

A Robust Real-Time Vision System for Autonomous Cargo Transfer by an Unmanned Helicopter

Shiyu Zhao, *Member, IEEE*, Zhangyuan Hu, Mingfeng Yin, Kevin Z. Y. Ang, Peidong Liu, Fei Wang, *Student Member, IEEE*, Xiangxu Dong, Feng Lin, *Member, IEEE*, Ben M. Chen, *Fellow, IEEE*, and Tong H. Lee, *Member, IEEE*

Abstract—Motivated by the 2013 International UAV Innovation Grand Prix, we design and implement a real-time vision system for an unmanned helicopter autonomously transferring cargoes between two platforms. In the competition, four cargoes are initially placed inside four circles on one platform, respectively. They are required to be transferred one by one into the four circles on the other platform. This paper presents the core algorithms of the proposed vision system on ellipse detection, ellipse tracking, and single-circle-based position estimation. Experiments and the great success of our team in the competition have verified the efficiency, accuracy, and robustness of the algorithms. Our team was ranked first in the final round competition.

Index Terms—Circle-based pose estimation, ellipse detection, ellipse tracking, unmanned aerial vehicle (UAV), vision-based navigation.

I. INTRODUCTION

A. Background

THE work presented in this paper is motivated by an international competition, the 2013 International UAV Innovation Grand Prix, which was held in September 2013 in Beijing, China. The competition requires an unmanned aerial vehicle (UAV) to autonomously transfer cargoes from one moving platform to the other (see Fig. 1). The cargoes are four buckets, each of which weighs 1.5 kg. In addition to cargo transfer, the UAV is also required to autonomously take off and land. The entire competition task must be completed by the UAV fully autonomously without any human intervention.

Manuscript received November 17, 2013; revised April 9, 2014 and May 31, 2014; accepted July 9, 2014. Date of publication August 7, 2014; date of current version January 7, 2015.

S. Zhao, K. Z. Y. Ang, P. Liu, B. M. Chen, and T. H. Lee are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576 (e-mail: shiyuzhao@nus.edu.sg; kevinang@nus.edu.sg; elelp@nus.edu.sg; bmchen@nus.edu.sg; eleleeth@nus.edu.sg).

Z. Hu is with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong (e-mail: mua08p@gmail.com).

M. Yin is with the Department of Automation, Nanjing University of Science and Technology, Nanjing 210044, China (e-mail: yinmingfeng-just@gmail.com).

F. Wang, X. Dong, and F. Lin are with the Temasek Laboratories, National University of Singapore, Singapore 117411 (e-mail: wangfei@nus.edu.sg; tsldngx@nus.edu.sg; linfeng@nus.edu.sg).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIE.2014.2345348

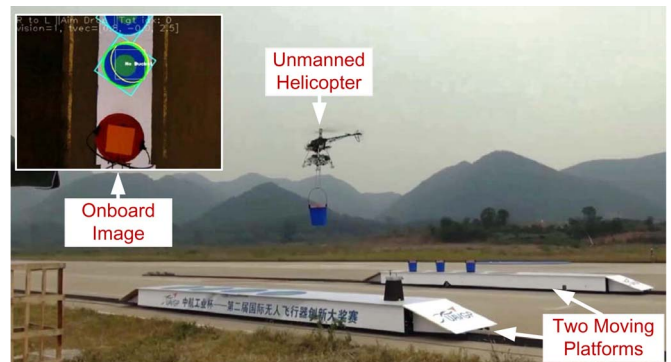


Fig. 1. Unmanned helicopter is approaching a platform to unload a cargo.



Fig. 2. Ship-to-ship vertical replenishment. (Use of released U.S. Navy imagery does not constitute product or organizational endorsement of any kind by the U.S. Navy.)

The techniques developed for the competition are potentially applicable to practical autonomous cargo transfer tasks such as vertical replenishment (see Fig. 2).

The team from the Unmanned Aircraft Systems (UAS) group at the National University of Singapore has developed a sophisticated unmanned helicopter system. The unmanned helicopter successfully completed the required competition tasks, and our team was ranked first in the final round competition. This paper focuses on the vision system of the unmanned helicopter. Details of the hardware, software, control, and navigation systems can be found in [1] and [2].

B. Related Work

Autonomous cargo transfer using UAVs is a practically important research topic, which has been studied in [3]–[6]. Most of the existing works are based on the assumption that the position of the cargo relative to the UAV is accurately known. However, the navigation of UAVs in outdoor is mainly based on GPS/Inertial Navigation System, whose measurement accuracy may practically reach several meters and does not meet the requirement to grasp or drop a cargo precisely. Up to now, implementations of vision-based autonomous cargo transfer by UAVs in outdoor environments cannot be found in the literature to the best of our knowledge.

Ellipse detection has been extensively investigated up to now [7]–[13]. Compared with the methods based on Hough transform [9], ellipse fitting [7], [8] is a very efficient method to find an ellipse for a given contour. However, ellipse fitting is not able to tell whether the fitted ellipse is a correct or a false detection. The work in [14] proposed affine moment invariants (AMIs) that are invariant to general affine transformations. Since an arbitrary ellipse can be obtained by applying an affine transformation to a circle, all ellipses and all circles have exactly the same AMIs. Hence, AMIs are powerful tools for ellipse detection [11]–[13].

There are generally two approaches to image tracking: 1) filtering and data association [15]–[18] and 2) target representation and localization [19]–[23]. The first approach usually relies on Kalman or particle filtering techniques. In UAV applications, the first approach can give reliable tracking performance with the assistance of other motion sensors [18]. Considering that this approach is usually computationally intensive and requires other types of sensors, we use the second approach based on CAMShift [19], [21]–[23] in our work.

Pose estimation based on circle features has been studied by many researchers [24]–[29]. The existing work mainly focused on the cases of concentric circles [25]–[28], whereas the aim of our work is to estimate the relative position of a single circle based only on its image, i.e., a single ellipse. It is pointed out in [29] that other information such as parallel lines is required to estimate the pose of a single circle. By adopting a reasonable assumption, we propose an efficient algorithm to estimate the relative position of a single circle without the assistance of any other information.

C. Main Contributions

- 1) We design and implement a real-time vision system for autonomous cargo transfer by an unmanned helicopter. The vision system has been proved as efficient, accurate, and robust by experimental results and the great success of our team in the competition.
- 2) This paper presents efficient and robust algorithms for ellipse detection, ellipse tracking, and single-circle-based position estimation. These algorithms are potentially applicable to a broad range of vision-based tasks such as takeoff and landing [27], [28], [30], target tracking and following [18], [31], visual servoing [32]–[35], and formation control [36] as long as circles are used as the visual features.

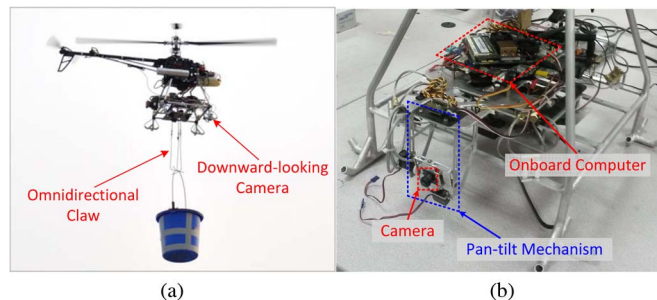


Fig. 3. Unmanned helicopter developed by the UAS group. (a) Helicopter grasping a cargo. (b) Onboard vision system.

