dots (w - 1)$ and $y' = 0 \dots (h - 1)$.

$$D = \frac{\sum_{x',y'} R(x', y') C(x + x', y + y')}{\sqrt{\sum_{x',y'} R(x', y')^2 \sum_{x',y'} C(x + x', y + y')^2}} \quad (11)$$

Table 1 presents the results of self-localization by all three methods. Method II and method III provide better in estimating the current segment than method I. The use of the SURF feature size relation with spatial distance has eliminated some incorrect features to improve the performance of self-localization. Method II has smallest failure numbers over trials. However, the estimating accuracy of method II is not much different as compared to method III, and suffers higher entropy evaluation. In other words, the estimation of method III produces very small errors due to Bayes'

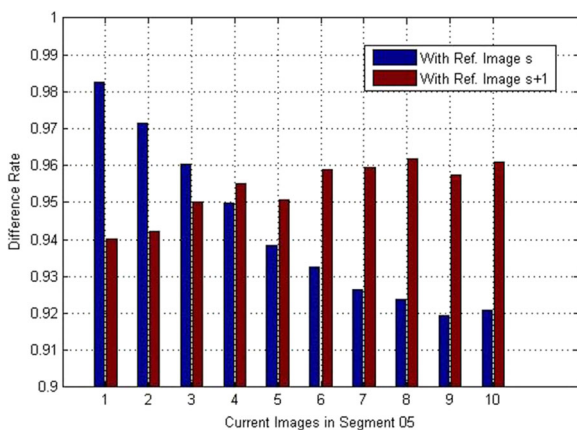


Fig. 10 Results of difference rate measurement in Segment 5 - New College database

rule. For example, if the quadrotor is truly in segment 5, the errors of method III is ± 1 segment (segment 4 or 6) while those of methods I and II are bigger (segment 1 or 9). Additionally, methods II and III have taken less processing time than method I because the elimination steps have removed incorrect matched features before applying RANSAC feature filtering.

Reference images (I_5^{Ref} , I_6^{Ref}) and a set of current image feedback (I_k^C) in segment 5 of New College database (Fig. 24) are chosen to evaluate the performance of self-localization methods within one segment. Figure 10 expresses difference rate between current images (I_k^C) and reference images (I_s^{Ref}). Figures 11 and 12 express the results of estimation and entropy on the current images of segment 5 in New College database (Fig. 24). When the vehicle is at position 1 (Current Image 1), closest to the reference image I_s^{Ref} , the value of the estimation reaches the highest value. When the vehicle moves forward, far from P_s^{Ref} and I_s^{Ref} , the value of the estimation tends to decrease. When the vehicle moves closed to P_{s+1}^{Ref} and I_{s+1}^{Ref} the value of the estimation tends to increase. Figure 10 can show the changes of the different rate when the vehicle is moving from P_s^{Ref} to P_{s+1}^{Ref} . The property happens to be the same with the entropy diagram. These results are different from those of the work [51], where the centre position receives the maximum estimation. The difference is caused by the type of camera used in the system. The work [51] uses camera with hyperbolic mirror providing front and back

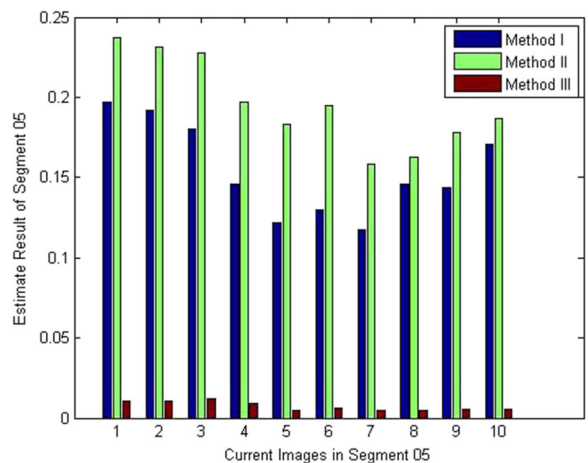


Fig. 11 Estimation results of Segment 5 - New College database. Estimation results are the calculation of $MethodI\{Seg_5\}$, $MethodII\{Seg_5\}$ and $MethodIII\{Seg_5\}$

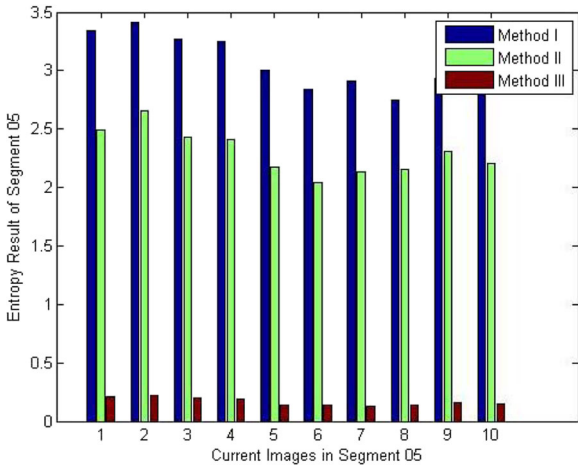


Fig. 12 Entropy results of three methods using Segment 5 - New College database

images, while this paper uses a monocular camera providing front images.

As a result, method III of self-localization is preferred to integrate with appearance-based motion control in VT&R aerial system.

5 Appearance-Based Motion Control

Self-localization determines the MAV’s true location, current segment of the desired route. After that,

motion control module needs to generate desired yaw and desired height commands for the MAV to reach the end position of the segment. Appearance-based motion control is developed on Funnel Lane theory.

5.1 The Theory of Funnel Lane

Funnel Lane theory was originally developed for 2D navigation by Z. Chen and S. Birchfield [11] in 2009 in order to navigate a ground vehicle to follow the desired route. The method first qualitatively defines possible positions where the vehicle can go straight by the constraints of feature coordinates between the current image and the reference image. Navigation is based on funnel-lane guided motion. If the vehicle is outside the funnel lane, it will be navigated back to the funnel lane. In this paper, Funnel Lane theory is extended to 3D navigation for MAV by considering the vehicle altitude (Z-direction).

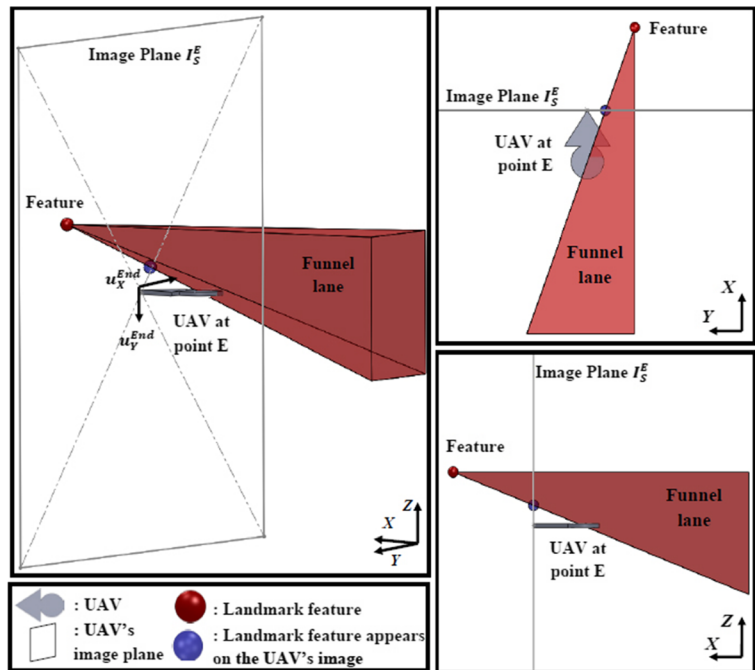
Applying Funnel Lane theory for the case of one fixed landmark (FL), MAV, which is locating at the current position, (P^C - point C), wants to reach the end of the segment s , (P^E - point E, $P^E = P_{s+1}^{Ref}$) as in Fig. 14. Notably, the origin of feature coordinates is assigned at the centre of the image plane, while the optical axis of the camera is parallel to the heading direction of the MAV. As shown in Fig. 13, the MAV

