# A LOW-COST POSITION AND VELOCITY ESTIMATION SYSTEM FOR MULTIROTOR AERIAL VEHICLES USING A STATIC KINECT SENSOR

Luiz Manoel Santos Santana*, Davi Antônio dos Santos*, Ícaro Bezerra Viana*, Raphael Ballet*

*Instituto Tecnológico de Aeronáutica, Praça Mal. Eduardo Gomes, 50. Vila das Acácias, 12228-900. São José dos Campos, São Paulo, Brasil

Emails: luizmss@ita.br, davists@ita.br, icaro@ita.br, raphaelballet@hotmail.com

**Abstract**— Indoor estimation of position and velocity of a multirotor aerial vehicle (MAV) is frequently required for experimental evaluation of control laws and on-board navigation systems. The present work proposes a low-cost indoor position and velocity estimation system using a static Kinect sensor fixed on the ceiling and pointing downwards. The Kinect sensor consists of an RGB camera, an infrared projector and a CMOS camera. Using the RGB camera, it is possible to determine the horizontal position of the vehicle. On the other hand, the infrared projector together with the CMOS camera provide depth images which allows altitude estimation. The Kinect measurements are acquired in a desktop computer and processed in MATLAB. The data processing includes 3D position determination from a camera model and 3D position/velocity estimation using a Kalman filter. The proposed system is demonstrated and evaluated experimentally by means of the feedback control of an MAV.

**Keywords**— State estimation, multirotor aerial vehicles, Kinect sensor.

**Resumo**— Um sistema de estimação de posição e velocidade de multicópteros em ambiente interior é frequentemente requerido para fins de avaliação de leis de controle e algoritmos de navegação. O presente trabalho propõe um tal sistema de baixo custo usando um sensor Kinect fixado ao teto e apontado verticalmente para baixo. O sensor Kinect é composto de uma câmera RGB, um emissor de sinal infravermelho e uma câmera CMOS. Mediante a câmera RGB, é possível determinar a posição horizontal do veículo. A câmera CMOS, por sua vez, provê uma imagem de profundidade baseada no tempo de viagem do sinal infravermelho emitido. Com essa última imagem, é possível determinar a altitude de veículo. As medidas do Kinect são adquiridas num computador e processadas em MATLAB. O processamento de dados inclui a determinação de posição 3D a partir do modelo do sensor, assim como estimação de posição/velocidade 3D usando um filtro de Kalman. O sistema proposto é demonstrado e avaliado experimentalmente mediante do controle realimentado de um MAV.

**Palavras-chave**— Estimação de estados, multicópteros, sensor Kinect.

## 1 Introduction

In the last few years, the world has seen an increasing interest for unmanned aerial vehicles (UAV) in both industry and academia. In particular, small multirotor aerial vehicles (MAV) have become very popular due to their simplicity, low cost, and capability to fly in confined areas. These vehicles are used in several applications, such as precision agriculture (Zhang and Kovacs, 2012), disaster monitoring (Alvissalim et al., 2012), search and rescue (Naidoo et al., 2011), surveillance (Grover, 2015), powerline inspection (Luque-Vega et al., 2014), aerial photography (Widodo and Sunardi, 2014), and building inspection (Emelianov et al., 2014; Bulgakov et al., 2014).

Many papers on control and navigation of MAVs have appeared in the literature in the last five years (Prado and dos Santos, 2014; Viana et al., 2015a; Viana et al., 2015b; dos Santos et al., 2013; Crowther et al., 2011; Mahony et al., 2012; Yan et al., 2013). In such works, experimental evaluation is always required to assess the performance of the proposed methods. In particular, indoor flight tests are more common for allowing an easier control of the flight environment. Some consolidated research groups have used expensive commercial tracking systems based on infrared cameras (Campbell et al., 2012; Ducard and D'Andrea, 2009; Zhou and Schwager, 2014).