- 3) Although computer vision provides promising approaches to many UAV tasks [23], [37], computationally intensive vision algorithms are practically inapplicable due to the very limited computational resources of UAVs. When designing the vision system, we put more emphasis on the efficiency of the algorithms and the convenience of implementation. All of the proposed algorithms can be implemented by the popular library OpenCV.

II. OVERVIEW OF THE VISION SYSTEM

A. Objectives

In the competition, four cargoes (buckets) are initially placed inside four circles on one platform, respectively. The four cargoes need to be transferred one by one into the four circles on the other platform. Apparently, circles are the key features for the vision system to process. The vision system should be able to detect all the circles and cargoes in the image and then intelligently decide which cargo the helicopter should load and where the cargo should be transferred. When loading or unloading a cargo, the target area must be continuously tracked and localized by the vision system. In emergency situations (for example, all the cargoes are out of the field of view of the camera), the vision system should inform the control system such that the unmanned helicopter can take appropriate actions.

B. Hardware Design

As shown in Fig. 3, the vision system consists of a downward-looking camera, a pan-tilt mechanism, and an onboard computer. The camera is mounted on a pan-tilt mechanism that is located on the nose of the helicopter [see Fig. 3(b)]. The pan-tilt mechanism is controlled at 50 Hz to compensate the fast dynamic rotation of the helicopter. The compensation of the helicopter rotation is essential for smooth target tracking in the image. In the meantime, the image plane of the downward-looking camera can be controlled to be horizontal and hence parallel to the circle features placed on the horizontal ground. As will be shown later, this parallel condition can greatly simplify the single-circle-based position estimation problem. The onboard computer for vision processing is an AscTec Mastermind personal computer with the CPU as Intel Core2Duo SL9400 (2×1.86 GHz). The vision programs can be executed at 10 Hz in the onboard computer.

C. Working Procedure

We next briefly describe the working procedure of the vision system. The helicopter autonomously takes off at a place far from the platforms, and then, it is guided to the platforms based on GPS navigation. Since the measurement error of GPS may reach more than 5 m, the helicopter may not be guided to the exact target point over the platforms. To handle this problem, the vision system will start to search circles when the helicopter flies close the platforms. Once several circles can be detected, the position of the helicopter will be adjusted based on the vision measurement. The cargo transfer mode of the helicopter will be triggered once the vision system can successfully detect all the eight circles on the two platforms. When loading or unloading a cargo, the helicopter first flies to a point about 10 m above the target area and then descends to about 1.5 m. During this descending procedure, the downward-looking camera can continuously observe the target.

Since each cargo is initially placed inside a circle, we need to first localize the circles and then search the cargoes inside. If multiple circles/cargoes are detected in the image, the vision system can intelligently select a target cargo to load. Once the target is selected, it will be tracked over the image sequence, and its relative position will be continuously estimated. From the above discussion, it is obvious that ellipse detection, ellipse tracking, and single-circle-based position estimation are the core problems to be solved. In fact, the three problems are also crucial for any other vision-based tasks that use circles as visual features. Although a series of other algorithms have been also developed for the competition, we will mainly introduce our proposed algorithms for the three problems in this paper due to space limitations.

III. ELLIPSE DETECTION

Here, we present a three-step procedure for efficient and robust ellipse detection. The three-step procedure consists of the following: 1) preprocessing based on AMIs; 2) ellipse fitting; and 3) postprocessing based on algebraic error. The proposed algorithm can robustly detect both whole ellipses and partially occluded ones.

A. Three-Step Ellipse Detection Procedure

In order to detect ellipses, we need to first obtain the contours in the image by using, for example, color thresholding or edge detection. Once the contours have been obtained, each of them will be processed by the following three steps to see if it corresponds to an ellipse.

Preprocessing Based on AMIs: The purpose of the preprocessing step is to compute the AMIs of a contour and compare them with the theoretical values. The theoretical values of the AMIs for an arbitrary circle or ellipse are calculated as below.

Four AMIs were proposed in [14, Section 2.1]. We only use the first three in our work as the fourth one is much more

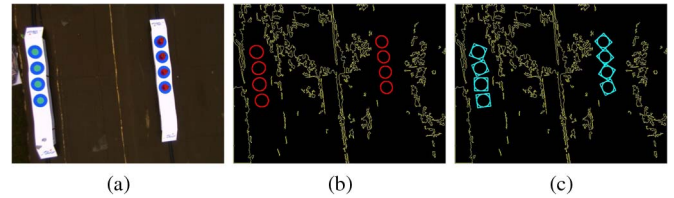


Fig. 4. Example to demonstrate the preprocessing and ellipse fitting. (a) Color image. (b) Elliptical contours detected based on AMIs. (c) Fitted ellipses with rotated bounding boxes.

complicated and less reliable than the first three. The first three AMIs are expressed as

$$\begin{aligned} I_1 &= (\mu_{20}\mu_{02} - \mu_{11}^2) / \mu_{00}^4 \\ I_2 &= (\mu_{30}^2\mu_{03}^2 - 6\mu_{30}\mu_{21}\mu_{12}\mu_{03} + 4\mu_{30}\mu_{12}^3 \\ &\quad + 4\mu_{21}^3\mu_{03} - 3\mu_{21}^2\mu_{12}^2) / \mu_{00}^{10} \\ I_3 &= (\mu_{20}(\mu_{21}\mu_{03} - \mu_{12}^2) - \mu_{11}(\mu_{30}\mu_{03} - \mu_{21}\mu_{12}) \\ &\quad + \mu_{02}(\mu_{30}\mu_{12} - \mu_{21}^2)) / \mu_{00}^7 \end{aligned} \quad (1)$$

where the central moment μ_{ij} is given by

$$\mu_{ij} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x - \bar{x})^i (y - \bar{y})^j \rho(x, y) dx dy. \quad (2)$$

In the preceding equation, (\bar{x}, \bar{y}) is the coordinate of the centroid, and $\rho(x, y)$ is the density distribution function. Suppose that $\rho(x, y) = 1$ when (x, y) is on the circle, and $\rho(x, y) = 0$ otherwise. Because all ellipses and circles have the same AMIs, we only need to calculate the AMIs of a special one: a circle centered at the origin with the radius as one unit. Then, (2) can be rewritten as $\mu_{ij} = \int_0^{2\pi} \cos^i \phi \sin^j \phi d\phi$, based on which it is easy to calculate that $\mu_{00} = 2\pi$, $\mu_{11} = 0$, $\mu_{02} = \mu_{20} = \pi$, and $\mu_{12} = \mu_{21} = \mu_{03} = \mu_{30} = 0$. Substituting these values into (1) yields the theoretical values of the AMIs of an arbitrary ellipse or circle, i.e.,

$$I_1 = \frac{1}{16\pi^2} \approx 0.006332 \quad I_2 = I_3 = 0. \quad (3)$$

If the AMIs of a contour are sufficiently close to the theoretical values, the contour can be classified as a good ellipse candidate. Note that the theoretical values are calculated based on an infinite number of points, whereas a contour in an image always has a finite number of pixel points. As a result, even if a contour corresponds to an ellipse, its AMIs still cannot be the same as the theoretical values. According to our experience, we recommend the thresholds for I_1 , I_2 , and I_3 to be ± 0.0003 , ± 0.0000001 , and ± 0.000001 , respectively. This set of thresholds can accept almost all elliptical contours, while rejecting contours with other various shapes. However, a small number of false detections may be accidentally accepted. The false detections can be further eliminated by the following postprocessing procedure, as will be shown later.