Table 1 Experiment results with multiple databases for validating three proposed methods

		Method I	Method II	Method III
Route A (2592x1936)	FT	31/51	19/51	21/51
	PS	39.22 %	62.75 %	58.82 %
	AE	3.2224	3.1089	2.4035
	PT	0.2901s	0.1255s	0.1252s
Route B (2592x1936)	FT	15/60	04/60	05/60
	PS	75 %	93.33 %	91.66 %
	AE	3.3224	3.2327	2.3955
	PT	0.0940s	0.0669s	0.0670s
COLD (640x480)	FT	67/151	37/151	51/151
	PS	55.63 %	75.50 %	66.22 %
	AE	2.8448	2.7503	1.9643
	PT	0.0661s	0.0384s	0.0383s
New College (384x512)	FT	46/175	7/175	20/175
	PS	73.71 %	96 %	88.57 %
	AE	3.5758	3.3905	2.1477
	PT	0.00862s	0.0603s	0.0602s

Note: FT: Failure number over Trials; PS: Percent of Success; AE: Average Entropy; PT: average Processing Time per image

Fig. 13 Funnel lane created by a fixed landmark in 3D view (LEFT), X-Y view (RIGHT-TOP), and X-Z view (RIGHT-BOTTOM)



at point E sees the FL (red point) at point $u^E(u_X^E, u_Y^E)$ in the destination image plane I_s^E . At point C , the MAV sees a landmark feature (red point) at point $u^C(u_X^C, u_Y^C)$ in the current image plane I_k^C . A funnel lane of one landmark feature is created by following definitions:

Definition 1 A funnel lane of a Fixed Landmark (FL) and an MAV at the end of the segment, point E , is the set of locations $\mathbb{F}_{FL,E}$ such that $C \in \mathbb{F}_{FL,E}$ for each four funnel lane constraints are satisfied:

$$|u_X^C| < |u_X^E| \quad (\text{Horizontal Constraint 1})$$

$$\text{sign}(u_X^C) = \text{sign}(u_X^E) \quad (\text{Horizontal Constraint 2})$$

$$|u_Y^C| < |u_Y^E| \quad (\text{Vertical Constraint 3})$$

$$\text{sign}(u_Y^C) = \text{sign}(u_Y^E) \quad (\text{Vertical Constraint 4})$$

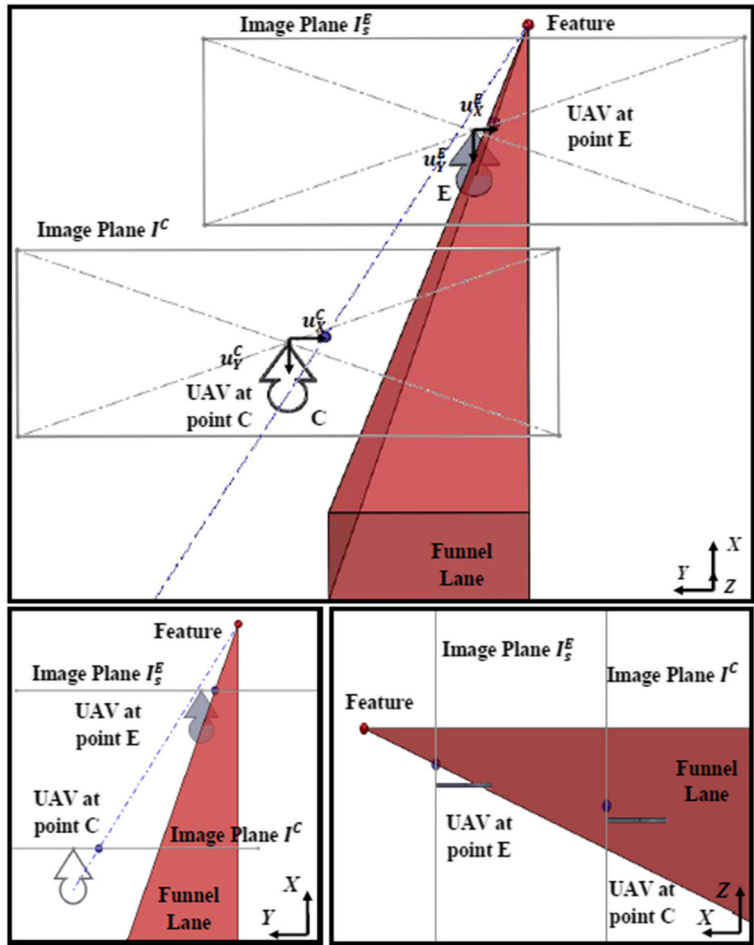
Definition 2 A funnel lane of a Fixed Landmark (FL), an MAV position at point E , and a relative angle α is the set of positions $\mathbb{F}_{FL,E,\alpha} \subset \mathbb{F}_{FL,E}$ such that $\psi_s^C - \psi_s^E = \alpha$ for each $C \in \mathbb{F}_{FL,E,\alpha}$. A relative angle α is in the plane which is parallel with X-Y plane.

The purpose of using Funnel Lane theory is to define possible positions where the MAV is maneu-

vered to fly straight. By this way, in Fig. 13, a funnel lane is produced and 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 MAV at the end of the segment, point E . The red landmark feature appears in the image plane of the MAV, which is indicated as a blue point. The MAV at P^E is presented in full-filled symbol while the MAV at P^C is presented in non-filled symbol. Figure 14 presents the case of the MAV's same heading angle ($\psi_s^C - \psi_s^E = 0$) at the current position and the end of the segment s . Figure 15 presents the case of different heading angles ($\psi_s^C - \psi_s^E \neq 0$). The funnel lane will rotate α angle as in definition 2.

Each landmark feature has a unique funnel lane. The intersection between these funnel lanes will satisfy both constraint conditions as Fig. 16. If the MAV is in the intersection funnel lane, it will fly forward. Another advantage given from this point is the reliability of navigation, the MAV needs a minimum of only one landmark feature for navigation. When the MAV falls outside the intersection funnel lane, the navigation is computed in such a way that the MAV will fly back to the intersection funnel lane. Its performance depends on the current position with respect to the intersection funnel lane. Nine possible positions of the MAV are defined on the violation of four constraints

Fig. 14 MAV from current position (non-filled symbol) is navigated to reach the end of the segment s (filled symbol) by funnel lane theory



and described in Fig. 17. The next two sections will depict the implementation of Funnel Lane theory into following the desired route.