The recent cost reduction of the so-called time-of-flight cameras has encouraged many research applications. In particular, the Microsoft Kinect sensor appears in many of those applications. Examples of Kinect applications are found in estimation of robot's 3D position (Siradjuddin et al., 2012; El-laithy et al., 2012; Ganganath and Leung, 2012), in 3D object tracking (Chow et al., 2012), in medicine (Clark et al., 2012), and in education (Hsu, 2011), just to cite a few.

The present paper proposes a low-cost position and velocity estimation system for indoor evaluation of MAV guidance and navigation methods. The system consists of a Kinect sensor fixed on the ceiling and pointing downwards, and a computer for data processing. The MAV under test carries a circular red mark for providing contrast with respect to the floor (the background). The Kinect images are processed in MATLAB using color and circle segmentation for 3D position determination, as well as a Kalman filter for position smoothing and 3D velocity estimation. The proposed system is evaluated experimentally through static and flight tests with an MAV.

The remaining text is organized in the following manner. Section 2 details the main components of the proposed estimation system. Section 3 details the implementation of the system. Section 4 evaluates the performance of the proposed system experimentally. Finally, Section 5 concludes the paper.

## 2 The System Components

The proposed low-cost 3D position/velocity estimation system consists of three main hardware components: a Kinect sensor fixed on the ceiling and pointing downwards; a ground computer for image processing and state estimation; and the object to be tracked, which, in particular, is an MAV. Figure 1 illustrates the overall system.

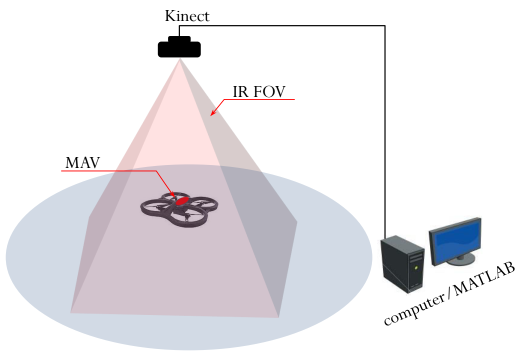


Figure 1: The overall position/velocity estimation system.

The following subsections describe the main components of the system.

### *2.1 Kinect Sensor*

The Kinect sensor is a low-cost time-of-flight camera offered by Microsoft for the XBox's video game consoles and for Windows operating system. It combines an RGB sensor with a depth sensor. The latter consists of an infrared (IR) projector and a monochrome CMOS sensor (Nakamura, 2011). In our estimation system, it provides the raw data sufficient for determining the 3D position of a red mark placed on the top of an MAV.

The Windows version of the Kinect sensor is adopted in the present work. It is installed on the ceiling at a height $h$ from the floor and pointing downwards vertically. Its field of view (FOV) is of $53^o$ about $X_{\mathrm{C}}$ and $47^o$ about $Y_{\mathrm{C}}$.

### *2.2 Multirotor Aerial Vehicle*

The multirotor aerial vehicle (MAV) chosen for evaluating the proposed estimation system is a Parrot AR.Drone 2 quadrotor vehicle. It is a very popular low-cost MAV suitable for indoor operation. Originally, it was developed to be manually controlled by a smartphone applicative, but it can also receive attitude and throttle commands via an Wi-Fi connection from an external computer. For more details about the navigation and control technology used in the AR.Drone, we suggest to read the reference (Bristeau et al., 2011).

In the present work, a circular red mark is attached on the top of the AR.Drone, as depicted in Figure 2. In this case, there is a sufficient contrast between the vehicle and the floor.

Figure 2: The AR.Drone with a circular red mark.

### *2.3 Ground Computer*

The ground computer receives the Kinect data via an USB interface. The RGB and depth images are acquired and processed in MATLAB, using commands available in its Image Processing Toolbox. In short, from the depth image, we immediately obtain the vehicle's vertical coordinate $z$ (altitude). On the other hand, the RGB images are segmented for separating the circular red mark from the backgroud. The horizontal coordinates ($x$ and $y$) of the mark's centroid are then computed in pixels. Using a pinhole camera model, this centroid is converted to millimeters. The $x$, $y$, and $z$ are used as measurement in a Kalman filter (KF) that estimates 3D position and velocity. Finally, the ground computer delivers the KF position and velocity via UDP protocol to an external computer responsible for the automatic position control.