Fig. 4 shows an example to verify the effectiveness of the preprocessing step. As shown in Fig. 4(b), even in the presence

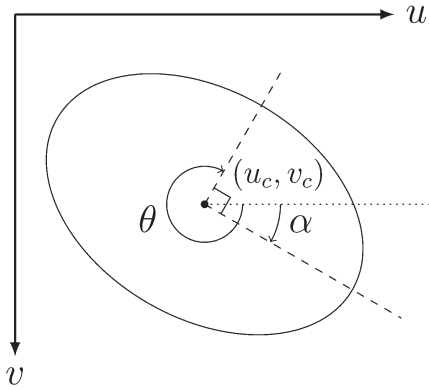


Fig. 5. Parameters of an ellipse in the image coordinate frame.

of a large number of nonelliptical contours, the ones that correspond to ellipses are all successfully detected.

Ellipse Fitting: The purpose of the ellipse fitting step is to fit an ellipse for each elliptical contour detected by the previous step. There are a variety of ellipse fitting algorithms in the literature (see, for example, [7] and [8]). In our work, we choose the ellipse fitting function, i.e., *fitEllipse*, implemented in OpenCV. Experiments show that this function is efficient and accurate enough for our work. As demonstrated in Fig. 4(c), all ellipses in the image can be successfully obtained.

Postprocessing Based on Algebraic Error: The purpose of the postprocessing step is to calculate the algebraic error between each fitted ellipse and the corresponding contour. If the algebraic error is larger than a threshold, the fitted ellipse will be classified as a false detection.

The algebraic error between a contour and its fitted ellipse is defined as follows. Let (u, v) be the coordinate of a pixel point in the image coordinate frame. Denote (u_c, v_c) as the coordinate of the ellipse center. Let a and b be the lengths of the semimajor and semiminor axes of the fitted ellipse, respectively. Denote α as the angle between the major axis and the u -axis. The angle α is positive clockwise based on the right-hand rule. All the preceding parameters are illustrated in Fig. 5. The equation of the ellipse in the image frame is expressed as

$$\frac{[(u - u_c) \cos \alpha + (v - v_c) \sin \alpha]^2}{a^2} + \frac{[(u - u_c) \sin \alpha - (v - v_c) \cos \alpha]^2}{b^2} = 1. \quad (4)$$

Based on (4), the algebraic error between a contour and the fitted ellipse is defined as

$$e_{\text{alg}} \triangleq \frac{1}{n} \sum_{i=1}^n \left| \frac{[(u_i - u_c) \cos \alpha + (v_i - v_c) \sin \alpha]^2}{a^2} + \frac{[(u_i - u_c) \sin \alpha - (v_i - v_c) \cos \alpha]^2}{b^2} - 1 \right| \quad (5)$$

where (u_i, v_i) with $i = 1, \dots, n$ are the coordinates of the pixel points in the contour.

A fitted ellipse will be classified as a false detection if its algebraic error is larger than a threshold. Since most of the de-

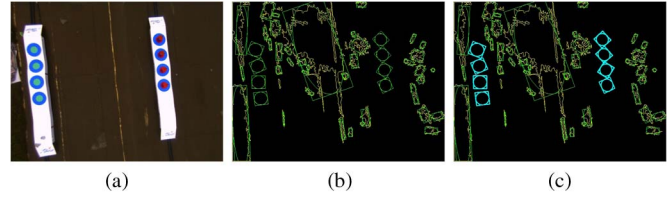


Fig. 6. Example to demonstrate the postprocessing. (a) Color image. (b) Fitted ellipses for all contours (contours with too few points are excluded). (c) Good ellipses detected based on the algebraic error.

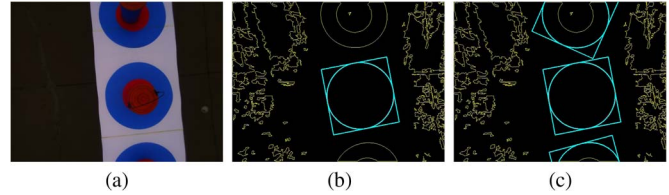


Fig. 7. Example to demonstrate the detection of partially occluded ellipses. (a) Color image. (b) Whole ellipse. (c) All ellipses.

tected contours are not perfectly elliptical, the threshold cannot be too small. According to our experience, the threshold 0.1 can give a satisfactory performance. Fig. 6 shows an example to demonstrate the effectiveness of the postprocessing step. Note that the ellipse detection in Fig. 6 is merely based on ellipse fitting and postprocessing, whereas preprocessing is not used. As shown in Fig. 6(b), ellipses are fitted for all of the contours. Fig. 6(c) shows that all ellipses are successfully detected in the presence of a large number of nonelliptical contours.

There are several issues to note in the implementation of the postprocessing in OpenCV (the version we used is 2.3.1). First, as shown in Fig. 5, the u - and v -axes of the image coordinate frame defined in OpenCV are pointing rightward and downward, respectively. The origin of the image frame is located at the upper left corner of the image. Second, an ellipse is described by a structure named *RotatedRect* in OpenCV. We can easily but may not directly obtain all the parameters required to compute the algebraic error from *RotatedRect*. For example, the angle returned by *RotatedRect* is not α ; instead, it is θ , as shown in Fig. 5. A series of experiments suggest that θ is the angle between the minor axis of the ellipse and the u -axis of the image frame. Moreover, the angle θ is positive clockwise, and it is always in the interval $[135^\circ, 315^\circ)$. As a result, the angle α can be calculated by $\alpha = \pi/2 + \theta$. It is worth noting that adding $k\pi$ to α with k as an integer does not affect the algebraic error in (5).

B. Special Case: Partially Occluded Circle

In practice, it is common that only part of a circle can be seen by the camera due to occlusion or limited field of view. The contour corresponding to the occluded circle will not be elliptical (see, for example, the top and the bottom contours in Fig. 7). We next propose an efficient method to detect partially occluded circles. First, compute the convex hull for each of the contours that are not classified as whole ellipses. Second, fit an ellipse for the convex hull and then compute the algebraic

error between the fitted ellipse and the convex hull (not the contour). If the algebraic error is too large, the fitted ellipse will be eliminated; otherwise, it can be classified as a partially occluded ellipse.

Fig. 7 shows an example to demonstrate the preceding convex-hull-based method. In this example, there exist three ellipses in the image. The middle is a whole ellipse and can be successfully detected by the pre- and postprocessing procedures. The top and the bottom ones are partially occluded. They can be successfully detected by the convex-hull-based method.

C. Summary of the Ellipse Detection Algorithm

The proposed ellipse detection algorithm is summarized in Algorithm 1. Several important remarks are given here. First, Algorithm 1 can detect both whole and partial ellipses. Second, as demonstrated in Figs. 4 and 6, either pre- or postprocessing together with ellipse fitting can independently detect ellipses. It is not necessarily required to use both of the pre- and postprocessing in practice. However, it is recommended to do that in order to improve the robustness of the algorithm unless the computational resource is extremely limited. Third, the pre- and postprocessing procedures have their own characteristics. For example, the preprocessing based on the AMIs can only detect whole ellipses, whereas the postprocessing based on the algebraic error can handle both whole and partial ellipses. However, the postprocessing is not able to judge whether an ellipse is whole or partial.