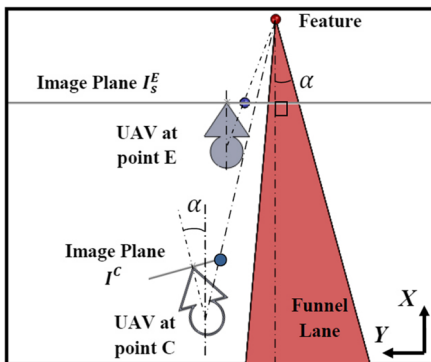


Fig. 15 Funnel lane in case of different heading angles

5.2 Building of The Desired Route

The desired route Ψ is broken into several continuous segments $\{Seg_s | s \in \{1, 2, \dots, n\}\}$. We have a set of reference images: $\{I_s^{Ref} | s \in \{1, 2, \dots, n\}\}$. If Funnel Lane theory is applied into each segment, I_{s+1}^{Ref} becomes I_s^E and P_{s+1}^{Ref} becomes P^E . The Funnel-Lane visual route is defined by the follows:

- *Hypothesis 1:* The MAV’s coordinate frame is in the segment s at current position P^C and the task is to reach the destination position P_{s+1}^{Ref} . Two key images I_k^C and I_{s+1}^{Ref} are respectively associated with P^C and P_{s+1}^{Ref} . There always exists an acceptable route χ from P^C to P_{s+1}^{Ref} .
- *Hypothesis 2:* In the segment s , a set of matched features $\{F_s^M\}$ between two key images I_k^C and

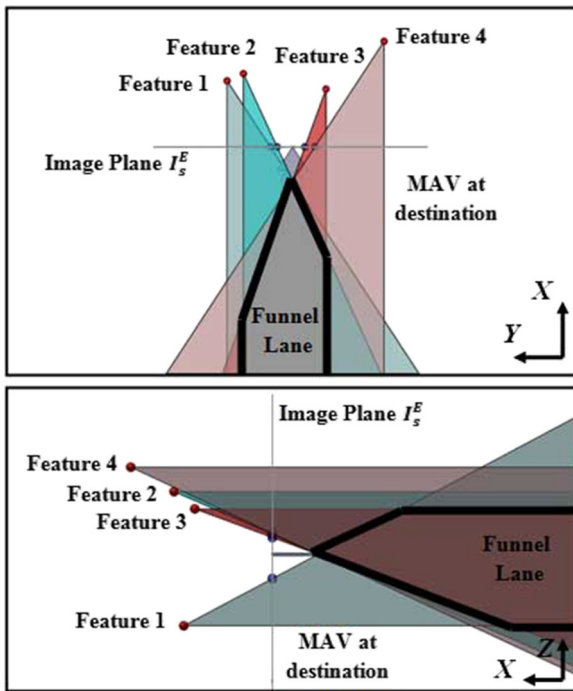


Fig. 16 Multiple feature (four features) case with X-Y and Y-Z views

I_{s+1}^{Ref} is observed along the path Ψ from P^C to P_{s+1}^{Ref} and allows the funnel-lane computation of the navigation. There exist nine possible locations of the MAV with respect to the intersection funnel lane formed by matched points $\{F_s^M\}, \{F_s^M|s \in \{1, 2, \dots, n\}\}$.

Fig. 17 Nine possible positions of the MAV w. r. t. the intersection funnel lane, which are defined by the violation of four constraints

<p>Fly Down $\text{sign}(u_Y^C) \neq \text{sign}(u_Y^E)$</p> <p>Turn Right $u_X^C > u_X^E$</p>	<p>Fly Down $\text{sign}(u_Y^C) \neq \text{sign}(u_Y^E)$</p>	<p>Fly Down $\text{sign}(u_Y^C) \neq \text{sign}(u_Y^E)$</p> <p>Turn Left $\text{sign}(u_X^C) \neq \text{sign}(u_X^E)$</p>
<p>Turn Right $u_X^C > u_X^E$</p>	<p>Fly Straight (In Funnel Lane)</p>	<p>Turn Left $\text{sign}(u_X^C) \neq \text{sign}(u_X^E)$</p>
<p>Turn Right $u_X^C > u_X^E$</p> <p>Fly Up $u_Y^C > u_Y^E$</p>	<p>Fly Up $u_Y^C > u_Y^E$</p>	<p>Turn Left $\text{sign}(u_X^C) \neq \text{sign}(u_X^E)$</p> <p>Fly Up $u_Y^C > u_Y^E$</p>

- *Hypothesis 3:* In the segment s , the condition to apply the Funnel Lane theory is that the transformation between P^C frame and P_{s+1}^{Ref} frame does not include the case which contains only lateral transformation.

5.3 Funnel-Lane 3D Motion Control Algorithm

During repeating mode, after self-localizing the current segment of the quadrotor, funnel-lane 3D motion control algorithm is performed to online generate the motion commands of desired heading and desired height:

5.3.1 The Desired Heading of Quadrotor

For each feature j , a signed distance to the line $u_X^C = u_X^E$ is calculated:

$$f(u_X^C, u_X^E) = \frac{1}{\sqrt{2}}(u_X^C - u_X^E) \tag{12}$$

And a desired heading of feature j is computed by the feature horizontal coordinate:

$$\begin{aligned} & \text{switch}(u_X^C, u_X^E) \\ & \text{case : } u_X^C > 0 \text{ and } u_X^C > u_X^E \\ & \quad \psi_d^{(j)} = \gamma_1 \cdot \min\{u_X^C, f(u_X^C, u_X^E)\} \\ & \text{case : } u_X^C < 0 \text{ and } u_X^C < u_X^E \\ & \quad \psi_d^{(j)} = \gamma_1 \cdot \max\{u_X^C, f(u_X^C, u_X^E)\} \\ & \text{case : otherwise} \\ & \quad \psi_d^{(j)} = 0 \end{aligned} \tag{13}$$

The final desired heading for quadrotor is given by:

$$\psi_d = \eta_1 \frac{1}{N} \sum_{j=1}^N \psi_d^{(j)} + (1 - \eta_1)\psi_0 \tag{14}$$

Where: γ_1 is the approximate conversion from pixels to degrees. N is the total number of matched features used in the algorithm. ψ_0 is the desired heading obtained from magnetometry heading measurements at the start and the end of the segment s in the teaching phase. η_1 ($0 \leq \eta_1 \leq 1$) is the confidence between visual measurements versus magnetometry heading measurements. In experiment, η_1 is chosen as 0.5.

