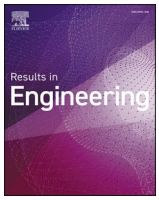
Results in Engineering 23 (2024) 102498

Contents lists available at ScienceDirect

Results in Engineering

journal homepage: www.sciencedirect.com/journal/results-in-engineering



Autopilot drone in construction: A proof of concept for handling lightweight instruments and materials

Zeiab Imani [a], Perry Forsythe [b], Alireza Ahmadian Fard Fini [b], Mojtaba Maghrebi [c,*], Travis S. Waller [d]

[a] Department of Computer Engineering, Ferdowsi University of Mashhad, Iran
[b] School of Built Environment, University of Technology Sydney, Australia
[c] Department of Civil Engineering, Ferdowsi University of Mashhad, Iran
[d] Chair of Transportation Modelling and Simulation, Technische Universität Dresden, Germany

ARTICLE INFO

Keywords:
Drone
UAV
Autopilot
GPS calibration

ABSTRACT

Applications of Unmanned Aerial Vehicles (UAVs) in construction have been increased in recent years mainly for purposes of automatically collecting field data. In the market, there is a growing opportunity for using UAVs in material handling. On the other side, vertical transportation in construction sites is a crucial challenge that leads to a drop in productivity efficiency. This paper presents a proof of concept for automating the lifting procedure of lightweight instruments and materials in construction sites. To do so, a platform is developed that could automatically manage material/instrument handling requests via an autopilot drone. The aircraft position is tracked with an onboard GPS module which is not accurate in urban areas. So inaccurate GPS signals could deviate the drone from the predefined path. To tackle this problem, a machine vision algorithm is utilized to calibrate the real-time position of the aircraft. A self-supervised learning algorithm is integrated into the system to handling collision avoidance during the flight. Finally, the developed methods are tested with an UAV in a real world environment to demonstrate the potentials of hiring autopilot drones in construction.

1. Introduction

As the population of urban areas grows, demand for high-rise building has been increased [1] which lead to a great request for vertical transportation of materials, equipment, and workers in high-rise construction sites.

The most important challenge with vertical transportation is time. Lack of proper time management in vertical transportation can cause bad effects on the efficiency of the manufacturing process. These effects could be summarized as follows.

- Vertical transportation of workers might be a bottleneck in construction sites in the peak time (at the beginning and the end of a working day) [2]. A recent study shows that around 20 % of workers' time is wasted waiting for vertical transportation.
- If the materials and equipment can't be delivered at the right time, it causes a substantial delay in the construction progress [3].

Tower cranes, lifts, and hoists are used for vertical transportation in construction sites (we call them *Vertical Transportation Equipment* in this paper). To address the ever-growing demand for vertical transportation, increasing the number of *Vertical Transportation Equipment* might be a quick solution in a high-rise construction project. However, due to cost and space limitations, increasing the number of *Vertical Transportation Equipment* in a site is technically impossible [4]. To tackle this problem, a number of studies have been tried to optimize the *Vertical Transportation Equipment* operation in order to increase their transportation capacities [3–6].

Maybe a drone-based method could be considered as a practical solution for vertical transportation. The recent development in technology brought a big market for UAVs in different industries such as military [7,8], commercial freight [9,10], agriculture [11,12], disaster management [13–15], wildlife monitoring [16], mining [17], cinema [18], and construction [19].

Particularly in construction, due to the convenient features like the ability to reach hard-to-access areas and the cost-effective price of the UAVs, they are utilized in the automation of different processes.

* Corresponding author.
E-mail address: mojtabamaghrebi@um.ac.ir (M. Maghrebi).

https://doi.org/10.1016/j.rineng.2024.102498
Received 22 December 2023; Received in revised form 27 June 2024; Accepted 1 July 2024
Available online 2 July 2024

- Data Gathering [20–23].
- Progress Monitoring [24,25].
- Safety Control [26,27].
- Mapping and Surveying [28–30].
- Real-scale Structure Building [31].

To the best of our knowledge, no research is attempted to automate the vertical transportation in construction sites relying on the current potentials of the UAVs. UAVs can be utilized as a low-cost, safe, and autonomous solution for purpose of vertical transportation.

In this paper, a proof of concept for lifting lightweight instruments and materials on construction sites is presented which examines the current technical challenges of using a drone on a construction site as transportation equipment. To do so a platform is developed that could automatically manage material/instrument handling requests via an autopilot drone. The platform consists of a mobile application that is designed for defining a mission and automatically controlling the UAV through the transportation operation. The aircraft position is tracked with an onboard GPS module which is not accurate in urban areas [32]. To prevent the aircraft from deviating from its path, due to inaccurate GPS signals, a machine vision algorithm is developed to calibrate the aircraft's position. Also, a self-supervised algorithm named Monodepth2 [33] is integrated into the system for collision avoidance. Some attempts are done to increase the accuracy of Monodepth2.

In summary, this paper tries to emphasize the potential applications of using drone-based solutions for vertical transportation, as well as, includes a practical demonstration for the proposed idea.

This paper offers more than a proof of concept, where the proposed idea is tested and implemented in a case study. This include, developing User Interface (UI), User Experience (UX), real-time calculation modules and image processing techniques which are seamlessly interconnected via a cross-platform mobile application. The developed modules are Mission definition, Threshold, Depth Estimation, Calibration, and Collision Avoidance that are encapsulated in the mobile app to reflect the practicality of the proposed method.

The rest of the paper is organized as follows: Section 2 is a summary of the proposed methods in autonomous navigation and auto-landing of the UAVs. Section 3 describes the implementation of the system that is consists of the mobile application and algorithms. Section 4 presents the results of the real-world experiments and sections 5 and 6 are the discussion and conclusion.

2. Related works

Autonomous navigation and auto-landing are two important subjects in the drone controlling system that have received considerable attention in the last two decades. Autonomous navigation of the UAVs means flying in an environment based on a pre-defined path or some moving strategies while avoiding the unpredicted obstacles getting in the way of the UAV. Auto-landing is the operation of correct landing on an appropriate area that is doing in known or sometimes unknown environments. The next two sub-sections describe a summary of the most recent studies that have been conducted in *i)* Autonomous navigation and, *ii)* Auto-landing.

2.1. Autonomous navigation

Autonomous navigation of UAVs is an important subject in the safe utilization of these aerial robots which are used in a great number of applications in recent years. Some approaches are based on the depth information of an environment that is provided by sensors (such as LiDAR [34]) or cameras (such as RGB_D camera [35–37]). Devos et al. [34] designed a system that could help the UAVs equipped with two light-weight LiDAR sensors to avoid the corners and deadlocks. Usenko et al. [36] proposed a system that tries to move in a predefined trajectory with the least deviation while avoiding unpredicted obstacles. They used an occupancy model of the environment which is made with the help of an RGB_D camera. Similarly, Brunner et al. [35] used depth images to make the occupancy map to prevent collisions in a delivery drone in an urban area. Wang et al. [37] combined RGB_D camera with a deep-learning-based object detection algorithm to design a system that could retrieve related information about the obstacles' classes, the 3D spatial position of obstacles and tries to optimally avoid them.

Nowadays most UAVs are equipped with a light-weight monocular camera and using extra depth sensors can increase their weight, energy consumption, and cost. So, many researchers have attempted to provide safe navigation for UAVs that are equipped with a monocular camera. The algorithms used in the area of autonomous navigation for UAVs can be grouped as follows.

- classical image processing [38,39].
- deep-learning-based approaches [40,41].
- reinforcement learning [42,43].

Among the classical image processing algorithms that have been hired to detect obstacles can refer to Saha et al. [39] which used the Speeded Up Robust Features (SURF) algorithm to design a mathematical model for detecting the obstacles and estimating the relative distance to them. Likely, Al-Kaff et al. [38] investigated the changes in the convex hull around the extracted features from the detected obstacles in order to avoid collisions.

Always classical image processing algorithms do not lead to promising results because of different shapes and positions of obstacles. So, to increase the accuracy, deep-learning-based approaches are proposed. Padhy et al. [41] used a Convolutional Neural Network (CNN) model to navigate in a corridor without any collision to the walls. Kouris et al. [40] used a regression CNN trained by images of an environment labeled with the distances to predict the distance to obstacles.

Deep-learning-based navigation methods are accurate but need a large amount of labeled data for training the system. To overcome this limitation, the drone can learn by itself from the surrounding environment using Reinforcement Learning (RL). Given that in RL methods, the drone itself learns the navigation, though the safety of the drone has to be considered during the learning time. Anwar et al. [42] proposed a system based on an end-to-end reinforcement learning algorithm that could navigate in an indoor environment. They generated a reward function based on the information of one monocular camera. Also, they designed a virtual crash reward for safety purposes during the learning time. Shin et al. [43] used the combination of a supervised segmentation network and an Actor-Critic network running on an RL platform to navigate the drone in a simulated environment. Their achieved results in both the simulation environment and the matching real-world experiments are admissible, but the errors increased when the locations of the obstacles were changed.