Algorithm 1 Robust Real-time Ellipse Detection

- 1: **Preparation:** Detect contours in the image using, for example, edge detection or color thresholding.
- 2: **Preprocessing:** For each contour, compute its central moments (OpenCV function *moments*), and then calculate I_1 , I_2 , and I_3 as given in (1). Compare the calculated I_1 , I_2 , and I_3 with the theoretical values in (3). If

$$\begin{aligned} |I_1 - 1/(16\pi^2)| &< 0.0003 \\ |I_2| < 0.0000001 \quad |I_3| < 0.000001 \end{aligned} \quad (6)$$

then the contour is a good candidate for a whole ellipse.

- 3: **Ellipse fitting:**

- a) For each contour that satisfies (6), fit an ellipse for it (OpenCV function *fitEllipse*).
- b) For each contour that does not satisfy (6), get the convex hull of the contour (OpenCV function *convexHull*) and then fit an ellipse for the convex hull.

- 4: **Postprocessing:**

- a) For each contour that satisfies (6), compute its algebraic error e_{alg} as defined in (5). If $|e_{\text{alg}}| < 0.1$, the contour corresponds to a *whole* ellipse.
 - b) For each contour that does not satisfy (6), compute its algebraic error e_{alg} between the fitted ellipse and the convex hull. If $|e_{\text{alg}}| < 0.1$, the contour corresponds to a *partially occluded* ellipse.
-

IV. ELLIPSE TRACKING

Multiple ellipses may be detected in an image, but we may be only interested in one of them. Then, an initialization procedure is required to intelligently choose a target from all the detected ellipses. The initialization procedure depends on the specific cargo transfer task. For example, if the helicopter is going to load a cargo, the simplest initialization procedure is to randomly choose an ellipse in the image as long as the ellipse encircles a cargo. Once the target ellipse has been chosen, it needs to be tracked over the image sequence such that the position of the corresponding circle can be continuously estimated.

There are several challenges for tracking an ellipse in the competition. First, the areas enclosed by the circles are exactly the same in terms of color and shape. As a result, pattern matching based only on color, shape, or feature points is not able to distinguish the target ellipse. Second, the circles are located near each other. Hence, the frame rate of the image sequence must be high in order to track the target ellipse continuously. This requires the tracking algorithm to be sufficiently efficient.

We choose the efficient image tracking method CAMShift [21] as the core of our tracking algorithm. Once the target ellipse has been chosen, the color probability distribution of a search window enclosing the target ellipse will be calculated. In the next frame, the algorithm will search a region whose color probability distribution is similar to the one of the search window.

The ellipse tracking algorithm is summarized in Algorithm 2. Note that the visual features of the ellipse area such as its scale and shape may dynamically vary when the camera and the target circle are relatively moving. In order to track the target robustly, the histogram of the target ellipse area must be continuously updated. We design a histogram update law (7): The histogram for frame $k + 1$ is a convex combination of the one for frame k and the one for the area enclosed by the target ellipse in frame k . If $w = 1$, there will be no histogram update; if $w = 0$, the history of the histogram is totally discarded. If the update of the histogram is too slow (w is close to 1), the algorithm cannot track the target when its shape, scale, or color keeps changing. If the update of the histogram is too fast (w is close to 0), the tracking may be unstable. According to a series of experiments, the value $w = 0.95$ can give a robust tracking performance even if the location, scale, or shape of the ellipse area dynamically varies.

Algorithm 2 Ellipse Tracking based on CAMShift

For frame $k + 1$:

- 1: Update the tracking window: choose the initial tracking window for CAMShift as the tangent rectangle of the target ellipse in frame k .
- 2: Update the histogram: denote h_k and h_{k+1} as the histogram used for tracking in frame k and $k + 1$, respectively. Let \tilde{h}_k be the histogram of the target ellipse area in frame k . Then, update h_{k+1} by

$$h_{k+1} = wh_k + (1 - w)\tilde{h}_k \quad (7)$$



Fig. 8. Example to demonstrate ellipse tracking over consecutive images. The target ellipse is the top one (highlighted in green). The yellow ellipse is the target area returned by CAMShift.

where $w \in [0, 1]$ is a weight factor.

- 3: Compute the back projection image based on the updated histogram (OpenCV function *calcBackProject*).
- 4: Obtain the final tracking window by CAMShift (OpenCV function *CamShift*). This can be done by substituting the back projection image and the initial tracking window into the CAMShift algorithm.
- 5: Find the ellipse that is located closest to the final tracking window given by CAMShift.

Fig. 8 shows an example to demonstrate the robustness of the tracking algorithm. As can be seen, the target ellipse is tracked robustly, although its location, scale, and shape keep changing in the image sequence. Note that the target circle is only partially observable in the first five images. However, it can still be tracked robustly due to the histogram update law (7).

V. SINGLE-CIRCLE-BASED POSITION ESTIMATION

This section presents an efficient algorithm to estimate the relative position of a single circle.

A. Circle Center Estimation From Four Point Correspondences

We first introduce three coordinate frames involved in the estimation problem: world frame, camera frame, and image frame. Without loss of generality, we set the world frame in the following way: the origin of the world frame coincides with the circle center, and the Z -axis of the world frame is orthogonal to the plane Π_1 that contains the circle. As a result, the coordinate of the circle center in the world frame is $(0,0,0)$, and the Z -component of any point on the circle is zero. The camera frame has its origin located at the camera center, and its z -axis is orthogonal to the image plane Π_2 . The image frame is shown in Fig. 5.

The problem that we are going to solve is summarized as follows.

Problem 1: Consider one single circle in the world frame. Its perspective projection, an ellipse, in the image frame has been detected. Suppose that the intrinsic matrix of the camera is known. Given the radius of the circle and all parameters of the ellipse, estimate the coordinate of the circle center in the camera frame.

Denote $p = [X, Y, Z]^T \in \mathbb{R}^3$ and $q = [x, y, z]^T \in \mathbb{R}^3$ as the coordinates of any point in the world frame and the camera frame, respectively. Let $R \in \mathbb{R}^{3 \times 3}$ and $T \in \mathbb{R}^3$ be the rotational and translational transformation from the world frame to the camera frame, respectively. Then, we have $q = Rp + T$. Use the subscript c to denote the coordinate of the circle center.

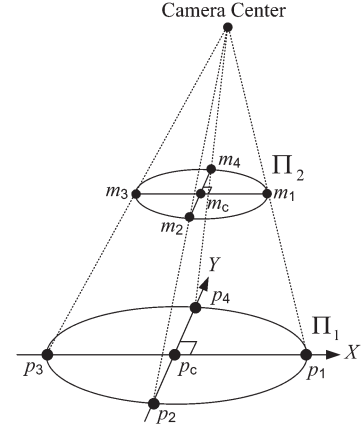


Fig. 9. Perspective projection of a circle and the four point correspondences.

Since the coordinate of the circle center in the world frame is $p_c = [0, 0, 0]^T$, the coordinate of the circle center in the camera frame would be $q_c = Rp_c + T = T$. Therefore, the translation T is the quantity to be estimated.