5.3.2 The Desired Height of Quadrotor

For each feature j , a signed distance to the line $u_Y^C = u_Y^E$ is calculated:

$$f(u_Y^C, u_Y^E) = \frac{1}{\sqrt{2}}(u_Y^C - u_Y^E) \tag{15}$$

And a desired height of feature j is computed by the vertical feature coordinate:

```

switch( $u_Y^C, u_Y^E$ )
case :  $u_Y^C > 0$  and  $u_Y^E > u_Y^C$ 
     $Z_d^{(j)} = \gamma_2 \cdot \min\{u_Y^C, f(u_Y^C, u_Y^E)\}$ 
case :  $u_Y^C < 0$  and  $u_Y^E < u_Y^C$ 
     $Z_d^{(j)} = \gamma_2 \cdot \max\{u_Y^C, f(u_Y^C, u_Y^E)\}$ 
case : otherwise
     $Z_d^{(j)} = 0$ 
    
```

(16)

The final desired height for quadrotor is given by:

$$Z_d = \eta_2 \frac{1}{N} \sum_{j=1}^N Z_d^{(j)} + (1 - \eta_2)Z_0 \tag{17}$$

Where: γ_2 is the approximate conversion from pixels to meters. N is the total number of matched features used in the algorithm. Z_0 is the desired height obtained from altimeter measurements at the start and the end of the segment in the teaching phase. η_2 ($0 \leq \eta_2 \leq 1$) is confidence between visual measurements versus altimeter measurements. In experiment, η_2 is chosen as 0.5.

6 Experimental Result of VT&R System

Appearance-based self-localization and motion control are implemented in VT&R navigation system of quadrotor. In teaching mode, quadrotor is manually controlled along the desired route Ψ to collect $n + 1$ reference images. It means that the desired route Ψ is divided into n segments. In repeating mode, the online calculation steps are presented in Algorithm 3. Method III of self-localization is chosen to integrate with funnel-lane 3D motion control algorithm.

Algorithm 3 Online calculation steps in repeating mode in order to navigate quadrotor follow the desired route Ψ , constructing of n segments

- 1 Load $n + 1$ reference images, I_s^{Ref} , from memories;
 - 2 Detect $\{F_{s,e}^{Ref}\}$ features of I_s^{Ref} ;
 - 3 Load ψ_0 and Z_0 ;
 - 4 **if** Current segment of initial position, s_{start} , is not available **then**
 - 5 Self-localization is performed to define current segment s_{start}
 - 6 **end**
 - 7 **for** ($s \in s_{start}, \dots, n$) **do**
 - 8 Capture the current image feedback I^C ;
 - 9 Detect $\{F_f^C\}$ SURF features in current image feedback, I^C ;
 - 10 Match $\{F_f^C\}$ and $\{F_{s,e}^{Ref}\}$ by its descriptors to have $\{F_s^{M,4}\}$;
 - 11 Compute self-localization *MethodIII*{*Seg_s*} on a number of matched features ($\{F_{s,e}^{Ref}\}, \{F_{s+1,e}^{Ref}\}, \{F_s^{M,4}\}, \{F_{s+1}^{M,4}\}$), to define the current segment $s = \gamma$ of quadrotor;
 - 12 Use $\{F_{s=\gamma+1,e}^{Ref}\}$ and $\{F_{s=\gamma}^M\}$ to compute ψ_d and Z_d by funnel-lane 3D motion control algorithm;
 - 13 Transmit to the basic control module to control quadrotor follow the desired route Ψ ;
 - 14 Obstacle avoidance function (optional);
 - 15 **end**
-

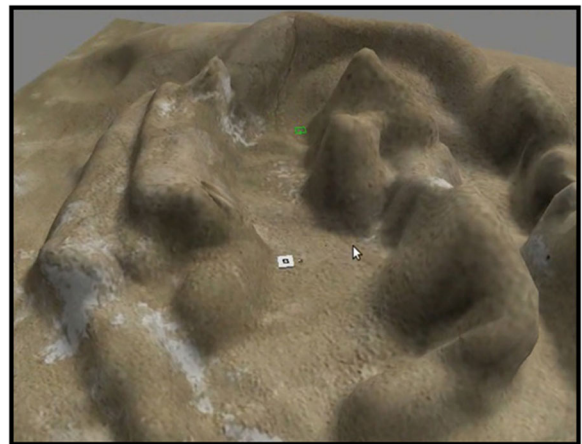


Fig. 18 Gazebo simulation of mountain area

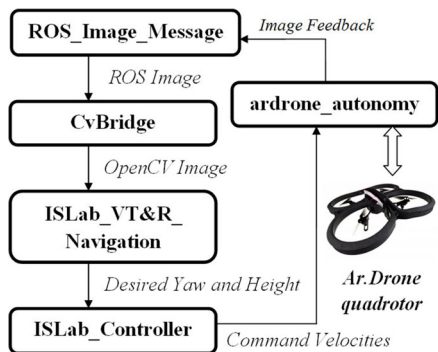


Fig. 19 ROS system architecture

The experiment is conducted on ROS Fuerte [2] with Gazebo simulator, Linux Ubuntu 12.04. Ar.Drone quadrotor model and simulated environment (Fig. 18) are created to sufficiently validate the performance of VT&R systems. The architecture of ROS system is described in Fig. 19. Image processing step utilizes OpenCV library [7]. *ISLab_VT&R_Navigation* and *ISLab_Controller* nodes are programmed to execute proposed VT&R techniques. Visual markers of the desired route Ψ (ground truth, segment notations) are generated in Rviz application (Fig. 20) in order to verify the properly working of the proposed navigation technique. In the teaching phase, quadrotor is manually controlled along the desired route Ψ

to collect reference images. The desired route Ψ has 55 reference images in mountain area. In the repeating phase, quadrotor starts at the same initial position as in the teaching phase. Online calculation steps in Algorithm 3 are proceeded to navigate the quadrotor follow the desired route Ψ . The video of the experiment is located at youtu.be/0YVGK1-ObGM. Table 2 presents the parameters used in the experiment.

7 Discussion

Experiments show that quadrotor with VT&R technique is able to navigate and follow the desired route with acceptable errors. In VT&R technique, the navigation is basing on the reference images. The positional errors are inevitable. Therefore, the performance of the system is not better than the other strategies with VICON motion capture system [27, 31, 34, 35]. It is a limitation of appearance-based VT&R technique but considered as practical within GPS-denied environment.

Figures 21, 22 and 23 show the experiment with the long desired route in mountain area, containing 55 reference images. A number of reference images negatively increase the computation cost if all reference images are processed at the same time. However, this challenge can be overcome if estimating the quadrotor location on all reference images

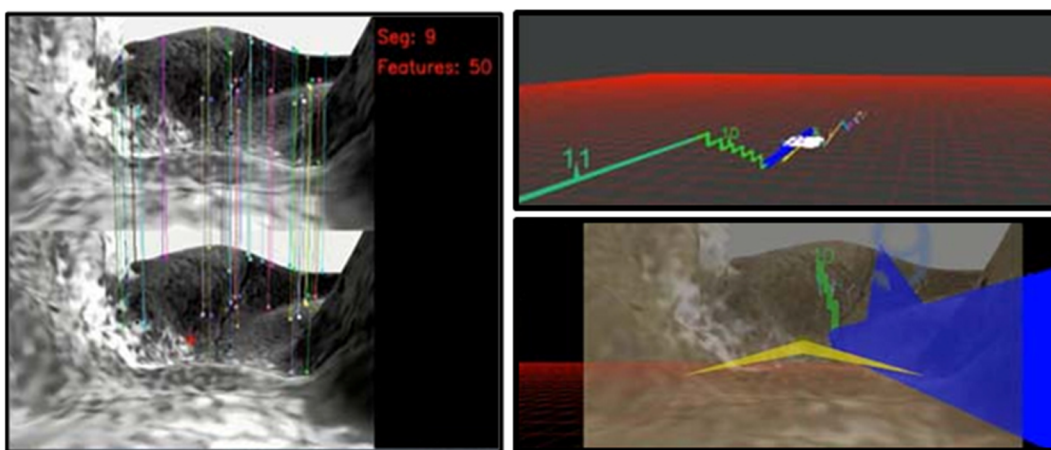


Fig. 20 Simulation performance of tracking the desired route. LEFT: Images showing matching SURF features between the reference image (Top) and the current image (Bottom). RIGHT-TOP: Rviz application with colorful visual markers

of the desired route in order to check ground truth tracking. RIGHT-BOTTOM: Forward-looking camera-view of quadrotor in Rviz