Rather than the methods mentioned above, there is another approach that mainly uses a depth map of the surrounding environment for detecting/avoiding obstacles. Monocular RGB cameras have no depth information, Laina et al. [44] proposed a single-scale CNN architecture that follows residual learning to get the depth map of a single image. Their model outperforms the previous approaches [45–48] in quality of the results, training time, and also, the amount of data needed for training. Godard et al. [33] proposed a self-supervised monocular depth estimation model that despite its simplicity, achieved acceptable results. Singla et al. [49] proposed a deep RL method for UAVs' safe navigation in unknown environments. The idea was to use the relevant information about the environment to have a better performance in

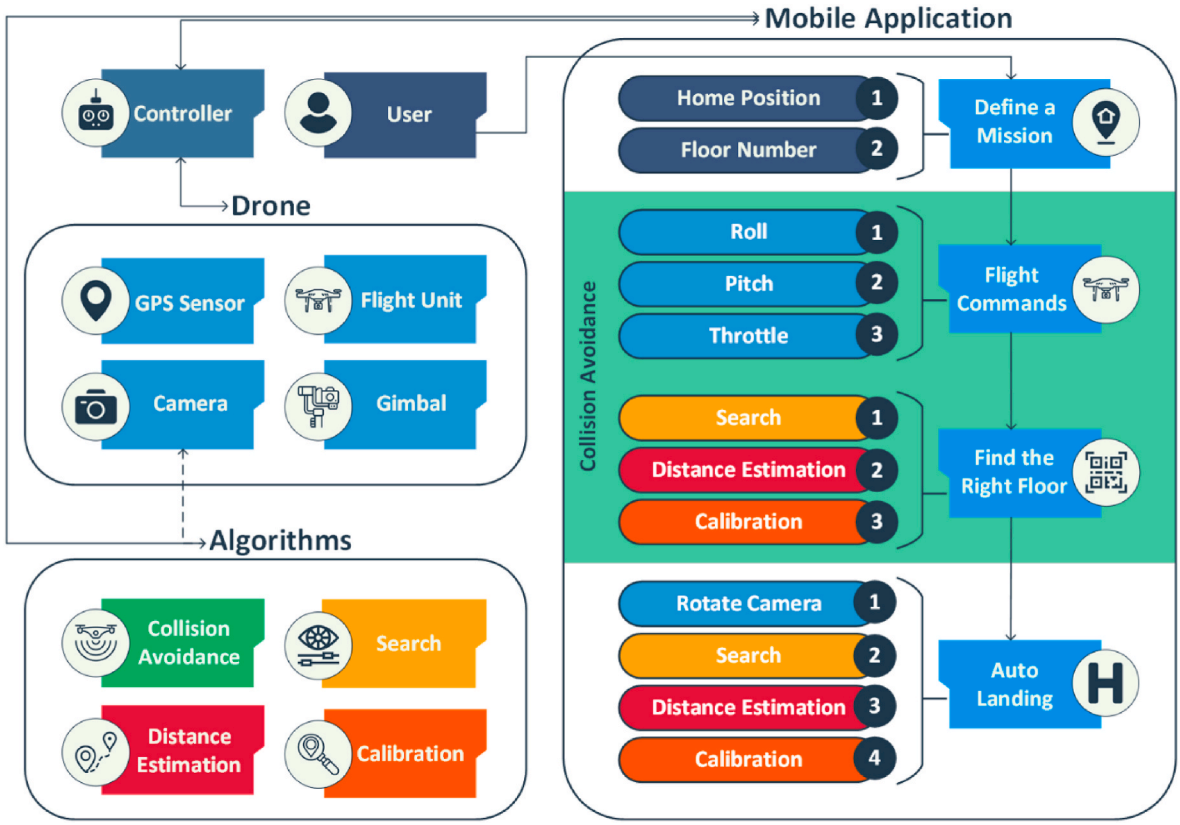


Fig. 1. The system overview.

obstacle avoidance. A Recurrent Neural Network (RNN) with temporal attention was also used to collect and store the information gathered over time. To have more information from the environment they used a conditional Generative Adversarial Network (cGAN) to make the depth map of a monocular view. Producing the depth map could lead to an enhancement in the accuracy of collision detection and avoidance because it provides more information about the surrounding environment.

2.2. Auto-landing

Due to the widespread use of UAVs in various industries, and the important role of auto-landing in UAV applications, this issue has been considered in a great number of studies. The auto-landing methods can be categorized into *i)* marker-based and, *ii)* marker-less approaches. Marker-based approaches use a marker as the landing station. The markers can be grouped based on the shapes which are used in them; i-a) self-designed markers [50–54], and i-b) fiducial markers [55–59]. The self-designed markers usually contain geometric shapes (such as circles or ellipses) and letters (such as 'H', 'T', or 'X'). Garcia-Pulido et al. [52] developed an image processing-based algorithm to detect a landing marker that consists of four circles and three ellipses. This approach could perform the landing procedure even if part of the landing marker is not completely distinguishable in the camera view, but has difficulties in the environments that contain elliptical shapes. Nguyen et al. [53] and Truong et al. [54] utilized deep learning-based methods to detect a marker that consists of three nested circles each is divided into 8 parts. Nguyen et al. [53] proposed a method that is capable of detecting the landing marker in different lighting conditions and works with less computation time. The errors increase when the landing marker is moving at a high speed in the image view. Truong et al. [54] used an image super-resolution (SR) reconstruction method that works well with low-resolution images captured by a cheap camera. In other studies, an 'X' letter surrounded by a circle is used as a landing marker for landing on a moving platform [50,51].

Fiducial markers are widely used as guidance in landing and localization. These markers are fast to detect and robust against noises [60]. Sani et al. [58] used an ArUco[1] marker as the landing marker and used the Inertial Measurement Unit (IMU) data to reach the marker when it is not in the field view to provide auto-landing. Wubben et al. [59] proposed a method that could detect an ArUco marker from a high distance up to 30 m and perform an accurate landing. An AprilTag[2] is used in Refs. [55–57] as a landing marker on a moving vehicle.

Marker-less approaches are not so accurate but can be used for landing in unknown environments and during emergency events. These methods could find an appropriate landing area in an environment without any pre-knowledge. Förster et al. [61] proposed a method that used a bottom-facing monocular camera to properly find a landing position beneath the drone. Chung et al. [62] developed an optic flow contour module that could explore the below surface to see if it is flat and appropriate for landing or it includes obstacles. They also used an object detector that could detect the objects pass by beneath the drone. Yang et al. [63] used the Simultaneous Localization and Mapping (SLAM) based method in combination with a grid map to find the appropriate landing zone for the drone in emergencies.

State-of-the-art autopilot drones consist of algorithms that provide autonomous navigation and auto-landing, as has been mentioned above. This paper tries to propose a platform that could handle an autopilot drone in construction. The platform consists of a computer vision algorithm along with a monocular depth map generator for autonomous

[1] A square shape marker with a black border that contains an identifier as a binary matrix.

[2] A square shape marker with a black border that contains an encoded data (4–12 bits) as a binary matrix.