## 3 Position and Velocity Estimation

The implementation of the position and velocity estimation system is divided into three parts: image processing, position determination, and Kalman filtering.

### *3.1 Image Processing*

The image processing is conducted in three steps in the following sequence:

1. Conversion of the color representation from the RGB model to the HSV model.

2. Image segmentation using thresholding in the HSV space. The output of this operation is a binary image where the original red pixels are converted into white and the other pixels are converted to black.
3. Circle recognition using the Hough transform. The output of this operation is the pixel coordinates of the black circle of the binary image generated in 2.

The above image processing methods are standard. For the readers who are interested in the details about them, we suggest reading the reference (Gonzalez et al., 2009).

### *3.2 Position Determination*

Figure 3 illustrates an MAV and four Cartesian coordinate systems (CCS). The body CCS $S_{\mathrm{B}} \triangleq \{X_{\mathrm{B}}, Y_{\mathrm{B}}, Z_{\mathrm{B}}\}$ is attached to the vehicle's structure, with its origin at vehicle's center of mass (CM) and the $Z_{\mathrm{B}}$ axis perpendicular to the rotor plane and pointing upwards. The ground CCS $S_{\mathrm{G}} \triangleq \{X_{\mathrm{G}}, Y_{\mathrm{G}}, Z_{\mathrm{G}}\}$ is fixed on the ground with the origin at a known point $O$ and the $Z_{\mathrm{G}}$ axis pointing upwards, aligned with the local vertical. The camera CCS $S_{\mathrm{C}} \triangleq \{X_{\mathrm{C}}, Y_{\mathrm{C}}, Z_{\mathrm{C}}\}$ is static with respect to $S_{\mathrm{G}}$; its origin is placed at the camera optical center $C$ and its $Z_{\mathrm{C}}$ axis points downwards, parallel to the camera optical axis. Finally, the image CCS $S_{\mathrm{I}} \triangleq \{X_{\mathrm{I}}, Y_{\mathrm{I}}\}$ is parallel to the X–Y plane of $S_{\mathrm{C}}$ and is centered at the camera principal point $P$.

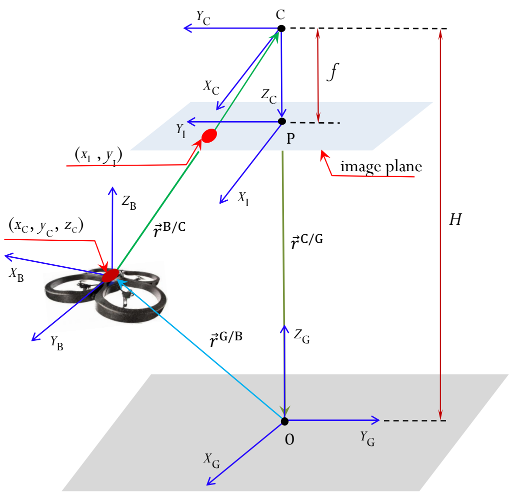


Figure 3: Cartesian coordinate systems and a pinhole camera model.

Let the geometric vectors $\vec{r}^{\,\mathrm{B/C}}$ and $\vec{r}^{\,\mathrm{B/G}}$ describe the CM position w.r.t. $S_{\mathrm{C}}$ and $S_{\mathrm{G}}$, respectively. On the other hand, let the geometric vector $\vec{r}^{\,\mathrm{C/G}}$ describe the position of $C$ w.r.t. $S_{\mathrm{G}}$. From the figure, one can write

$$\vec{r}^{\,\mathrm{B/G}} = \vec{r}^{\,\mathrm{C/G}} + \vec{r}^{\,\mathrm{B/C}}. \qquad (1)$$