In order to estimate T , we next identify four sets of corresponding points on the ellipse and the circle, respectively. Let m_1, m_2, m_3 , and m_4 be the four vertices of the ellipse in the image frame (see Fig. 9). Given the parameters of the ellipse such as (u_c, v_c) , a , b , and α , as shown in Fig. 5, the coordinates of the four points can be easily calculated by $m_1 = (u_c + a \cos \alpha, v_c + a \sin \alpha)$, $m_2 = (u_c - b \sin \alpha, v_c + b \cos \alpha)$, $m_3 = (u_c - a \cos \alpha, v_c - a \sin \alpha)$, and $m_4 = (u_c + b \sin \alpha, v_c - b \cos \alpha)$. Let p_1, p_2, p_3 , and p_4 be the four corresponding points on the circle. Point p_i corresponds to m_i ($i = 1, \dots, 4$). The following assumption on p_1, \dots, p_4 is important to our algorithm.

Assumption 1: The four points p_1, \dots, p_4 are evenly distributed on the circle, which means p_1, p_c , and p_3 are collinear; p_2, p_c , and p_4 are collinear; and the line p_1p_3 is perpendicular to p_2p_4 .

Recall that the rotation R from the world frame to the camera frame is not of our interest. As a result, we can set the orientation of the world frame arbitrarily without affecting T . Since p_1, \dots, p_4 are evenly distributed on the circle, as stated in Assumption 1, we can set the world frame in a way that the four points are located at the X - and Y -axes of the world frame, respectively (see Fig. 9). If the diameter of the circle is given as r , then we have $p_1 = (r, 0, 0)$, $p_2 = (0, -r, 0)$, $p_3 = (-r, 0, 0)$, and $p_4 = (0, r, 0)$. Thus, we successfully obtain four sets of point correspondences $\{m_1, p_1\}$, $\{m_2, p_2\}$, $\{m_3, p_3\}$, and $\{m_4, p_4\}$, which can be consequently substituted to any pose estimation algorithms [38]–[40] to calculate T . The single-circle-based relative position estimation algorithm is summarized in Algorithm 3.

Algorithm 3 Single-Circle-Based Relative Position Estimation

- 1: Obtain the coordinates of the four vertices of the ellipse in the image frame:

$$m_1 = (u_c + a \cos \alpha, v_c + a \sin \alpha)$$

$$m_2 = (u_c - b \sin \alpha, v_c + b \cos \alpha)$$

$$m_3 = (u_c - a \cos \alpha, v_c - a \sin \alpha)$$

$$m_4 = (u_c + b \sin \alpha, v_c - b \cos \alpha).$$

- 2: Obtain the coordinates of the four corresponding points on the circle in the world frame:

$$p_1 = (r, 0, 0) \quad p_2 = (0, -r, 0)$$

$$p_3 = (-r, 0, 0) \quad p_4 = (0, r, 0).$$

- 3: Substitute the four correspondences $\{m_1, p_1\}$, $\{m_2, p_2\}$, $\{m_3, p_3\}$, and $\{m_4, p_4\}$ to a pose estimation algorithm (OpenCV function *solvePnP*) to estimate the translation T from the world frame to the camera frame. The position of the circle center in the camera frame is $q_c = T$.
-

B. Necessary and Sufficient Conditions for Assumption 1

Assumption 1 plays a key role in our proposed position estimation algorithm. The following theorem gives the necessary and sufficient condition for Assumption 1.

Theorem 1: Assumption 1 holds if and only if the ellipse center coincides with the projection of the circle center.

Proof: The proof is based on the fact that collinearity is preserved under perspective projection.

Necessity: If p_c is collinear with p_1 and p_3 , the projection of p_c is collinear with m_1 and m_3 . Analogously, the projection of p_c is also collinear with m_2 and m_4 . As a result, the projection of p_c is the intersection of the lines m_1m_3 and m_2m_4 . Denote m_c as the center of the ellipse. Since m_c is also the intersection of m_1m_3 and m_2m_4 , we have m_c coincide with the projection of p_c .

Sufficiency: In order to prove the sufficiency, we only need to prove the following: first, the lines p_1p_3 and p_2p_4 intersect at p_c ; second, the two lines p_1p_3 and p_2p_4 are perpendicular to each other. First, if m_c coincides with the projection of p_c , we have that p_1 , p_3 , and p_c are also collinear because m_1 , m_3 , and m_c are collinear. Analogously, the points p_2 , p_4 , and p_c are also collinear. As a result, the lines p_1p_3 and p_2p_4 intersect at p_c . Second, note that $\|m_1 - m_c\| = \|m_3 - m_c\|$ and $\|p_1 - p_c\| = \|p_3 - p_c\|$, where $\|\cdot\|$ denotes the Euclidean norm of a vector. If m_c is the perspective projection of p_c , it is easy to prove that p_1p_3 is parallel to m_1m_3 based on similar triangles. Analogously, it can be also proved that p_2p_4 is parallel to m_2m_4 . Thus, p_1p_3 is perpendicular to p_2p_4 because m_1m_3 is perpendicular to m_2m_4 . ■

It has been proved in [24] that the ellipse center generally does not coincide with the projection of the circle center. Next, we further show when they coincide with each other.

Theorem 2: The ellipse center coincides with the projection of the circle center if and only if the image plane is parallel to the plane that contains the circle.

Proof: Necessity: If m_c coincides with the projection of p_c , we know that p_c is collinear with p_1 and p_3 . Furthermore, since $\|m_1 - m_c\| = \|m_3 - m_c\|$ and $\|p_1 - p_c\| = \|p_3 - p_c\|$, it is easy to prove that p_1p_3 is parallel to m_1m_3 based on similar triangles. It can be analogously proved that p_2p_4 is parallel to m_2m_4 . As a result, the image plane Π_2 is parallel to the plane Π_1 . **Sufficiency:** See the paragraph below equation (16) in [24, Section 2.3]. ■

Remark 1: The sufficiency of Theorem 2 is a well-known result proved by [24]. The contribution of Theorem 2 is that it shows that the parallel condition is also necessary for the ellipse center coinciding with the projection of the circle center.

Theorems 1 and 2 clearly indicate that Assumption 1 is valid if and only if the image plane is parallel to the plane that contains the circle. In many practical cargo transfer tasks, the cargo usually is required to be placed on a horizontal plane. In order to satisfy the parallel assumption, the camera can be mounted on a pan-tilt mechanism, which can compensate the dynamic rotation of the helicopter such that the image plane is also horizontal. As pan-tilt mechanisms (or called gyrostabilized cameras) are common for UAVs nowadays, the parallel assumption does not significantly impact the practical applicability of Algorithm 3.

VI. EXPERIMENTAL AND COMPETITION RESULTS

A. Experimental Results for Algorithm 3

In our vision system, the camera is mounted on a pan-tilt mechanism such that the image plane is controlled to be horizontal and hence parallel to the circle plane. However, due to time delay or measurement noise of the gyro, there may exist an error on the order of 1° in pan-tilt control. As a result, the image plane may not be perfectly parallel to the circle plane. We next show a numerical experimental result to see if Algorithm 3 can still work well when the parallel assumption is not strictly satisfied.