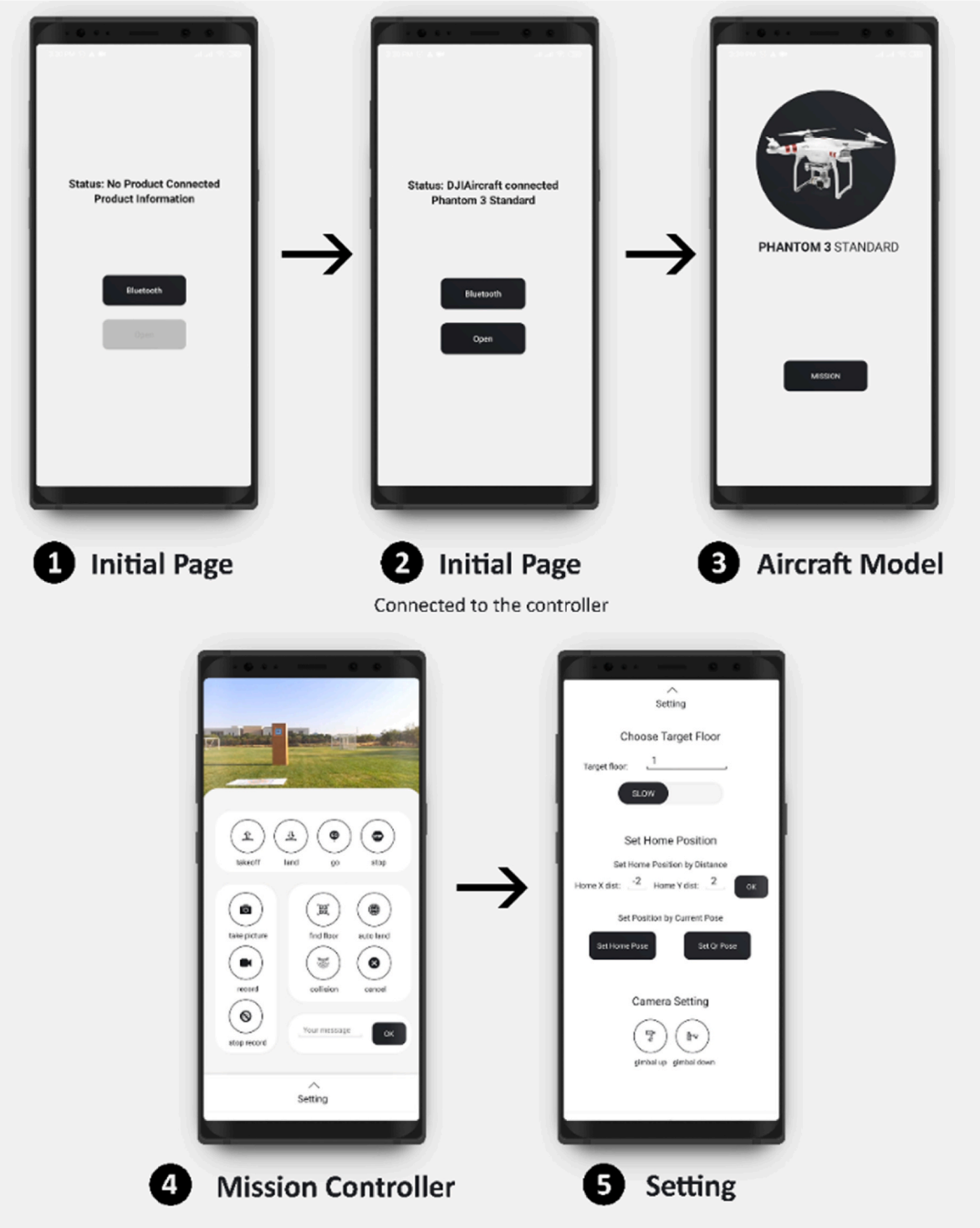


Fig. 2. The mobile application chart.

and safe navigation and also, a self-designed marker-based auto-landing approach.

3. Methodology

This paper proposing a proof of concept for using an autopilot drone to lift material/instrument in construction sites. Due to the large scope of autopilot drones, this work is focused on automatic maneuvering by calculating flight commands, obstacle avoidance, and auto-landing. To do so, a platform is developed which is depicted in Fig. 1. The platform consists of a mobile application that is in charge of controlling the drone through a flight. It provides a chance for a user to define a material/instrument supply mission.

The user should define the target floor number and the home position in the mobile application. Then, a path to the destination is calculated in the application and the aircraft will automatically follow the defined pass up to the desired floor. The aircraft position is tracked with an onboard GPS module during the flight. Due to inaccurate GPS signals in the urban areas [32], the aircraft should update its altitude with a QR code that is attached to the surrounding environment. The floor number

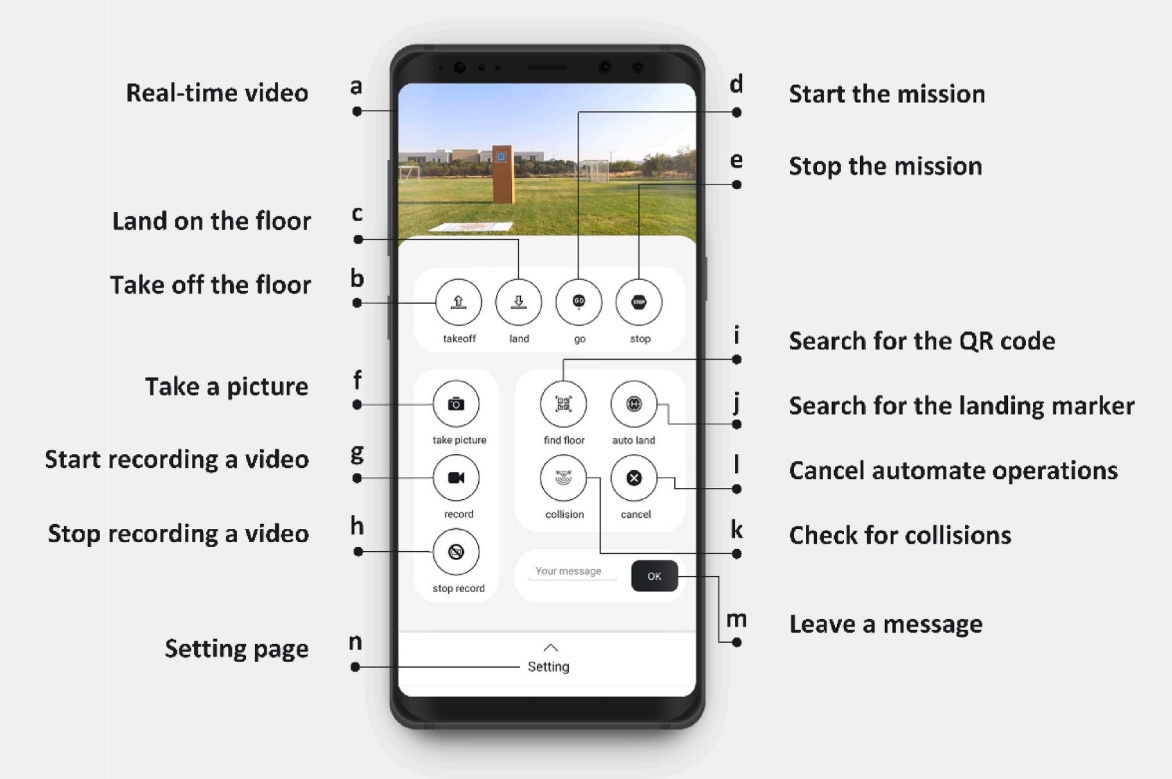


Fig. 3. Mission controller page of the mobile application.

is encoded in the QR code. After finding the target floor by decoding the QR code information, the aircraft will rotate its camera to view downward and search for the landing marker which assists the drone to update its real-time position. Finally, the 3D position of the aircraft (include latitude, longitude, and altitude) will be updated based on the information acquired from its environment. So, the drone does not only rely on the information of GPS signal for maneuvering and utilizes the camera to increase accuracy. The valid spatial information provides a chance for the drone to land in the desired position.

Rather than correcting unreliable GPS signals during a flight, this paper has covered another practical challenge which is automatically detected and avoided obstacles during a flight. The developed platform is able to handle the events when there is a high risk of collision between the drone and an object. During the flight, if a high-risk event is detected, the platform automatically cancels the mission and the drone will be landed in a near safe area.

In the following sections, the mobile application module and utilized algorithms are described in detail.

3.1. Mobile application

The mobile application is in charge of mission definition for transferring the material/instrument, and automatically performing the mission utilizing an aircraft. As it has been shown in Fig. 1, the mobile application is designed to getting the home position and the target floor number to define a new mission as well as providing a machine vision algorithm for autonomous navigation and auto-landing. A review of the mobile application is presented in Fig. 2. The application consists of four different pages. Fig. 2 (1) shows the initial page of the application. The app does not open until it connects to the drone's remote controller via a Wi-Fi connection (Fig. 2 (2)). After connecting to the remote controller, the model of the aircraft can be shown on the next page (Fig. 2 (3)). The user can go to the main part of the application by pressing the *Mission* button. Fig. 2 (4) is the mission controller page that assists the user to have complete control over the procedure of the platform. The setting page is presented in Fig. 2 (5). The two important pages of the application (mission controller and setting) are described completely in the following paragraphs.

Fig. 3; shows the mission controller page of the application. On this page, *(a)* a real-time video is shown at the top of the page, and the user assistant functions are grouped on the rest of the page. The first group of functions below the camera view is for the drone's flight control functions consists of *(b)* takeoff, *(c)* land, and *(d,e)* mission start/stop. The second group of functions on the left side of the page is for working with the drone's camera, which consists of *(f)* taking pictures and *(g,h)* start/stop recording the video. The last group of functions on the right side of the page is dedicated to controlling the drone vision-based procedures consists of *(i)* searching for the QR code, *(j)* automatic landing on the landing marker, *(k)* checking for collision during flight, and *(l)* canceling the automatic operations. The user can leave a note in the dedicated message box *(m)*. The setting page will be open by pressing the setting bar *(n)*.

Fig. 4 shows the settings which are done once before starting the mission. On this page, the user can *(a)* specify the target floor number and *(b)* the speed of the aircraft during flight, *(c)* set the home position and, *(d)* change the gimbal settings.

The next section describes the algorithms that were developed to handle autonomous navigation and auto-landing tasks.

3.2. Methods and algorithms

In this study, two kinds of markers are used for calibrating the 3D position (include latitude, longitude, and altitude) of the aircraft during flight. After detecting each marker, a machine vision algorithm is used for estimating the aircraft distance to the marker. This distance is used during the calibration procedure.