Denote the algebraic vectors representing $\vec{r}^{\,\mathrm{B/G}}$ and $\vec{r}^{\,\mathrm{C/G}}$ in $S_{\mathrm{G}}$ by $\mathbf{r}_{\mathrm{G}}^{\mathrm{B/G}}$ and $\mathbf{r}_{\mathrm{G}}^{\mathrm{C/G}}$, respectively. On the other hand, denote the algebraic vector representing $\vec{r}^{\,\mathrm{B/C}}$ in $S_{\mathrm{C}}$ by $\mathbf{r}_{\mathrm{C}}^{\mathrm{B/C}}$. Moreover, denote the attitude matrix representing the attitude of $S_{\mathrm{C}}$ w.r.t. $S_{\mathrm{G}}$ by $\mathbf{D}^{\mathrm{C/G}}$. Using the above notation, one can rewrite equation (1) as

$$\mathbf{r}_{\mathrm{G}}^{\mathrm{B/G}} = \left(\mathbf{D}^{\mathrm{C/G}}\right)^{\mathrm{T}} \mathbf{r}_{\mathrm{C}}^{\mathrm{B/C}} + \mathbf{r}_{\mathrm{G}}^{\mathrm{C/G}}, \qquad (2)$$

where $\mathbf{r}_{\mathrm{G}}^{\mathrm{C/G}} \triangleq [0 \; 0 \; h]^{\mathrm{T}}$ is supposed to be known,

$$\mathbf{D}^{\mathrm{C/G}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\pi & -\sin\pi \\ 0 & \sin\pi & \cos\pi \end{bmatrix}, \qquad (3)$$

which corresponds to a rotation of $\pi$ rad about $X_{\mathrm{C}}$, and $\mathbf{r}_{\mathrm{C}}^{\mathrm{B/C}} \triangleq [x_{\mathrm{C}} \; y_{\mathrm{C}} \; z_{\mathrm{C}}]^{\mathrm{T}}$ has to be computed from the output of the image processing.

Suppose that, from the image processing, it was found that the centroid of the red mark is located at $(x_{\mathrm{I}}, y_{\mathrm{I}})$ in $S_{\mathrm{I}}$ coordinates. On the other hand, consider that $z_{\mathrm{C}}$ is directly provided by the depth camera. Therefore, using a pinhole camera model (Hartley and Zisserman, 2003), the other two components of $\mathbf{r}_{\mathrm{C}}^{\mathrm{B/C}}$ can be computed by

$$x_{\mathrm{C}} = \frac{z_{\mathrm{C}}}{f} x_{\mathrm{I}}, \qquad (4)$$

$$y_{\mathrm{C}} = \frac{z_{\mathrm{C}}}{f} y_{\mathrm{I}}, \qquad (5)$$

where $f$ is the camera focal length, which can be estimated using standard calibration packages.

### *3.3 Kalman Filtering*

In this section, we design a discrete-time Kalman filter for estimating the MAV position $\mathbf{r}_{\mathrm{G}}^{\mathrm{B/G}} \triangleq [r_x \; r_y \; r_z]^{\mathrm{T}}$ and velocity $\dot{\mathbf{r}}_{\mathrm{G}}^{\mathrm{B/G}} \triangleq [v_x \; v_y \; v_z]^{\mathrm{T}}$. To begin with, define the state vector $\mathbf{x} \triangleq [r_x \, r_y \, r_z \, v_x \, v_y \, v_z]^{\mathrm{T}}$. Now, we describe the position dynamics as the integral of a sum of velocity and velocity noise. On the other hand, we model the velocity dynamics as the integral of acceleration noise. Therefore, the continuous-time state-space model for $\mathbf{x}$ is given by

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{w}(t) \qquad (6)$$

where $\mathbf{w} \in \mathbb{R}^6$ is the continuous-time state noise vector and

$$\mathbf{A} \triangleq \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}. \qquad (7)$$

The discrete-time state equation equivalent to (7) is immediately obtained as

$$\mathbf{x}_{k+1} = \mathbf{\Phi}\mathbf{x}_k + \mathbf{w}_k, \qquad (8)$$

where $\mathbf{\Phi} \in \mathbb{R}^{6\times6}$ is the state transition matrix given by

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{I}_3 & T_s\mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 \end{bmatrix}, \quad (9)$$

with $T_s$ representing the sampling time and $\mathbf{w}_k \in \mathbb{R}^6$ is the discrete-time state noise vector, which is assumed to be a zero-mean white Gaussian sequence with covariance $\mathbf{Q}_k$.