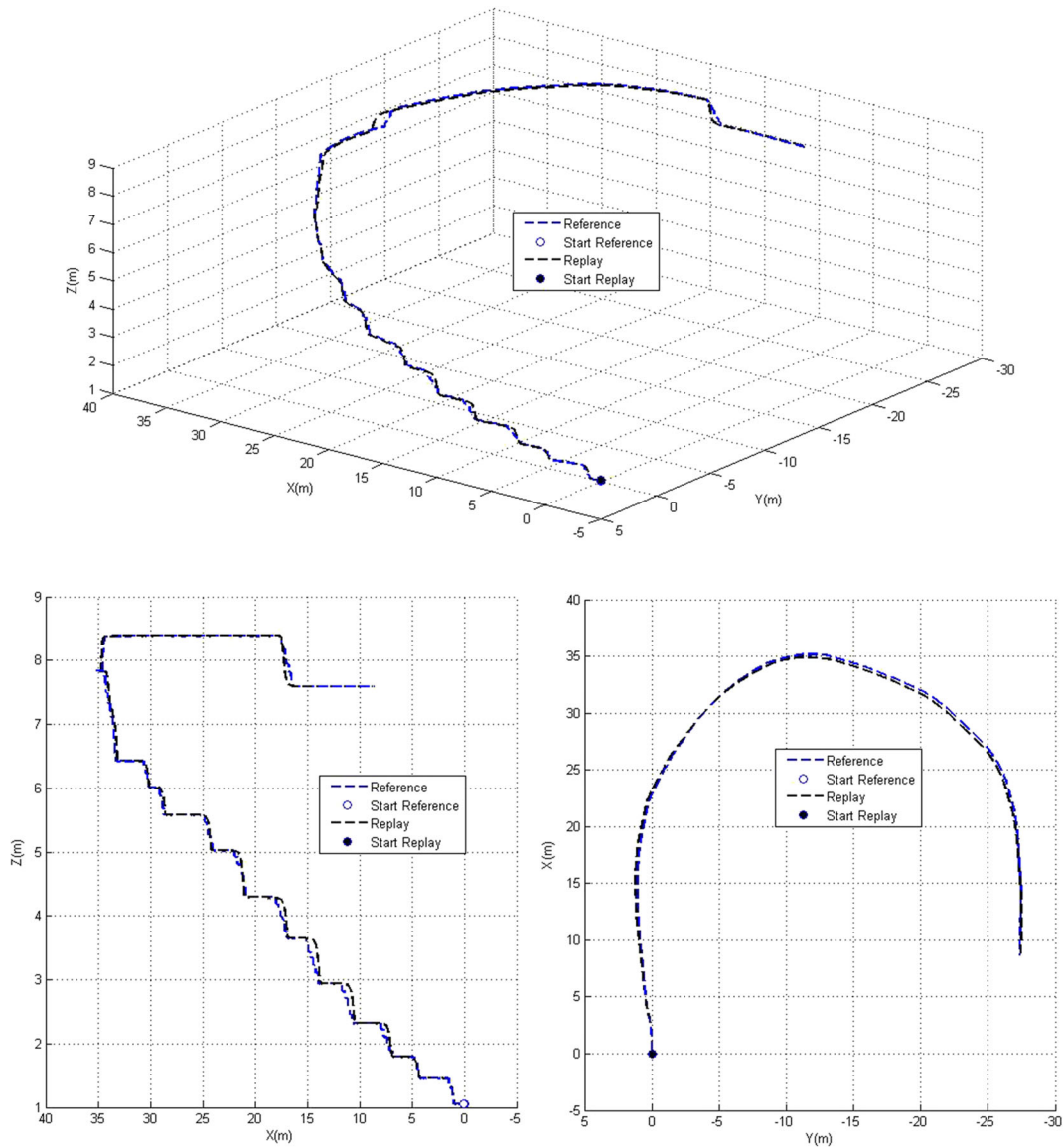
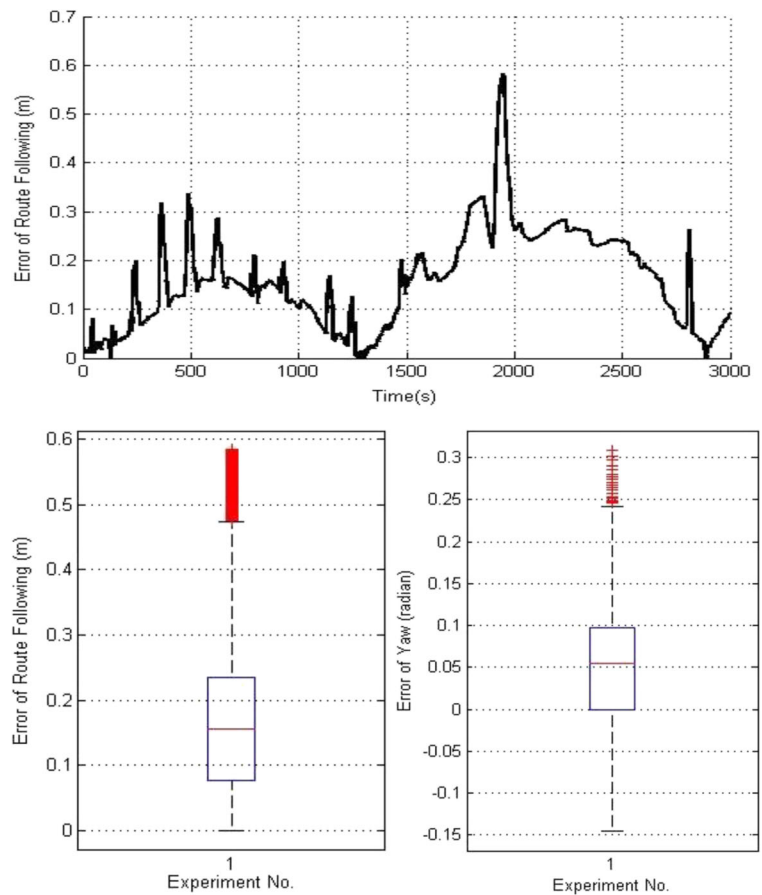


Fig. 21 Simulation results of tracking the trained route in mountain area. *TOP*: 3D view; *LEFT-BOTTOM*: X-Z view; *RIGHT-BOTTOM*: X-Y view

in the beginning of the repeating phase. Then, estimating location is performed by 3 or 4 reference images logically related to the current segment. For example, if quadrotor is known at segment 5, self-localization will use reference images of segment 4, 5 and 6.