As it has been shown in Fig. 1 the vision-based algorithms are divided into four main parts, which consist of:

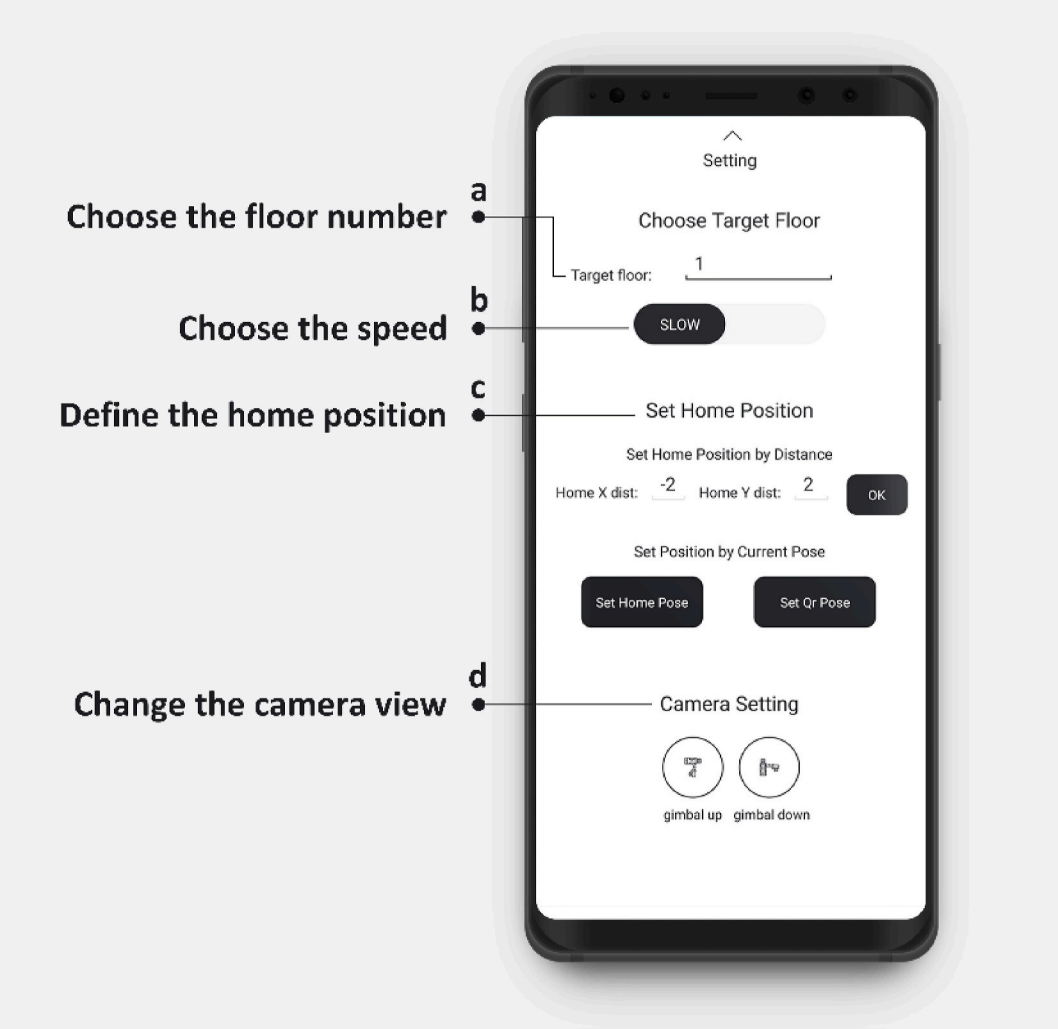


Fig. 4. Setting page of the mobile application.

- Threshold (Search)
- Depth Estimation
- Calibration
- Collision avoidance

A *Threshold* algorithm is implemented to detect the color markers in different lighting conditions and during the movement of the aircraft. *Depth Estimation* is performed for *Calibration* purposes. *Calibration* is the procedure of utilizing the *Threshold* and *Depth Estimation* algorithms in order to compute the flight commands for autonomous navigation and auto-landing. Also, a self-supervised algorithm named Monodepth2 [33] is utilized to ensure drone safety through autonomous navigation.

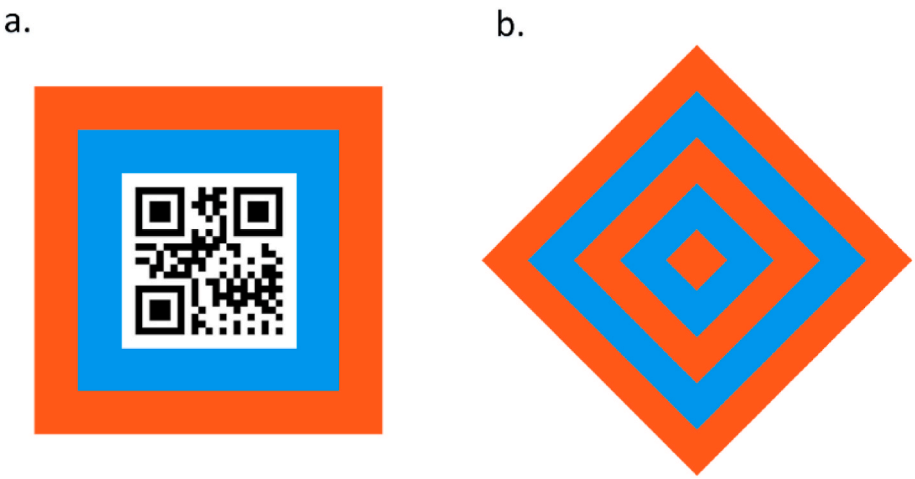


Fig. 5. Calibration markers. a) The QR code, b) The landing marker.

Monodepth2 supplies the depth map from the mounted RGB camera on the drone for checking a high risk of collision in front of the aircraft. Also, brightness adjustment is used to increase the accuracy of the depth estimation approach. In other words, the proposed method can detect any obstacle (stationary or mobile) in different lighting conditions while aircraft is flying. When an obstacle is detected, first the aircraft is stopped at the current aerial position and then due to safety issues the mission is cancelled and aircraft safely is returned, under autopilot control, to the home point where the flight was started.

3.2.1. Markers

Two different kinds of markers are used for calibrating the position of the aircraft. One is used as the navigation guides and the other is used as the landing marker.

Quick Response (QR) code is a famous 2D marker that can store information and also is simple and fast to detect. In this study, each floor number in the target building is encoded via QR code and labeled to the corresponding floor in the construction site. The user specifies the floor number to which he wants to send the material/instrument in the application. QR codes help the drone to distinguish the right floor and correct the mission if GPS signals are not accurate.

As it has been shown in Fig. 5 (a), a specific color pattern square is used around the marker to facilitate the *Threshold* algorithm. The QR code is a 15 × 15 cm square with two colored squares with a size of 26 × 26 cm and 35 × 35 cm around.

In this study, a self-design marker is used as the landing marker to determine the exact location of the landing area for the aircraft so that it can calibrate itself with it. This marker consists of five nested squares (Fig. 5 (b)). The inner squares are captured in shorter distances while the bigger squares are not complete in the image view. Also, the bigger squares can be detected from far distances when the inner ones are not so clear and distinguishable. The marker is designed in a specific color pattern to facilitate the *Threshold* algorithm. The size of the smallest and the biggest squares are 7 × 7 cm and 35 × 35 cm, respectively.

The next section describes the *Threshold* algorithm that is used to detect markers in different situations covering blur images caused by drone motion and color change in different outdoor lighting conditions.

3.2.2. Threshold

Due to the different lighting conditions in the outdoor environment and also the image motion blur caused by the movement of the aircraft, a color pattern is used to facilitate the searching procedure for the markers. The color pattern is used rather than grayscale because it can work better in different lighting conditions. The *Threshold* is done in HSV Color Space rather than RGB. Unlike RGB, color and intensity information are separate in HSV so the *Threshold* will have better performance [64].

The *Threshold* algorithm which is shown in Algorithm 1 finds the border between two specific colors. The chosen colors are blue and orange, which have high contrast with each other. This method could provide fast marker detection in different lighting conditions and also is robust against noises. The similar colors in the environment cannot make noise in the threshold approach unless the blue and orange colors are exactly next to each other.

In the next section of the paper, the *Depth Estimation* unit is explained that is designed to find the distance to the wall and also the altitude of the aircraft during the auto-landing.

Algorithm 1 Thresholding

Input Raw camera frame

1: **procedure** THRESHOLD
2: Transfer frame to the HSV color space
3: Find the region of each color performing a simple threshold
4: Perform ***Morphology Dilation*** on both regions

$$a)\ \mathrm{K} \leftarrow \underset{9\times 9}{\begin{bmatrix} 1 & 1 & 1 \\ 1 & \ddots & 1 \\ 1 & 1 & 1 \end{bmatrix}}$$

$$b)\ \mathrm{A}_k \leftarrow translation\ of\ A\ by\ k$$

$$c)\ \mathrm{A} \oplus K \leftarrow \bigcup_{k\in K} A_k$$

5: Doing ***Bitwise AND*** operation on both regions to find the border
6: **end procedure**

3.2.3. Depth estimation

As it has been shown in Fig. 1 after traversing the user-defined path up to the target floor, the aircraft needs to calibrate its coordinates with the corresponding markers. This process could assist the drone to land on the target delivery station properly. To do so a machine vision algorithm is implemented that computes the distance to the markers.