Assume that the initial state is a Gaussian vector with known mean $\bar{\mathbf{x}}$ and covariance $\bar{\mathbf{P}}$.

Now, define the filter measurement vector $\mathbf{y}$ to be the CM position as computed by equation (2), *i.e.*, $\mathbf{y} \triangleq \mathbf{r}_{\mathrm{G}}^{\mathrm{B/G}}$. Therefore, from the state definition, one can establish the following discrete-time measurement equation:

$$\mathbf{y}_{k+1} = \mathbf{C}\mathbf{x}_{k+1} + \mathbf{v}_{k+1}, \quad (10)$$

where $\mathbf{C} \in \mathbb{R}^{3\times6}$ is the measurement matrix given by

$$\mathbf{C} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times3} \end{bmatrix} \quad (11)$$

and $\mathbf{v}_{k+1}$ is the measurement noise, which is assumed to be a zero-mean white Gaussian sequence with known covariance $\mathbf{R}_{k+1}$.

Finally, using the state-space model given by (8) and (10), the discrete-time formulation of the Kalman filter algorithm (Bar-Shalom et al., 2004) can be immediately applied to optimally estimate the state vector $\mathbf{x}_k$ over time.

## 4 Results

The low-cost position and velocity estimation system proposed in this paper is now evaluated experimentally. For this end, two experiments are conducted: a static test and a flight test of feedback position control of an AR.Drone 2 quadrotor vehicle. The KF covariance matrices $\mathbf{P}_0$, $\mathbf{Q}_k$, and $\mathbf{R}_k$ are tuned with values $\mathbf{P}_0 = \mathrm{diag}\left\{1.0\times10^{-2}\mathbf{I}_3, 1.0\times10^{-2}\mathbf{I}_3\right\}$, $\mathbf{Q}_k = \mathrm{diag}\left\{1.0\times10^{-8}\mathbf{I}_3, 1.0\times10^{-7}\mathbf{I}_3\right\}$, and $\mathbf{R}_k = 1.0\times10^{-5}\mathbf{I}_3$. Using least-squares estimation, the camera focal length is calibrated in $fx = 570$ and $fy = 533$ . The Kinect is installed at $H = 3.3$ m above the floor. The sampling time is set to $T_s = 0.08$ s.

### 4.1 Static Evaluation

The first test to evaluate the estimation system present in Section 2–3 is a static one. The target, which is a circular red mark with a diameter of 10 cm, is placed with its center at a given position $\mathbf{p}^i$ with zero velocity. The position $\mathbf{p}^i$ is then measured by means of a measuring tape and estimated with the proposed system. Denote the position measure by $\check{\mathbf{p}}^i$ and the corresponding estimate by $\hat{\mathbf{p}}^i$.

Table 1: Statistical test with 10 samples.

| | $\mathbf{w}_1$ | $\mathbf{w}_2$ | $\mathbf{w}_3$ | $\mathbf{w}_4$ | $\mathbf{w}_5$ |
|---|---|---|---|---|---|
| $e_x$ | 0.004 | 0.001 | 0.048 | 0.002 | 0.016 |
| $e_y$ | 0.045 | 0.050 | 0.033 | 0.019 | 0.008 |
| $e_z$ | 0.034 | 0.012 | 0.058 | 0.056 | 0.012 |
| $\mu_x$ | 0.004 | 0.501 | -0.451 | -0.497 | 0.484 |
| $\mu_y$ | 0.045 | -0.800 | -0.783 | 0.730 | 0.758 |
| $\mu_z$ | 0.734 | 0.687 | 0.758 | 0.756 | 0.712 |
| $\sigma_x$ | 0.001 | 0.002 | 0.001 | 0.002 | 0.005 |
| $\sigma_y$ | 0.001 | 0.002 | 0.005 | 0.002 | 0.006 |
| $\sigma_z$ | 0.007 | 0.008 | 0.005 | 0.008 | 0.009 |