Consider that a circle with the radius as 1 m is placed in front of a camera such that the circle center coincides with the principal axis of the camera. The image of the circle, an ellipse, on the normalized image plane can be numerically calculated. We choose the EPnP pose estimation algorithm [40] to estimate the circle center based on Algorithm 3. Fig. 10 shows the estimation error when the angle between the image and circle planes is nonzero. It can be seen that the estimation error is less than 0.05 m when the angle is less than 10° . The numerical results suggest that the small error in pan-tilt control does not impact the estimation accuracy significantly.

We next present real experimental results in a motion capture system named Vicon to verify the accuracy of Algorithm 3. The experimental setup is shown in Fig. 11. The Vicon system is used to obtain the ground truth of the range from the circle center to the camera center. The diameter of the target circle is 1 m. The images of the target circle are given in Fig. 12. The estimation results by Algorithm 3 are given in Table I. As can

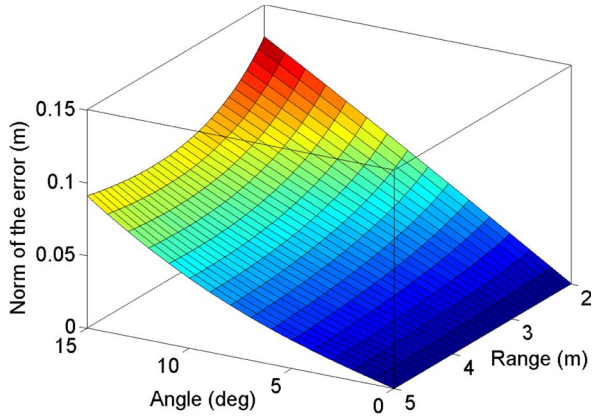


Fig. 10. Estimation error of Algorithm 3 caused by the nonzero angle between the circle and image planes.

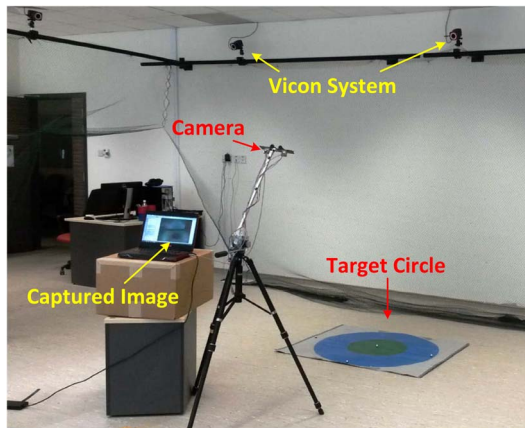


Fig. 11. Experiment setup in a Vicron system to verify Algorithm 3.

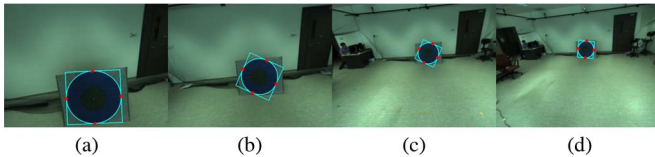


Fig. 12. Images (640×480 pixels) captured in the experiment.

TABLE I
POSITION ESTIMATION RESULTS USING THE IMAGES IN FIG. 12

Image	Circle center q_c by vision	Range $\ q_c\ $ by vision	Range $\ q_c\ $ by Vicon	Abs error of $\ q_c\ $
(a)	(-0.566, 0.074, 1.566) m	1.667 m	1.650 m	0.017 m
(b)	(-0.278, 0.160, 2.361) m	2.383 m	2.363 m	0.020 m
(c)	(0.644, 0.507, 4.120) m	4.201 m	4.218 m	0.017 m
(d)	(1.044, 0.253, 5.194) m	5.304 m	5.253 m	0.051 m

be seen, the proposed algorithm can give accurate estimates, although the circle is not strictly parallel to the image plane.

B. Competition Results

Successfully completing the competition tasks is the strongest evidence for the efficiency, accuracy, and robustness of the vision system. All of the previous images used to demonstrate the ellipse detection and tracking algorithms in this paper are actual images captured by the onboard camera in the competition. For more data of the competition, please see the video at <http://youtu.be/GSeafBsASTs>.

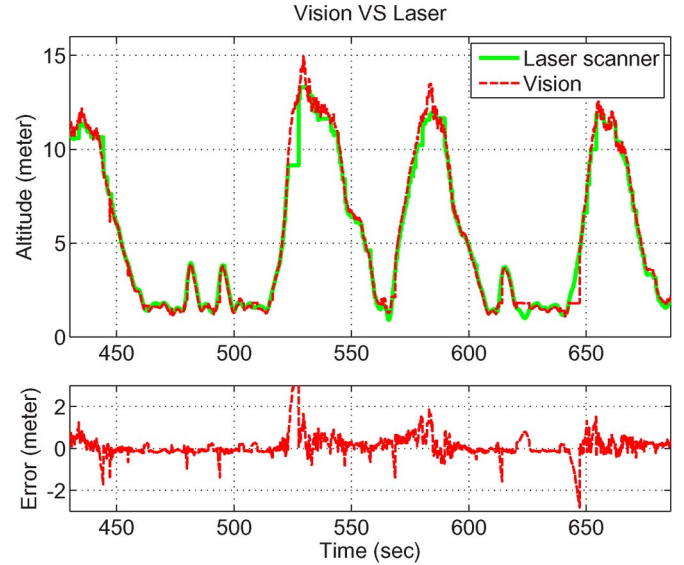


Fig. 13. Altitude estimated by vision and the altitude measured by the laser scanner.

Although the helicopter is equipped with a GPS, the accuracy of the GPS (± 5 m in the horizontal direction) is much worse than that of the vision system. As a result, the GPS data cannot be used as the ground truth to verify the accuracy of the vision estimation. A laser scanner is mounted at the back of the helicopter to measure the altitude of the helicopter [1]. Since the laser measurement is very accurate, it can be treated as the ground truth of the altitude. Although the vision system is not designed to estimate the altitude, it is able to provide altitude estimation. Then, we can compare the altitude estimates given by the vision and the laser systems to quantitatively verify the accuracy of the vision system.

Fig. 13 shows the altitude estimation results given by the vision and laser systems. When the altitude is smaller than 10 m, the error of the vision estimation is generally less than 1 m. When the helicopter descends to about 2 m, the error is less than 0.2 m. It is noticed that there exist a small number of spikes in the error of the vision estimation, for example, at 525 and 645 s, etc. That is because either the vision or the laser system fails to give valid data at those time instances.

C. Efficiency Test

The efficiency of the algorithms has been tested on a laptop with the CPU as Intel Core i5-2520M 2.5 GHz. The computational capability of the laptop is similar to the onboard vision computer whose specifications are given in Section II. The proposed algorithms are used to process 100 consecutive images captured by the onboard camera in the competition. The size of each image is 640×480 pixels. The average time consumption of each algorithm is given in Table II. As can be seen, the algorithms for ellipse detection, ellipse tracking, and single-circle-based position estimation are very efficient.