The distance from the aircraft to pre-defined markers is estimated by solving the *Perspective-n-Point* problem [65]. This algorithm tries to find the relative 3D position of the camera in the world-coordinate with the help of *i)* three or more points of a known object in the field and, ii*)* the corresponding 2D coordinates acquired from the captured image. To find the 3D position of the camera in the world coordinate the camera extrinsic matrix $[R|t]$ (consists of rotation R and translation t matrix) needs to be computed using Eq. (1).

$$w \times p = K \times [\mathrm{R}|\mathrm{t}] \times \mathrm{P} \tag{1}$$

In the above equation, K is the camera intrinsic matrix, which is consist of the camera focal length (f), the skew parameter which is zero if the image axes are perpendicular (s), number of pixels in the x and y-direction (m_x and m_y), and the principal point of the camera ($\left(p_x, p_y\right)$) (Eq.

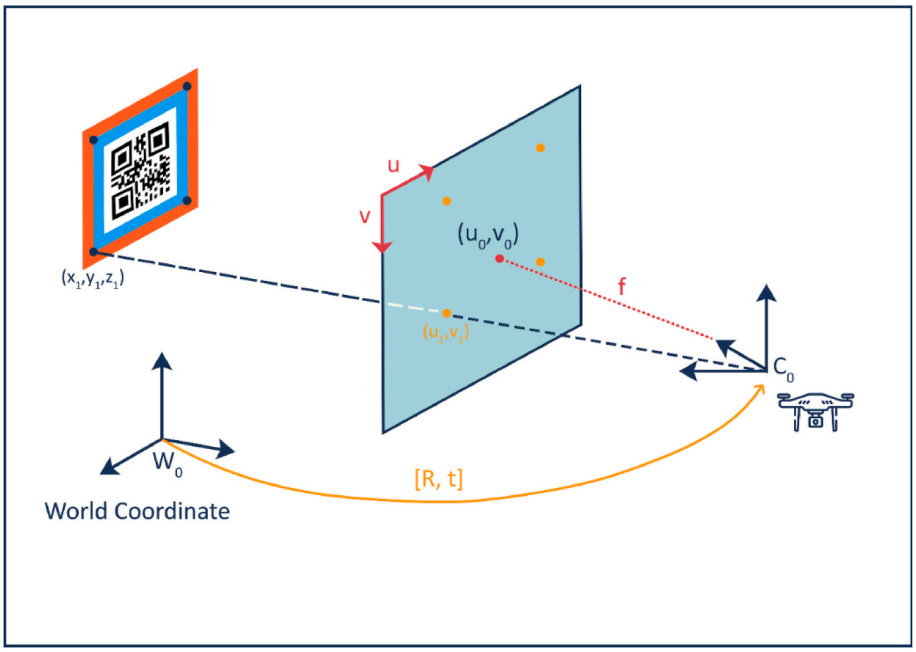


Fig. 6. Perspective-*n*-Point problem.

(2)). The f_x and f_y are the camera focal length in pixel and (u_0,v_0) is the principal point of the camera in pixel (Eq. (3)).

$$\mathrm{K}=\begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} f & s/m_x & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$\mathrm{K}=\begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$[R|t]$ matrix is the [3 × 4] camera extrinsic matrix where R is the [3 × 3] camera rotation matrix (Eq. (5)) that is the product of the rotation matrix on the X, Y, and Z-axis (Eq. (4)) and t is the [3 × 1] camera translation matrix.

$$\textit{Y-axis Z-axis X-axis}\ \mathrm{R}=\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix}\begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$\mathrm{R}=\begin{bmatrix} \cos\phi\cos\gamma & -\cos\phi\sin\gamma & \sin\phi \\ \cos\theta\sin\gamma+\cos\gamma\sin\theta\sin\phi & \cos\theta\cos\gamma-\sin\theta\sin\phi\sin\gamma & -\cos\phi\sin\theta \\ \sin\theta\sin\gamma-\cos\theta\cos\gamma\sin\phi & \cos\gamma\sin\theta+\cos\theta\sin\phi\sin\gamma & \cos\theta\cos\phi \end{bmatrix} \tag{5}$$

P is the 3D points of the specific object in the world coordinate that is represented as $[x \quad y \quad z \quad 1]^T$. Also, p is their 2D corresponding projection in the image that is represented as $[u \quad v \quad 1]^T$. Also, w is a scale factor for image points.

After calculating the camera extrinsic matrix, the camera center in the world-coordinate can be calculated using Eq. (6):

$$\mathrm{C}=-R^{-1}\times \mathrm{t} \tag{6}$$

The rotation performed on the camera can be undone using the inverse of the rotation matrix. The distance from the aircraft to the marker is also determined by the third element of the center vector C. A visual view of the algorithm is shown in Fig. 6.

The *Depth Estimation* algorithm is utilized to get the distance to the QR code and the landing marker during the autonomous navigation and auto-landing. This module is used in the *Calibration* algorithm which is explained in the next section.

3.2.4. Calibration

The aircraft position is calibrated with two kinds of markers during the flight for correcting the errors that might be happened because of the inaccurate GPS signals. This *Calibration* process is performed after the aircraft reached the right floor and during landing operation (Fig. 1).

Each floor has a labeled QR code that stores the floor number so the aircraft can find the right floor by searching and scanning the markers. When the aircraft gets closer to the targeted floor, the algorithm starts the searching procedure. To do so the aircraft moves on a square-shape path parallel to the wall and runs the *Threshold* algorithm to find the QR code. The aircraft increases its distance to the wall gradually to see more space and this process continues till the algorithm finds the corresponding marker.

After detecting one QR code, the information will be compared with the target floor number defined in the flight mission. If the QR code matches with the drone task, the aircraft calibrates its coordinate with the marker and starts the landing process, otherwise, the aircraft will continue the searching procedure.

After finding the right marker, the aircraft adjusts its coordinates with the QR code utilizing the *Depth Estimation* algorithm.

When the distance to the QR code is computed, the aircraft moves close to the marker with a certain distance (e.g. in this study we set the