In the experiment, the complexity of the proposed technique is measured: average 0.35s for processing time; 1360M for the virtual size of memory (VIRT); 66092 for the resident size of memory (RES); 18780 of shareable memory in VIRT (SHR). The Image Feedback topic from quadrotor camera

Fig. 22 Errors of Route following. *LEFT-TOP*: Errors following time, *LEFT-BOTTOM*: Box plot of errors, *RIGHT-BOTTOM*: Box plot of yaw errors



is received at the rate of 20Hz. Then the proposed technique is performed following sequential steps of calculation (Fig. 19) in order to produce the topic

of command velocities to the quadrotor at the rate of 7Hz. The frequency configuration of publishing these topics is adequate for the quadrotor tracking the

Fig. 23 *TOP*: Yaw in teaching phase; *BOTTOM*: Yaw response in repeating phase

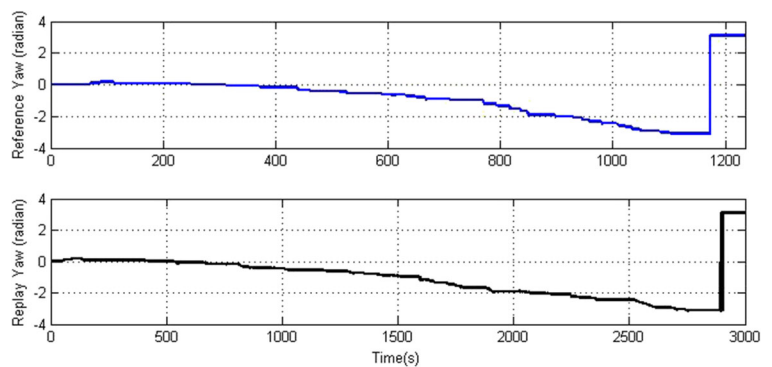


Table 2 Parameters of the experiment

Number of reference image	55
Maximum number of feature	200
Feature Detection	SURF
Feature Descriptor	BRIEF
Descriptor BRIEF length	32 bytes
Descriptor BRIEF patch size	48
Descriptor BRIEF kernel size	9
Descriptor Matcher	BruteForce-Hamming
Confidence between measurements	$\eta_1 = 0.5, \eta_2 = 0.5$

route. The proposed technique can also perform with higher frequency of image feedback in order to produce higher frequency of control topic. However, the limitation of hardware computation should be considered at that time.

The structure of VT&R system requires considerable memory to store the reference images of the desired route. The quality and quantity of the reference images decide the size of required memories. Hence, an effective method to store and update the database of the desired route is also needed. Due to the limitation of Ar.Drone quadrotor hardware, the paper does not attempt to store the database of the desired route onboard, which removes the independent operations of VT&R technique. However, this problem can be easily overcome by the additional flight recorder onto Ar.Drone quadrotor hardware [1] or using other models of quadrotor. On the other hand, for practical applications, the VT&R does not need to store the whole set of reference images. Storing the SURF features and its descriptors is enough to perform feature matching technique.

The construction of funnel-lane 3D motion control algorithm indirectly handles the problem of accidental collision during the repeating phase. After accidentally colliding to other objects in environment, quadrotor can be rotated to unpredictable heading angle. Hence, all matched landmarks will fall out of the image plane of quadrotor camera, and fail the VT&R system. However, since the paper utilizes heading and height measurements from other sensors than vision in the calculation of motion control commands, quadrotor will rotate back as before collision and correct matched features will come back to the image plane of quadrotor camera (for video: youtu.be/WVq9IttJx0g).

8 Conclusion and Future Work

The paper has developed appearance-based VT&R technique on quadrotor aerial vehicle in order to navigate in GPS-denied environment. VT&R technique can be considered as the simpler form of SLAM technique, which reduces the demand of higher computational cost and complexity in implementation. Navigating while comparing reference images along the desired route, VT&R technique can overcome the drift problem of quadrotor. The proposed design of VT&R aerial system is constructed using two different components: self-localization and motion control. Firstly, three methods of self-localization technique are proposed to improve the performance of estimating the current segment of quadrotor. Utilizing the relation between spatial distance and the size of SURF feature, method II and method III show better self-localization than method I. Method II produces insignificantly better self-localization than method III but shows some disadvantages at the uniformity of the probability distribution. Secondly, qualitative motion control is developed on Funnel Lane theory to decrease the computational cost and to reach the simpler form of motion control command calculation. Qualitative motion control commands are calculated by funnel-lane 3D motion control algorithm.

In order to increase the stability and reliability of the VT&R system for practical applications, additional parts such as obstacle avoidance, memory management will be developed. Obstacle avoidance and path planning will handle the unexpected appearance of obstacles on the desired route in the repeating phase. Memory management is to effectively manage and adaptively update the database of the desired route in dynamic environment. Path planning shows potential to develop in case of multiple databases, which allows many ways to reach destination. In addition, the ROS system should be redesigned to enable onboard processing and fully autonomous applications. The considerable changes of the working environment between the teaching phase and the repeating phase can make the VT&R system unable to navigate. It requires another effective fusion of numerous sensors and advanced navigating calculation to help quadrotor working in dynamic environment but still satisfies the limitation of quadrotor's payload. The navigation on Funnel Lane theory is also extended for the case of rear-looking or omnidirectional camera.

Acknowledgments This work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), C-CORE J. I. Clark Chair, Memorial University of Newfoundland (MUN) and RDC Ocean Industries Student Research Awards (5404-1774-101).

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.

Appendix A: Reference images from used databases

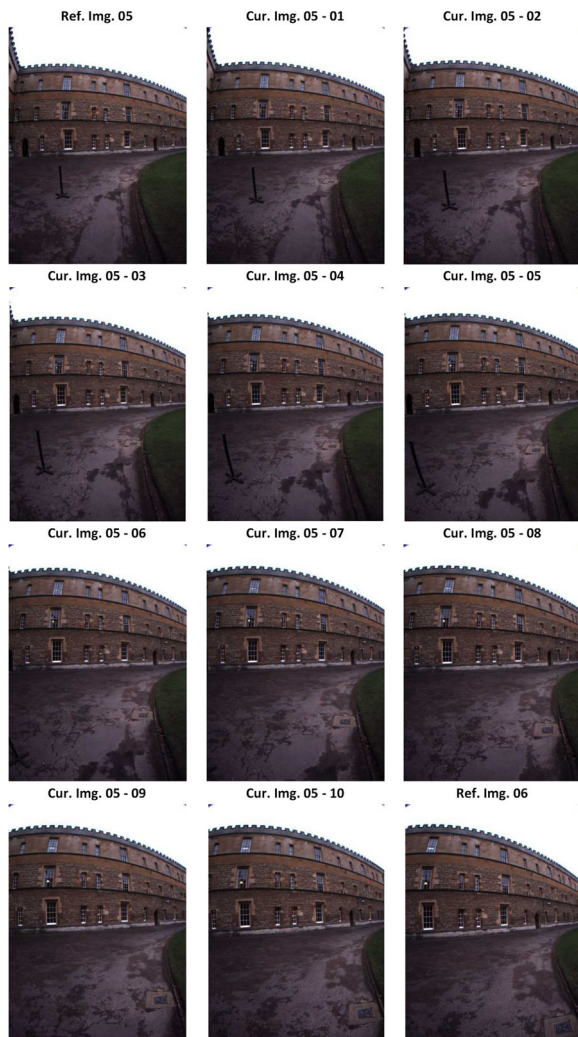


Fig. 24 Reference images and tested current images of segment 5, New College database