VII. CONCLUSION AND FUTURE WORK

We have designed and implemented a robust real-time vision system for autonomous cargo transfer by an unmanned helicopter. The vision system has been proved as efficient, accurate,

TABLE II
AVERAGE TIME CONSUMPTION OF EACH
PROCEDURE IN THE VISION SYSTEM

Procedure	Average Time Consumption
Undistortion	0.0240 sec
RGB to HSV	0.0093 sec
Color Thresholding	0.0042 sec
Contour Detection	0.0011 sec
Ellipse Detection	0.0012 sec
Ellipse Tracking	0.0084 sec
Single-Circle-based Position Estimation	0.0030 sec

and robust by the competition and experimental results. In practice, one approach to improve the robustness of the vision system is to implement the circle in some special ways instead of painting such that the circle can be detected more robustly.

REFERENCES

- [1] F. Wang *et al.*, "Guidance, navigation and control of an unmanned helicopter for automatic cargo transportation," presented at the 33rd Chinese Control Conf., Nanjing, China, Jul. 2014.
- [2] P. Liu, X. Dong, F. Wang, and B. M. Chen, "Development of a comprehensive software system for cargo transportation by unmanned helicopters," to appear in *Proc. 2014 Int. Conf. Intell. Unmanned Syst.*, Montreal, QC, Canada, Sep. 29–Oct. 1, 2014.
- [3] M. Bernard and K. Kondak, "Generic slung load transportation system using small size helicopters," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, 2009, pp. 3258–3264.
- [4] M. Bisgaard, A. la Cour-Harbo, and J. D. Bendtsen, "Adaptive control system for autonomous helicopter slung load operations," *Control Eng. Pract.*, vol. 18, no. 7, pp. 800–811, Jul. 2010.
- [5] M. Orsag, C. Korpela, and P. Oh, "Modeling and control of MM-UAV: Mobile manipulating unmanned aerial vehicle," *J. Intell. Robot. Syst.*, vol. 69, no. 1–4, pp. 227–240, Jan. 2013.
- [6] J. Thomas, J. Polin, K. Sreenath, and V. Kumar, "Avian-inspired grasping for quadrotor micro UAVs," in *Proc. ASME Int. Design Eng. Techn. Conf.*, 2013, pp. 1–9.
- [7] A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least square fitting of ellipses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 476–480, May 1999.
- [8] S. J. Ahn, W. Rauh, and H.-J. Warnecke, "Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola," *Pattern Recognit.*, vol. 34, no. 12, pp. 2283–2303, Dec. 2001.
- [9] R. A. McLaughlin, "Randomized Hough transform: Improved ellipse detection with comparison," *Pattern Recognit. Lett.*, vol. 19, no. 3/4, pp. 299–305, Mar. 1998.
- [10] S.-C. Zhang and Z.-Q. Liu, "A robust, real-time ellipse detector," *Pattern Recognit.*, vol. 38, no. 2, pp. 273–287, Feb. 2005.
- [11] P. L. Rosin, "Measuring shape: Ellipticity, rectangularity, and triangularity," in *Proc. 15th Int. Conf. Pattern Recognit.*, Barcelona, Spain, 2000, pp. 952–955.
- [12] F. Mokhtarian and S. Abbasi, "Shape similarity retrieval under affine transforms," *Pattern Recognit.*, vol. 35, no. 1, pp. 31–41, Jan. 2002.
- [13] Y. Wang and E. K. Teoh, "2D affine-invariant contour matching using B-spline model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1853–1858, Oct. 2007.
- [14] J. Flusser and T. Suk, "Pattern recognition by affine moment invariants," *Pattern Recognit.*, vol. 26, no. 1, pp. 167–174, Jan. 1993.
- [15] C. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.
- [16] M. Isard and A. Blake, "CONDENSATION—Conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28, 1998.
- [17] Q. Zhou and J. Aggarwal, "Object tracking in an outdoor environment using fusion of features and cameras," *Image Vis. Comput.*, vol. 24, pp. 1244–1255, 2006.
- [18] F. Lin, X. Dong, B. M. Chen, K. Y. Lum, and T. H. Lee, "A robust real-time embedded vision system on an unmanned rotorcraft for ground target following," *IEEE Trans. Ind. Electron.*, vol. 59, no. 2, pp. 1038–1049, Feb. 2012.
- [19] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *Intel Technol. J.*, vol. 2, no. 2, pp. 12–21, 1998.
- [20] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, pp. 564–577, 2003.
- [21] J. G. Allen, R. Y. D. Xu, and J. S. Jin, "Object tracking using CamShift algorithm and multiple quantized feature spaces," in *Proc. Pan-Sydney Area Workshop Vis. Inf. Process.*, Darlinghurst, Australia, 2004, pp. 3–7.
- [22] C. Martínez, I. F. Mondragón, M. A. Olivares-Méndez, and P. Campoy, "On-board and ground visual pose estimation techniques for UAV control," *J. Intell. Robot. Syst.*, vol. 61, no. 1–4, pp. 301–320, Jan. 2011.
- [23] F. Lin *et al.*, "Development of an unmanned coaxial rotorcraft for the DARPA UAVForge Challenge," *Unmanned Syst.*, vol. 1, no. 2, pp. 211–245, Oct. 2013.
- [24] J. Heikkilä and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Juan, Puerto Rico, 1997, pp. 1106–1112.
- [25] J.-S. Kim, P. Gurdjos, and I.-S. Kweon, "Geometric and algebraic constraints of projected concentric circles and their applications to camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 637–642, Apr. 2005.
- [26] G. Jiang and L. Quan, "Detection of concentric circles for camera calibration," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, Beijing, China, 2005, pp. 333–340.
- [27] D. Eberli, D. Scaramuzza, S. Weiss, and R. Siegwart, "Vision based position control for MAVs using one single circular landmark," *J. Intell. Robot. Syst.*, vol. 61, no. 1–4, pp. 495–512, Jan. 2011.
- [28] S. Lange, N. Sunderhauf, and P. Protzel, "A vision based onboard approach for landing and position control of an autonomous multirotor UAV in GPS-denied environments," in *Proc. Int. Conf. Adv. Robot.*, Munich, Germany, Jun. 2009, pp. 1–6.
- [29] G. Wang, J. Wu, and Z. Ji, "Single view based pose estimation from circle or parallel lines," *Pattern Recognit. Lett.*, vol. 29, no. 7, pp. 977–985, May 2008.
- [30] S. Yang, S. A. Scherer, and A. Zell, "An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle," *J. Intell. Robot. Syst.*, vol. 69, no. 1–4, pp. 499–515, Jan. 2013.
- [31] Y. Hu, W. Zhao, and L. Wang, "Vision-based target tracking and collision avoidance for two autonomous robotic fish," *IEEE Trans. Ind. Electron.*, vol. 56, no. 5, pp. 1401–1410, May 2009.
- [32] J. Wang and H. Cho, "Micropeg and hole alignment using image moments based visual servoing method," *IEEE Trans. Ind. Electron.*, vol. 55, no. 3, pp. 1286–1294, Mar. 2008.
- [33] Y. Fang, X. Liu, and X. Zhang, "Adaptive active visual servoing of non-holonomic mobile robots," *IEEE Trans. Ind. Electron.*, vol. 59, no. 1, pp. 486–497, Jan. 2012.
- [34] D.-H. Park, J.-H. Kwon, and I.-J. Ha, "Novel position-based visual servoing approach to robust global stability under field-of-view constraint," *IEEE Trans. Ind. Electron.*, vol. 59, no. 12, pp. 4735–4752, Dec. 2012.
- [35] H. Lang, M. T. Khan, K.-K. Tan, and C. W. de Silva, "Developments in visual servoing for mobile manipulation," *Unmanned Syst.*, vol. 1, no. 1, pp. 143–162, Jul. 2013.
- [36] M. Defoort, T. Floquet, A. Kökösy, and W. Perruquetti, "Sliding-mode formation control for cooperative autonomous mobile robots," *IEEE Trans. Ind. Electron.*, vol. 55, no. 11, pp. 3944–3953, Nov. 2008.
- [37] P. C. Niefeldt *et al.*, "Enhanced UAS surveillance using a video utility metric," *Unmanned Syst.*, vol. 1, no. 2, pp. 277–296, Oct. 2013.
- [38] R. M. Haralick *et al.*, "Pose estimation from corresponding point data," *IEEE Trans. Syst., Man, Cybern.*, vol. 19, no. 6, pp. 1426–1446, Nov./Dec. 1989.
- [39] L. Quan and Z. Lan, "Linear n-point camera pose determination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 8, pp. 774–780, Aug. 1999.
- [40] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate $O(n)$ solution to the PnP problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, pp. 155–166, Feb. 2009.