Define the following metric to evaluate the system performance regarding its capability to estimate position:

$$e_k^i = \|\check{\mathbf{p}}_k^i - \hat{\mathbf{p}}_k^i\|. \quad (12)$$

The above test is repeated to obtain 10 samples of $e_k^i$. The sample mean $\mu^i$ and standard deviation $\sigma^i$ are registered in Table 1. The test is realized at five different positions, corresponding to $i = 1, 2, ..., 5$.

where it is maintained static validation of the positions estimated by the proposed system were made during tests with metric tapes. For this evaluation, it was taken 10 measurements for five desired variables of position (waypoints) in a 3D indoor environment. The test waypoints are defined as: $\mathbf{w}_1 = [0\ 0\ 0.7]^{\mathrm{T}}$ m, $\mathbf{w}_2 = [0.5\ -0.75\ 0.7]^{\mathrm{T}}$ m, $\mathbf{w}_3 = [-0.5\ -0.75\ 0.7]^{\mathrm{T}}$ m, $\mathbf{w}_4 = [-0.5\ 0.75\ 0.7]^{\mathrm{T}}$ m, $\mathbf{w}_5 = [0.5\ 0.75\ 0.7]^{\mathrm{T}}$ m. The precision of this test are showed in Table 1 which includes summary statistics such as the mean $\mu$, standard deviation $\sigma$, and the error between the desired values and the mean position of each dimension provided by the Kalman Filter output.

As the reader can see in Table 1, estimated position errors stay below 2 cm in most samples, and is around 5 cm in the worst case, that is acceptable for our application.

### 4.2 Closed-Loop Validation

In this subsection, experimental results validate the proposed estimation system by using a position controller to guide the AR.Drone quadrotor in the environment. The waypoints chosen for the flight test are trajectories composed by a set of straight lines from the initial position $\mathbf{r}_i = [0\ 0\ 0.5]^{\mathrm{T}}$ passing by the waypoints $\mathbf{w}_1 = [0.4\ 0\ 0.5]^{\mathrm{T}}$, $\mathbf{w}_2 = [0.4\ -0.5\ 0.5]^{\mathrm{T}}$, $\mathbf{w}_3 = [-0.4\ -0.5\ 0.5]^{\mathrm{T}}$, $\mathbf{w}_4 = [-0.4\ 0.5\ 0.5]^{\mathrm{T}}$, $\mathbf{w}_5 = [0.4\ 0.5\ 0.5]^{\mathrm{T}}$. The position controller is composed by the altitude and horizontal controllers. The altitude control is a Proportional (P) controller and the horizontal control is a Proportional-Derivative (PD) controller. The constant velocity chosen in the trajectory generation was $v = 0.5$ m/s$^2$.

Analyzing the flight data obtained through the proposed system, Figure 4 shows the innovation $e_k$ during the flight test. As it can be seen in Figure 4 the variable $e_k$ tends to converge to zero, except for the waypoint switching transients.

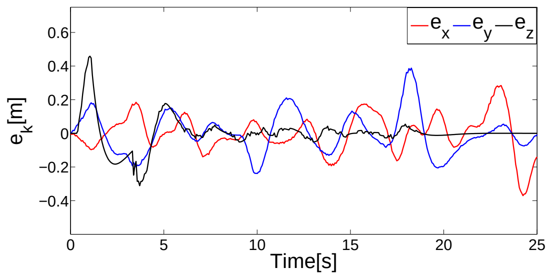


Figure 4: Innovation in a flight test.