Fig. 25 Reference images of the route A - Second Floor of Engineering building

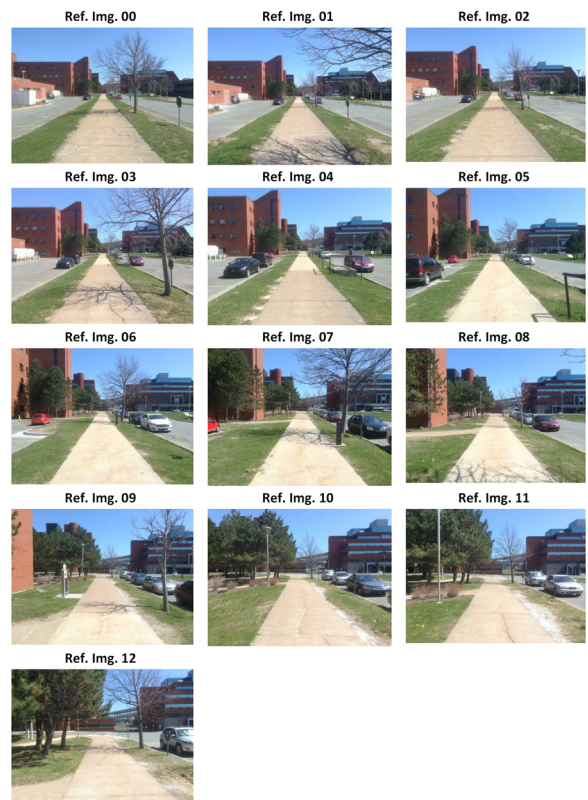


Fig. 26 Reference images of the route B - Outside of Engineering building

References

1. Parrot AR.Drone ver 2.0 quadrotor, <http://ardrone2.parrot.com>
2. Robot Operating System (ROS), <http://www.ros.org>
3. Bachrach, A., Prentice, S.: RANGE: Robust autonomous navigation in GPS-denied environments. *Journal of Field Robotics* **28**(5), 644–666 (2011)
4. Barfoot, T.D., Stenning, B., Furgale, P., McManus, C.: Exploiting Reusable Paths in Mobile Robotics: Benefits and Challenges for Long-term Autonomy. In: *Proceedings of Conference on Computer and Robot Vision*, pp. 388–395 (2012)
5. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-Up Robust Features (SURF). *Comp. Vision Image Underst.* **110**(3), 346–359 (2008)
6. Becerral, H.M., Courbon, J., Mezouar, Y., Sagues, C.: Wheeled mobile robots navigation from a visual memory using wide field of view cameras. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5693–5699 (2010)
7. Bradski, G., Kaehler, A.: *Learning OpenCV: Computer vision with the openCV library*. O'Reilly Media (2008)
8. Calonder, M., Lepetit, V., Ozuysal, M., Trzcinski, T., Strecha, C., Fua, P.: BRIEF: Computing a Local Binary Descriptor very Fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2011)
9. Fischler, M.A., Bolles, R.C.: Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun ACM* **24**(6), 381–395 (1981)
10. Chen, Z., Birchfield, S.T.: Qualitative vision-based mobile robot navigation. In: *Proceedings of IEEE International Conference on Robotics and Automation*, May, pp. 2686–2692 (2006)
11. Chen, Z., Birchfield, S.T.: Qualitative vision-based path following. *IEEE Trans. Robot.* **25**(3), 749–754 (2009)
12. Churchill, D., Vardy, A.: An orientation invariant visual homing algorithm. *J. Intell. Robot. Syst.* (2013)
13. Courbon, J., Mezouar, Y.: Visual navigation of a quadrotor aerial vehicle. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5315–5320 (2009)
14. Courbon, J., Mezouar, Y., Martinet, P.: Visual navigation of a quadrotor Aerial Vehicle. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5315–5320 (2009)
15. Dame, A., Marchand, E.: A new information theoretic approach for appearance-based navigation of non-holonomic vehicle. In: *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2459–2464 (2011)
16. De Silva, O., Mann, G.K.I., Gosine, R.G.: Development of a relative localization scheme for ground-aerial multi-robot systems. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 870–875 (2012)
17. Eckert, J., German, R., Dressler, F.: An indoor localization framework for four-rotor flying robots using low-power sensor nodes. *IEEE Trans. Instrum. Meas.* **60**(2), 336–344 (2011)
18. Eckert, J., German, R., Dressler, F.: Self-Organized localization in indoor environments using the ALF framework. *International Journal of Communication Networks and Distributed Systems* **10**(2), 102–122 (2013)
19. Erhard, S., Wenzel, K.E., Zell, A.: Flyphone: Visual Self-Localisation Using a Mobile Phone as Onboard Image Processor on a Quadcopter. *J. Intell. Robot. Syst.* **57**(1–4), 451–465 (2009)
20. Forster, C., Pizzoli, M., Scaramuzza, D.: Air-ground localization and map augmentation using monocular dense reconstruction. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3971–3978 (2013)
21. Fraundorfer, F., Heng, L.: Vision-based autonomous mapping and exploration using a quadrotor MAV. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4557–4564 (2012)
22. Fu, C., Olivares-Mendez, M.A., Suarez-Fernandez, R., Campoy, P.: Monocular Visual-Inertial SLAM-Based Collision Avoidance Strategy for Fail-Safe UAV Using Fuzzy Logic Controllers. *J. Intell. Robot. Syst.* **73**(1–4), 513–533 (2013)
23. Furgale, P., Barfoot, T.: Stereo mapping and localization for long-range path following on rough terrain. In: *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 4410–4416 (2010)
24. Furgale, P., Barfoot, T.D.: Visual teach and repeat for long range rover autonomy. *Journal of Field Robotics* **27**(2006), 534–560 (2010)
25. Hochdorfer, S., Schlegel, C.: 6 DoF SLAM using a ToF camera: The challenge of a continuously growing number of landmarks. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3981–3986 (2010)
26. Krajník, T., De Cristóforis, P., Faigl, J., Szücsová, H., Nitsche, M., Preucil, L., Mejail, M.: Image features for long-term mobile robot autonomy. *IEEE Conference on Robotics and Automation, Workshop on Long-Term Autonomy* (2013)
27. Kushleyev, A., Mellinger, D., Powers, C., Kumar, V.: Towards a swarm of agile micro quadrotors. *Auton. Robot.* **35**(4), 287–300 (2013)