Shiyu Zhao (M'09) received the B.Eng. and M.Eng. degrees from Beijing University of Aeronautics and Astronautics, Beijing, China, in 2006 and 2009, respectively. He is currently working toward the Ph.D. degree in the NUS Graduate School for Integrative Sciences and Engineering, National University of Singapore, Singapore.

His research interests are control of multi-agent systems and robot vision.



Zhangyuan Hu received the B.Eng. degree in information engineering from The Chinese University of Hong Kong, Hong Kong, in 2014.

In summer 2013, he was a Visiting Researcher with the Temasek Laboratories, National University of Singapore. He is currently an IT Graduate Trainee with the Innovation Centre, Cathay Pacific Airways Ltd., Hong Kong. His main research interests are target tracking, target recognition, and file deduplication in cloud cryptography.



Mingfeng Yin received the B.Eng. degree in automation from China University of Mining and Technology, Xuzhou, China. He is currently working toward the Ph.D. degree in control science and engineering at Nanjing University of Science and Technology, Nanjing, China.

His research interests include object detection and tracking and pattern recognition.



Kevin Z. Y. Ang received the B.Eng. degree in mechanical and production engineering from Nanyang Technological University, Singapore, in 2006. He is currently working toward the Ph.D. degree in electrical and computer engineering at the National University of Singapore, Singapore, under a DSO Ph.D. Research Award

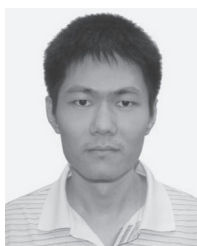
After completing his undergraduate degree, he joined the Singapore Armed Forces as a Maintenance Engineering Officer, fulfilling his scholarship bond. His current research interests

are obstacle detection and unmanned aerial vehicle navigation in urban environments.



Peidong Liu received the B.Eng. degree (*with first-class honors*) in electrical engineering from the National University of Singapore, Singapore, in 2012.

He is currently a Research Engineer with the Department of Electrical and Computer Engineering, National University of Singapore. His main research interests are unmanned aerial vehicles, robotics, and real-time embedded software systems.



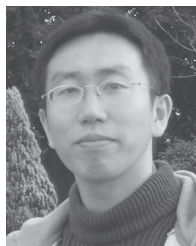
Fei Wang (S'09) received the B.Eng. and Doctoral degrees from the National University of Singapore (NUS), Singapore, in 2009 and 2014, respectively.

He is currently a Research Scientist with the Control Science Group, Temasek Laboratories, NUS. His main research areas include unmanned aerial vehicle (UAV) modeling and control and navigation systems without GPS for UAVs.



Xiangxu Dong received the B.S. degree from Xiamen University, Xiamen, China, in 2006. He is currently working toward the Ph.D. degree at the National University of Singapore, Singapore.

He is also currently a Research Scientist with the Temasek Laboratories, National University of Singapore. His research interests include real-time software systems, formation flight control, and unmanned aerial vehicles.



Feng Lin (S'09–M'11) received the B.Eng. degree in computer science and control and the M.Eng. degree in system engineering from Beihang University, Beijing, China, in 2000 and 2003, respectively, and the Ph.D. degree in computer and electrical engineering from the National University of Singapore, Singapore, in 2011.

He is currently a Senior Research Scientist with the Temasek Laboratories, National University of Singapore. His main research interests are unmanned aerial vehicles, vision-aided control and navigation, target tracking, robot vision, and embedded vision systems.



Ben M. Chen (S'89–M'92–SM'00–F'07) received the B.S. degree in mathematics from Xiamen University, Xiamen, China, in 1983, the M.S. degree in electrical engineering from Gonzaga University, Spokane, WA, USA, in 1988, and the Ph.D. degree in electrical and computer engineering from Washington State University, Pullman, WA, 1991.

He is currently a Professor and the Director of the Control, Intelligent Systems and Robotics Area of the Department of Electrical and Computer Engineering with the National University of Singapore, Singapore, where he is also the Head of the Control Science Group, Temasek Laboratories. He has authored/coauthored ten research monographs, including *H₂ Optimal Control* (Prentice-Hall, 1995), *Robust and H_∞ Control* (Springer, 2000), *Hard Disk Drive Servo Systems* (Springer, 1st Edition, 2002; 2nd Edition, 2006), *Linear Systems Theory* (Birkhäuser, 2004), *Unmanned Rotorcraft Systems* (Springer, 2011), and *Stock Market Modeling and Forecasting* (Springer, 2013). His current research interests are in systems and control, unmanned aerial systems, and financial market modeling.

Dr. Chen currently serves as Editor-in-Chief of *Unmanned Systems* and Deputy Editor-in-Chief of *Control Theory and Technology*. He had served on the editorial boards of a number of journals, including the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, *Systems and Control Letters*, and *Automatica*.



Tong H. Lee (M'90) received the B.A. degree (*with first-class honors*) from the University of Cambridge, Cambridge, U.K., in 1980, the M.Eng. degree from the National University of Singapore (NUS), Singapore, in 1985, and the Ph.D. degree from Yale University, New Haven, CT, USA, in 1987.

He is currently a Professor in the Department of Electrical and Computer Engineering and the NUS Graduate School at NUS, where he served as Vice-President (Research). He has coauthored five research monographs (books) and is the holder four patents (two of which are in the technology area of adaptive systems, and the other two are in the area of intelligent mechatronics). The 2013 Thomson Reuters citation report includes him in the list of highly cited researchers in engineering. His research interests are in the areas of adaptive systems, knowledge-based control, intelligent mechatronics, and computational intelligence.

Dr. Lee is currently an Associate Editor for the IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS; IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS; *Control Engineering Practice* (an IFAC journal); and the *International Journal of Systems Science* (Taylor and Francis, London). In addition, he is the Deputy Editor-in-Chief of *Mechatronics*. He was a recipient of the Cambridge University Charles Baker Prize in Engineering, the 2004 ASCC (Melbourne) Best Industrial Control Application Paper Prize, the 2009 IEEE ICMA Best Paper in Automation Prize, the 2009 ASCC Best Application Paper Prize, and the 2013 ACA Wook Hyun Kwon Education Prize. He has been a repeat winner of the NUS Faculty of Engineering Engineering Educator Award and is currently on the Engineering Educator Honour Roll.