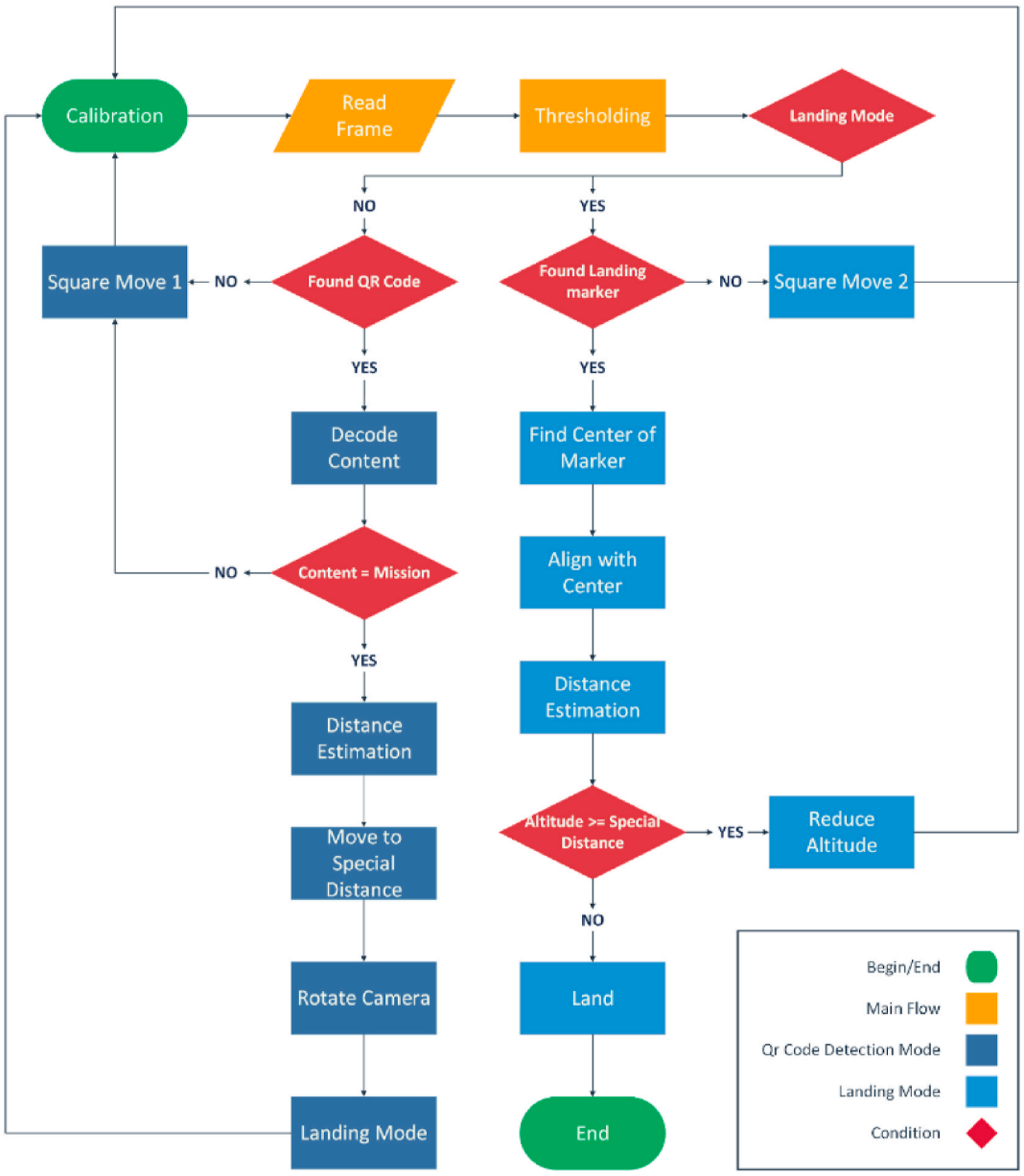


Fig. 7. Calibration flowchart.

distance 3 m) and then changes its camera view to downward. The landing marker is located on a platform along with the QR code. Similar to the previous step, the aircraft moves on a square-shape path parallel to the ground while running the *Threshold* algorithm and gradually increase its altitude to see more area. This process continues till the aircraft detects the landing marker.

After detecting the landing marker, the aircraft coordinates its position with the center of the landing marker. Due to the inaccuracy of GPS signals, the *Depth Estimation* algorithm is utilized for estimating the altitude of the aircraft during the auto-landing operation. Then the drone reduces its altitude until reaching a certain distance (e.g. in this study we set the distance 50 cm) above the landing marker. Now is the time that the application sends the landing command to the drone and the aircraft starts the landing procedure. After receiving the landing command, the aircraft reduces its altitude and turns off the engines after making sure the landing is doing correctly. The flowchart of the *Calibration* procedure is presented in Fig. 7.

In the next section, the safety unit of the proposed platform is described which is utilized to prevent frontal collisions during the flight.

3.2.5. *Collision avoidance*

Determining distances to the surrounding objects is necessitated during an autonomous flight, in order to avoid collisions. As it was previously mentioned, there are two main approaches for obstacle avoidance.

1 The approaches using depth sensors (such as LiDAR and RGB_D camera).
2 The approaches using a monocular camera.

This study tries to provide a low-cost and accurate solution. So, instead of using drones with high costs and features, we focused on developing the software of the platform. To do so we target micro UAVs which are easily available out-of-shelf. If the developed framework works on this type of aircraft, possibly could be easily adapted to the higher range of UAVs including Mini or Close range.

Inexpensive micro UAVs in the market, normally do not come with built-in depth sensors. So, to avoid collisions during the mission a self-supervised monocular Depth Estimation algorithm named *Monodepth2*

Fig. 8. The Phantom3 Standard and the simplified version of a building wall.

[33] is used. *Monodpth2* could handle the problem of occluded pixels in monocular view and ignore static pixels and the pixels that contradict the movement of the camera. Accordingly, *Monodpth2* outperforms the previous approaches [66–70]. So, we integrated the *Monodepth2* with our system to provide an accurate depth map of the monocular view. The algorithm can be trained with monocular images, stereo images, or both. The pre-trained model that was trained with 640 × 192 resolution monocular images is used in this platform. If Monodepth2 detects an object within a close distance (e.g. in this study we set the distance 74 cm or less) in front of the UAV, the platform recognizes it as a high-risk event. Then, the mission will be canceled and the aircraft will land immediately.

In order to get the best accuracy from monodepth2, some pre-processing techniques were examined. These techniques consist of median noise removal, local and global histogram equalization, brightness adjusting using simple transformation, and gamma filter. The accuracy increases in the case of simple brightness transformation but the other methods have negative impact on the results. Brightness adjusting is used in Ref. [71] to align the brightness of images in the training phase in which led to an improve in the accuracy. In our case it is applied just on the test images. This simple solution indicates that aligning the brightness between test and train images can lead to an improvement in the Monodepth2 performance.

4. Experimental results

This paper tries to prove that auto-pilot UAVs can be used as a solution for vertical transportation in construction. In this section, the proposed ideas are evaluated in the field. The experiments that are done for the system evaluation are described in the following paragraphs.

4.1. High-level equipment

Due to legal, safety, and privacy concerns, we had not found a chance to test the platform in a real construction project. But to demonstrate how effectively the platform could handle an autonomous material transportation mission, a very simplified version of a building wall is designed as the experimental environment.

The experiment is done in Ferdowsi University of Mashhad open courtyard. A DJI Phantom 3 Standard [72] is used to implement the developed method. The aircraft's 3-axis gimbaled camera provides the front and downward view with the 1280 × 720 (720p) resolution videos. The video stream and the computed flight commands are transferred between the drone and a mobile device through a self-designed application that is developed in this study. All the computations are performed in real-time using a mobile phone that has 8 GB RAM equipped with an Octa-core (2 × 2.05 GHz Cortex-A76 & 6 × 2.0 GHz Cortex-A55) CPU powered with Android 9.0 (Pie) OS.

The aircraft in the pre-designed environment is shown in Fig. 8.

4.2. System implementation

To implement the introduced platform, first, the mobile application is built based on DJI Mobile SDK [73] and it is accessible to download from [https://github.com/smartconstructiongroup/AutoPilot_Drone_in_Construction/]. DJI mobile SDK is a platform that enables Android or iOS devices to connect to the remote controller through either USB or Wi-Fi. The application supplies an augmented flight experience reflecting the live camera feed and real-time information of the subject aircraft. Also, the application performs all the algorithms for autonomous navigation, auto-landing, and collision avoidance. The image processing algorithms which are used include: *Mathematical Morphology*, *Connected Component*, and *Image Moment* are implemented in Java using the OpenCV library [74]. Monodepth2 [33] is implemented in Python and is connected to the developed application using Chaquopy 8.0 Android plugin [75].

4.3. Testing scenarios

The autopilot platform consists of different parts and algorithms.

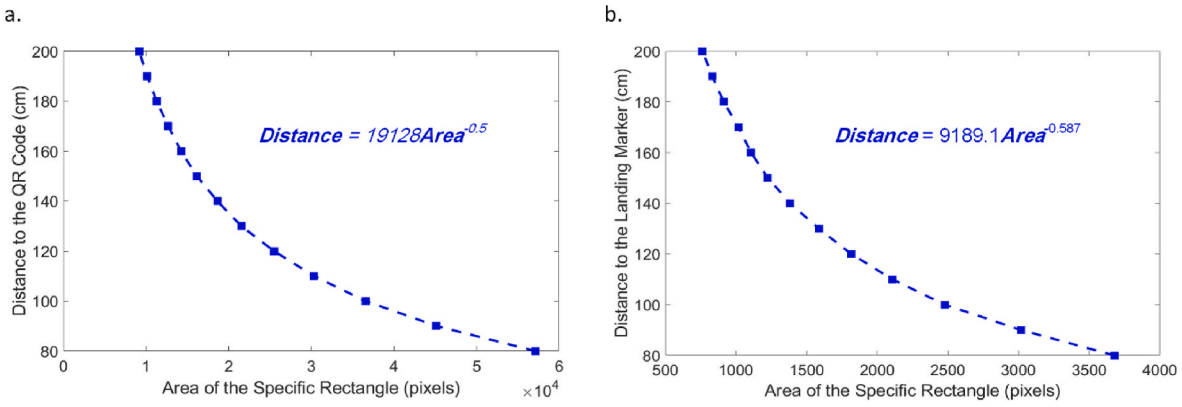


Fig. 9. The Area/Distance Equation. a) The QR code equation, b) The landing marker equation.