For evaluation of the position estimation were generated separate graphics for each position state comparing with their desired values, see Figure 5. Note that the estimated positions are well close to the waypoints references, where the position error oscillate by $\pm 0.02m$ around their desired values, that is a good precision.

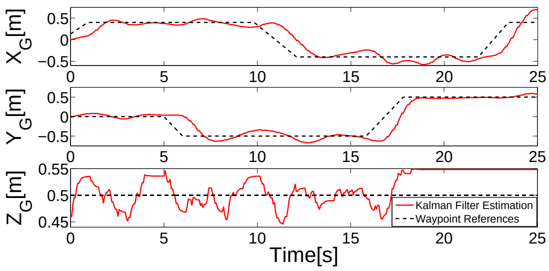


Figure 5: Position state estimation.

## 5 Conclusions

This paper demonstrated a low-cost alternative for the estimation of velocity and position of a MAV in indoor flight. To develop this solution it was used three main components: a low-cost Kinect sensor; a MAV; and a ground computer equipped with MATLAB. First, from the sensor data provided, the computer vision methods were used to identify and tracking a round red color mark attached over the AR.Drone 2 quadrotor. To find the round color mark and determine its centers the Hough transform (HT) was used. After the centroid of the color mark be well detected, to determine its position coordinates in $S_G$ from the image coordinates $S_I$, we used geometric transformation as detailed in section 3. In sequence, to estimate the velocities, we aided a Kalman filter as output of the system.

Finally, the proposed system was evaluated trough a static experiment and by using a closed-loop validation. In both, the experimental results showed a good performance of the estimation system, where the errors in the position estimation oscillated within a range allowed for our application, which is intended to validate guidance and navigation methods in the future.

## Acknowledgments

We would like to thank "Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq)" (Grant: 475251/2013), "Fundação de Amparo à Pesquisa do Estado de São Paulo" (Grant: 2015/04393-2) and "Coordenação de Aperfeiçoamento de Pessoal de Nível Superior" for supporting this project. The authors are also grateful to Elio Tecnologia.

## References

Alvissalim, M. S., Zaman, B., Hafizh, Z. A., Ma'sum, M. A., Jati, G., Jatmiko, W. and Mursanto, P. (2012). Swarm quadrotor robots for telecommunication network coverage area expansion in disaster area, *SICE Annual Conference (SICE), 2012 Proceedings of*, IEEE, pp. 2256–2261.

Bar-Shalom, Y., Li, X. R. and Kirubarajan, T. (2004). *Estimation with applications to tracking and navigation: theory algorithms and software*, John Wiley & Sons.

Bristeau, P.-J., Callou, F., Vissière, D. and Petit, N. (2011). The Navigation and Control technology inside the AR . Drone micro UAV, *Proceedings of the 18th IFAC World Congress, 2011* **18**: 1477–1484.

Bulgakov, A., Emelianov, S., Bock, T. and Sayfeddine, D. (2014). Control of hovering altitude of a quadrotor with shifted center of gravity for inspection of high-rise structures, *Proceedings of the 31st International Symposium on Automation and Robotics in Construction (ISARC), Sydney, Australia.*

Campbell, J., Hamilton, J., Iskandarani, M. and Givigi, S. (2012). A systems approach for the development of a quadrotor aircraft, *Systems Conference (SysCon), 2012 IEEE International*, pp. 1–7.

Chow, J., Ang, K., Lichti, D. and Teskey, W. (2012). Performance analysis of a low-cost triangulation-based 3d camera: Microsoft kinect system, *International Society for Photogrammetry and Remote Sensing Congress (ISPRS)*, Vol. 39, pp. 175–180.

Clark, R. A., Pua, Y.-H., Fortin, K., Ritchie, C., Webster, K. E., Denehy, L. and Bryant, A. L.

(2012). Validity of the microsoft kinect for assessment of postural control, *Gait & posture* **36**(3): 372–377.