28. Lee, G.H., Achtelik, M., Fraundorfer, F., Pollefeys, M., Siegwart, R.: A benchmarking tool for MAV visual pose estimation. In: Proceedings of 11th International Conference on Control Automation Robotics & Vision, vol. 1, pp. 1541–1546 (2010)
29. Liu, W., Zheng, N., Xue, J., Zhang, X., Yu, Z.: Visual appearance-based unmanned vehicle sequential localization. *Int. J. Adv. Robot. Syst.* **10** (2013)
30. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**(2), 91–110 (2004)
31. Lupashin, S., Hehn, M., Mueller, M.W., Schoellig, A.P., Sherback, M., DAndrea, R.: A platform for aerial robotics research and demonstration: The Flying Machine Arena. *Mechatronics* **24**(1), 41–54 (2014)
32. Majdik, A.L., Albers-Schoenberg, Y., Scaramuzza, D.: MAV urban localization from Google street view data. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3979–3986 (2013)
33. McManus, C., Furgale, P., Stenning, B., Barfoot, T.D.: Visual Teach and Repeat using appearance-based lidar. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 389–396 (2012)
34. Mellinger, D., Kumar, V.: Minimum snap trajectory generation and control for quadrotors. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 2520–2525 (2011)
35. Michael, N., Mellinger, D., Lindsey, Q., Kumar, V.: The GRASP Multiple Micro-UAV Testbed. *IEEE Robotics Automation Magazine* **17**, 56–65 (2010)
36. Michael, N., Shen, S., Mohta, K., Mulgaonkar, Y., Kumar, V., Nagatani, K., Okada, Y., Kiribayashi, S., Otake, K., Yoshida, K., Ohno, K., Takeuchi, E., Tadokoro, S.: Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *Journal of Field Robotics* **29**(5), 832–841 (2012)
37. Mori, T., Scherer, S.: First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 1750–1757 (2013)
38. Munoz, L.E., Castillo, P., Garcia, P.: Observer-control scheme for autonomous navigation: Flight tests validation in a quadrotor vehicle. In: Proceedings of International Conference on Unmanned Aircraft Systems, pp. 795–804 (2013)
39. Murillo, A.: Surf features for efficient robot localization with omnidirectional images. In: Proceedings of IEEE International Conference on Robotics and Automation, April, pp. 10–14 (2007)
40. Nguyen, T., Mann, G.K., Gosine, R.G.: Vision-based qualitative path-following control of quadrotor aerial vehicle. In: Proceedings of International Conference on Unmanned Aircraft Systems (2014)
41. Nguyen, T., Mann, G.K., Gosine, R.G.: Vision-Based Qualitative Path-Following Control of Quadrotor Aerial Vehicle with Speeded-Up Robust Features. In: Proceedings of Canadian Conference on Computer and Robot Vision, pp. 321–327 (2014)
42. Ostafew, C.J., Schoellig, A.P., Barfoot, T.D.: Visual teach and repeat, repeat, repeat: Iterative Learning Control to improve mobile robot path tracking in challenging outdoor environments. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 176–181 (2013)
43. Pfrunder, A., Schoellig, A.P., Barfoot, T.D.: A Proof-Of-Concept Demonstration of Visual Teach and Repeat on a Quadcopter Using an Altitude Sensor and a Monocular Camera. In: Proceedings of Canadian Conference on Computer and Robot Vision (2014)
44. Pronobis, A., Caputo, B.: COLD: The CoSy Localization Database. *Int. J. Robot. Res.* **28**(5), 588–594 (2009)
45. Sebastian Thrun Wolfram Burgard, D.F.: Probabilistic Robotics. MIT Press (2005)
46. Shen, S., Michael, N., Kumar, V.: Autonomous multi-floor indoor navigation with a computationally constrained MAV. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 20–25 (2011)
47. Smith, M., Baldwin, I., Churchill, W., Paul, R., Newman, P.: The New College Vision and Laser Data Set. *Int. J. Robot. Res.* **28**(5), 595–599 (2009)
48. Vaca-Castano, G., Zamir, A.R., Shah, M.: City scale geospatial trajectory estimation of a moving camera. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1186–1193 (2012)
49. Valgren, C., Lilienthal, A.J.: SIFT, SURF & seasons: Appearance-based long-term localization in outdoor environments. *Robot. Auton. Syst.* **58**(2), 149–156 (2010)
50. Vandrish, P., Vardy, A., King, P.: Towards AUV Route Following Using Qualitative Navigation. In: Proceedings of IEEE Canadian Conference on Computer and Robot Vision, pp. 425–432 (2012)
51. Vardy, A.: Using feature scale change for robot localization along a route. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4830–4835 (2010)
52. Zhang, A.M., Kleeman, L.: Robust Appearance Based Visual Route Following for Navigation in Large-scale Outdoor Environments. *Int. J. Robot. Res.* **28**(3), 331–356 (2009)
53. Zhou, Q., Zhang, Y.: Dead reckoning and Kalman filter design for trajectory tracking of a quadrotor UAV. In: Proceedings of IEEE/ASME International Conference on Mechatronics and Embedded Systems, pp. 119–124 (2010)

Trung Nguyen received the B.Eng. degree in mechanical engineering from Ho Chi Minh City University of Technology, Vietnam in 2012; M.Eng. degree from Memorial University of Newfoundland, Canada in 2014. Since 2014, he has been working toward the Ph.D. degree at the Faculty of Engineering and Applied Science, Memorial University of Newfoundland, Canada. His main research interests include intelligent control, machine vision, autonomous underwater vehicles and micro aerial vehicles.

George K. I. Mann received B.Sc. in engineering from the University of Moratuwa, Sri Lanka, M.Sc. from Loughborough University in U.K., and Ph.D. from Memorial University of Newfoundland, Canada. Following two years of post-doctoral work at C-CORE and Queen's University, in 2001 he joined the Memorial University of Newfoundland as a Faculty Member. From 2002-2007 he also served as the C-CORE Junior Chair in Intelligent Systems. Currently, he is a Professor in the Faculty of Engineering and Applied Science at MUN. His main research areas are intelligent control, autonomous robotics, machine vision, and hybrid power systems.

Raymond G. Gosine received the B.Eng. degree in electrical engineering from Memorial University of Newfoundland, St. John's, NL, Canada, and the Ph.D. degree in robotics from Cambridge University, Cambridge, U.K., in 1990. From 1991 to 1993, he was the NSERC Junior Chair of Industrial Automation and an Assistant Professor in Department of Mechanical Engineering, the University of British Columbia, Vancouver, BC, Canada. In 1994, he joined the Faculty of Engineering, Memorial University of Newfoundland, and also served as the Director for the Intelligent Systems Group at C-CORE. Currently, he is a Professor and Vice-President Research (pro term) at Memorial University of Newfoundland. His main research areas are telerobotics, machine vision, and pattern recognition.

Andrew Vardy completed a B.Eng in electrical engineering from Memorial University in 1999. After graduating, he became quite interested in bio-inspired computing and pursued this interest by completing a master's degree in evolutionary and adaptive systems from the University of Sussex in Brighton, UK. He continued his studies by completing a PhD in computer science from Carleton University in Ottawa. In 2005, he returned to Memorial University as an assistant professor, jointly appointed to the Department of Computer Science and the Faculty of Engineering and Applied Science. In 2011, he was promoted to associate professor. His research interests include swarm robotics, bio-inspired robotics, and visual robot navigation.