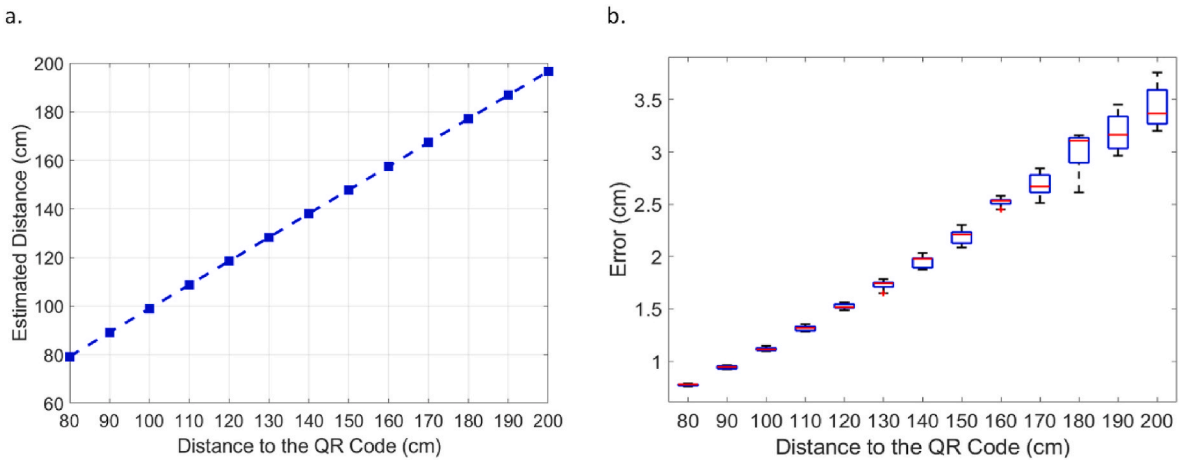


Fig. 10. Heuristic Depth Estimation algorithm. a) The mean estimated distance to the QR code b) The errors in each distance.

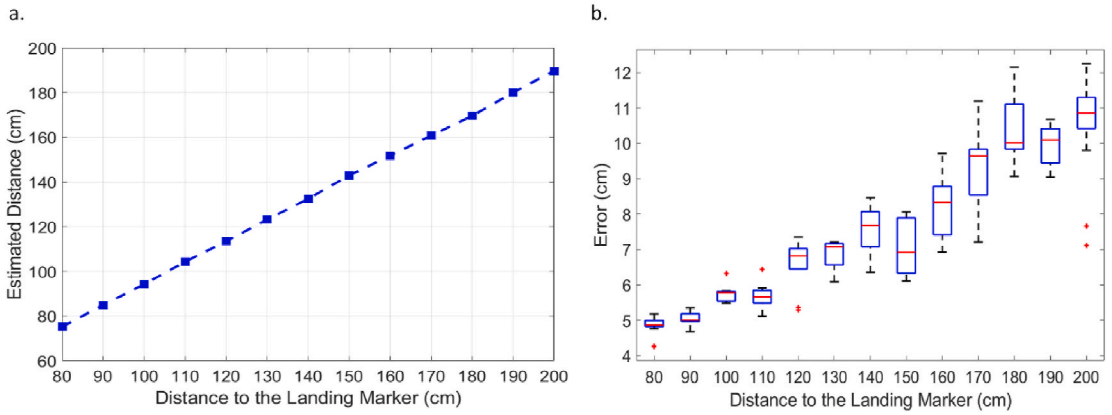


Fig. 11. Heuristic Depth Estimation algorithm. a). The mean estimated distance to the landing marker, b). The errors in each distance.

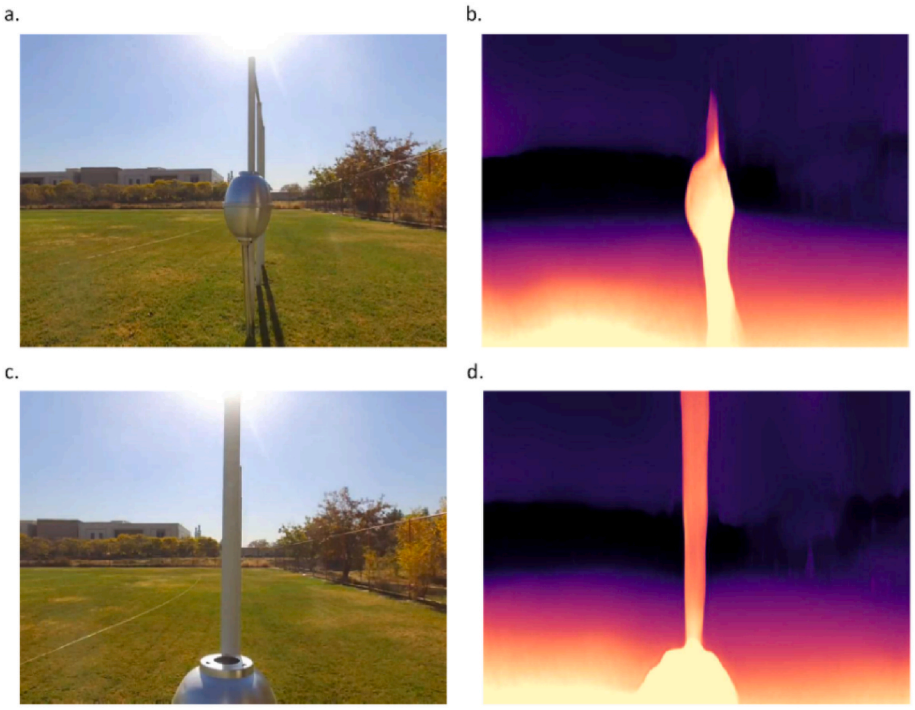


Fig. 12. Obstacle founded in the path of the aircraft. a) Drone camera view test1, b) Depth map test1. c) Drone camera view test2, b) Depth map test2.

Table 1
The comparison between pre-processing techniques.

	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Monodepth2 (full)	0.115	0.903	4.863	0.193	0.877	0.959	0.981
Global Histogram Eq.	0.118	0.923	4.924	0.196	0.867	0.957	0.981
Local Histogram Eq.	0.117	0.929	4.908	0.194	0.873	0.958	0.981
Noise removal	0.117	0.947	4.953	0.195	0.875	0.958	0.980
Gamma filer ($\Gamma = 1.1$)	0.116	0.914	**4.862**	0.194	0.875	0.959	0.981
Gamma filer + Brightness Adjusting	0.116	0.916	**4.854**	0.194	0.873	0.959	0.981
Brightness Adjusting	**0.114**	**0.896**	4.844	**0.192**	0.877	0.959	0.981

Some experimental tests are designed to evaluate the accuracy of each small part and also the accuracy of the mission that is performed by the platform. The testing scenarios consist of.

1. Testing the *Depth Estimation* module
2. Testing the landing operation
3. Testing the *Collision Avoidance* module
4. Testing the whole operation

The next sections are a complete demonstration of the 4 testing scenarios.

4.3.1. Depth estimation module

To overcome the erroneous GPS signals two kinds of visual markers are used to calibrate the spatial information supplies with the drone IMU. The *Depth Estimation* algorithm is used to compute the distance to each marker to make sure that the aircraft could accomplish its mission accurately. The *Depth Estimation* algorithm which is previously described is the classical machine vision algorithm that is used to estimate the distance to a previously known object. In this study, we developed a heuristic approach for detecting the distance to each kind of marker. Given that the shapes of markers are already known; the area of a specific rectangle in each marker is used to find the distance to each one.

First, the specific rectangle in each marker is detected with the help of the *Threshold* module and the *connected component* algorithm in the OpenCV library and then, the area of the rectangle is computed. The areas of the specific rectangles are computed in 260 images of the landing marker and 195 images of the QR code from different distances (80 cm–200 cm). Two equations are made based on distances and the mean areas in each distance which is shown in Fig. 9. So, in each frame, the distance to the marker can be computed using the area of the specific rectangle. This method provides promising results when the angles of the camera and markers are fixed to each other.

In the first testing scenario, to measure the accuracy of the distance provided by the *heuristic Depth Estimation* module, the aircraft was manually located at different distances to the wall while the camera is parallel to the QR code or slightly deviated in the z-direction direction. This operation was repeated within the range of 80 cm–200 cm far from the QR code and 15 images are taken at each distance. The actual distances are compared with outputs of the *heuristic Depth Estimation* module in Fig. 10 (a) and the errors in each test are shown in Fig. 10 (b).

Similarly, the distance to the landing marker is measured by the same approach except that the camera axis is always fixed and the landing marker might be a bit twisted in the x and y-direction. 25 images are taken at each distance. Fig. 11 (a) shows the mean estimated distance to the landing marker compared with the real distance and the errors in each distance are shown in Fig. 11 (b).

The results (Figs. 10 and 11) indicate that the errors increase at far distances because the area of the specified rectangle is smaller and a slight change in the area can make a bigger error in the computed distance.

4.3.2. Landing operation

After the aircraft calibrates its position with the QR code, the landing operation is started.

In the second testing scenario, to evaluate the performance of the landing operation, the drone started the flight from different positions and completed the landing operation. This test is done 6 times and the distance between the center of the landing marker and the center of the drone camera is measured at each test. The mean error of the landing operation was 6 cm after the tests.

The landing approach has promising results compared with the same approaches. The mean error of the landing operation is 6 cm in this paper where it is measured 6 cm in Ref. [58], 11 cm in Ref. [59] and, 13 cm in Ref. [55]. But because of different effective parameters like the environmental conditions, aircrafts type, and discrepancy in scenarios this comparison is not so accurate but could provide a general view about the system.