Crowther, B., Lanzon, A., Maya-Gonzalez, M. and Langkamp, D. (2011). Kinematic analysis and control design for a nonplanar multirotor vehicle, *Journal of Guidance, Control, and Dynamics* **34**(4): 1157–1171.

dos Santos, D. A., Saotome, O. and Cela, A. (2013). Trajectory control of multirotor helicopters with thrust vector constraints, *Control & Automation (MED), 2013 21st Mediterranean Conference on*, IEEE, pp. 375–379.

Ducard, G. and D'Andrea, R. (2009). Autonomous quadrotor flight using a vision system and accommodating frames misalignment, *Industrial Embedded Systems, 2009. SIES '09. IEEE International Symposium on*, pp. 261–264.

El-laithy, R. A., Huang, J. and Yeh, M. (2012). Study on the use of microsoft kinect for robotics applications, *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, pp. 1280–1288.

Emelianov, S., Bulgakow, A. and Sayfeddine, D. (2014). Aerial laser inspection of buildings facades using quadrotor, *Procedia Engineering* **85**: 140–146.

Ganganath, N. and Leung, H. (2012). Mobile robot localization using odometry and kinect sensor, *Emerging Signal Processing Applications (ESPA), 2012 IEEE International Conference on*, pp. 91–94.

Gonzalez, R. C., Eddins, S. L. and Woods, R. E. (2009). *Digital image processing using MATLAB*, Gatesmark Publishing.

Grover, M. (2015). International journal of engineering sciences & research technology low cost flying bot used for aerial photography and face detection.

Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*, Cambridge university press.

Hsu, H.-m. J. (2011). The potential of kinect in education, *International Journal of Information and Education Technology* **1**(5): 365.

Luque-Vega, L. F., Castillo-Toledo, B., Loukianov, A. and Gonzalez-Jimenez, L. E. (2014). Power line inspection via an unmanned aerial system based on the quadrotor helicopter, *Mediterranean Electrotechnical Conference (MELECON), 2014 17th IEEE*, pp. 393–397.

Mahony, R., Kumar, V. and Corke, P. (2012). Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor, *IEEE Robotics Automation Magazine* **19**(3): 20–32.

Naidoo, Y., Stopforth, R. and Bright, G. (2011). Development of an uav for search & rescue applications, *AFRICON, 2011*, IEEE, pp. 1–6.

Nakamura, T. (2011). Real-time 3-d object tracking using kinect sensor, *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, pp. 784–788.

Prado, I. A. A. and dos Santos, D. A. (2014). A safe position tracking strategy for multirotor helicopters, *22nd Mediterranean Conference on Control and Automation*.

Siradjuddin, I., Behera, L., McGinnity, T. M. and Coleman, S. (2012). A position based visual tracking system for a 7 dof robot manipulator using a kinect camera, *Neural Networks (IJCNN), The 2012 International Joint Conference on*, IEEE, pp. 1–7.

Viana, Í. B., Prado, I. A. A., dos Santos, D. A. and Góes, L. C. S. (2015a). Formation flight control of multirotor helicopters with collision avoidance, *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, IEEE, pp. 757–764.

Viana, Í. B., Prado, I. A. A., dos Santos, D. A. and Góes, L. C. S. (2015b). Trajectory tracking control of an aerial robot with obstacle avoidance, *IFAC-PapersOnLine* **48**(19): 81–86.

Widodo, N. S. and Sunardi, A. Y. (2014). Low cost open source based uav for aerial photography.

Yan, J., dos Santos, D. A. and Bernstein, D. S. (2013). Adaptive control with convex and saturation constraints, *ASME 2013 Dynamic Systems and Control Conference*, American Society of Mechanical Engineers, pp. V001T02A002–V001T02A002.

Zhang, C. and Kovacs, J. M. (2012). The application of small unmanned aerial systems for precision agriculture: a review, *Precision agriculture* **13**(6): 693–712.

Zhou, D. and Schwager, M. (2014). Vector field following for quadrotors using differential flatness, *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, IEEE, pp. 6567–6572.