4.3.3. Collision avoidance module

As it has been shown in Fig. 1, any possibility of collision during the flight is checked with the *Collision Avoidance* module. In the third

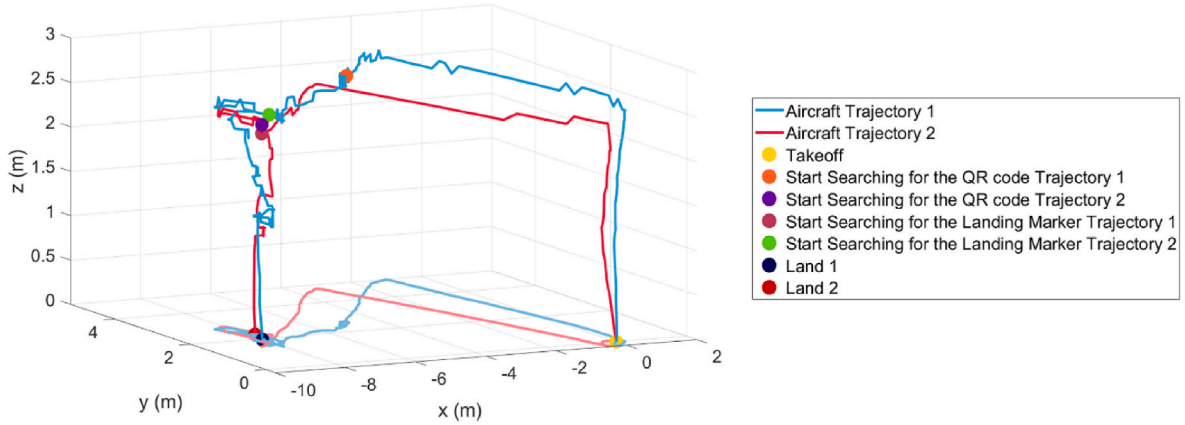


Fig. 13. The 3D trajectory of two flight tests started from the same place.

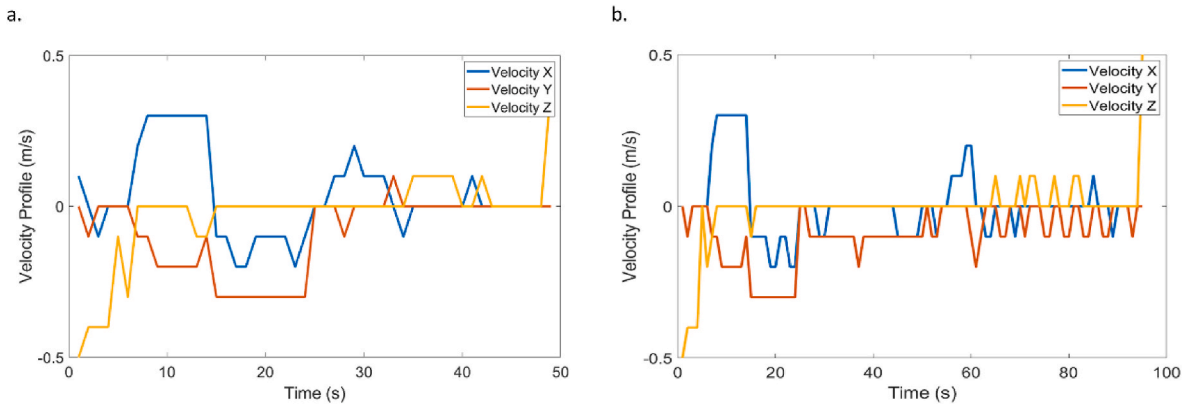


Fig. 14. Velocity profile of two tests - a. test1 b. test2.

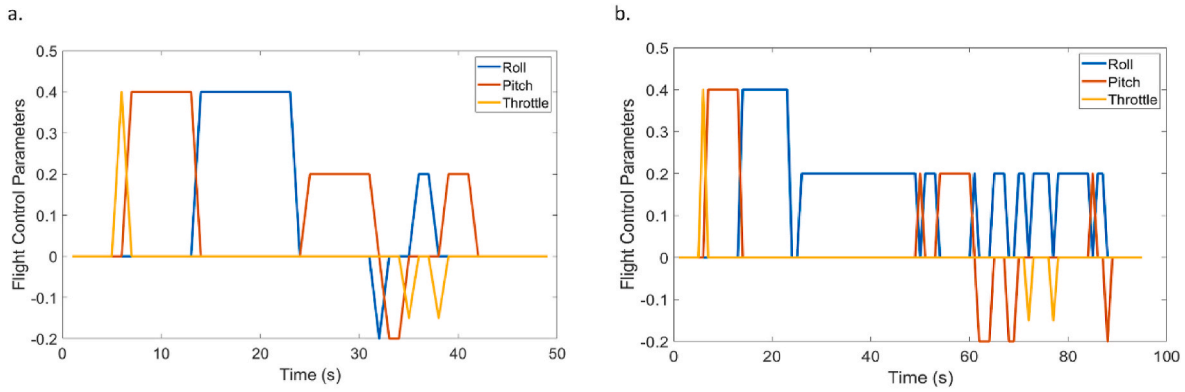


Fig. 15. Roll, pitch, throttle profile of two tests - a. test1 b. test2.

scenario, to evaluate the *Collision Avoidance* algorithm a self-designed obstacle (Fig. 12 (a),(c)) was located in the path of the aircraft. The aircraft is flying and when it overtakes the specific distance to the obstacle (e.g. in this study we set the distance 74 cm), the high-risk event is detected with the help of the *Collision Avoidance* module and the flight is canceled. The results of the two tests are shown in Fig. 12.

In the first test, the aircraft is flying at a certain speed towards the obstacle. Fig. 12 (a) is the drone camera view after detecting the obstacle with the *Collision Avoidance* module in the first test. Fig. 12 (b) shows the depth map of the camera view in the same test.

In the next test the aircraft moves to a closer distance to the obstacle and a high-risk event is detected when the obstacle is presented in the drone camera view (Fig. 12 (c), (d)).

4.3.4. Monodepth2 evaluation

Some preprocessing techniques are tested to increase the performance of the monodepth2 algorithm. In order to compare the effect of these techniques with the previous approaches, the changes are evaluated on the KITTI dataset [76]. The Eigen test split is used in the evaluation. The evaluation information is shown in Table 1 for each approach.

The preprocessing techniques consist of median noise removal, local and global histogram equalization, brightness adjusting using simple transformation, and gamma filter.

The results indicate that brightness adjusting can improve the accuracy of the Monodpepth2 but, other methods have a negative impact. The brightness adjustment is applied in the testing.

The improvement is less compare to D3VO [71] which use the brightness transformation in the training phase, but our simple solution improves the results without any training.

This shows that applying the brightness adjustment between the train and test images can lead to a good enhancement in the Monodepth2 performance.

We also change one of the coefficients in the encoder network of Monodepth2. The input image is divided by 0.225 before entering ResNet18, which cause increase in contrast. We growth this increase a bit by changing this coefficient into 0.205. This has a positive impact on the performance.

4.3.5. Real-world experiment

As it has been demonstrated before, the real-world tests are performed in the designed experimental environment (Fig. 8).

In the fourth scenario, the drone started the missions from the specified home position and ended the missions by landing on the provided landing marker. After traversing the calculated path up to the wall, the drone coordinates are calibrated by the QR code that is labeled on the wall and the landing marker that is placed nearby. This scenario is repeating two times from the same home position. To have a better view of the system performance, the 3D trajectory of two flight tests is shown in Fig. 13. The aircraft started the missions from the same home position but because of the conditions such as engine vibrations and wind the flight path is not the same in the two testing scenarios. Eventually, after calibrating with the visual markers, the aircraft landed in the same place in both scenarios.

Also, the velocity profile during the mentioned tests is shown in Fig. 14 and the flight parameter includes roll, pitch, and throttle during these two tests are shown in Fig. 15. The yaw parameter is always constant because the drone does not rotate during the flight. The test

duration was 95s and 49s in the two tests respectively. This difference is because of the difference in the *Calibration* time.

The throttle parameter changes make the aircraft move vertically. The roll parameter leads the aircraft to move forward or backward and the pitch parameter is used to make the aircraft move to left or right. As is shown in Figs. 14 and 15 the velocity is changed based on the aircraft movements.

5. Discussion

The introduced platform could automatically manage the delivery requests for lightweight material/instrument in construction sites. The platform consists of a mobile application that is designed to handle the material/instrument supply mission and a low-cost micro UAV that is responsible for the transfer of material/instrument. The application is powered with a machine vision algorithm that could minimize the deviation that might happen because of the inaccurate GPS signals. Also, a deep-learning-based algorithm is integrated into the system that is responsible for the safety of the aircraft during the missions.

The platform could help the construction sites to increase the efficiency of workers by reducing the wasted time for vertical transportation. It could also reduce the workload of the *Vertical Transportation Equipment* so they can be used for transferring the heavy materials and workers.

It is worthy to mention that the proposed method just focuses on a proof-of-concept of using autopilot based tools in construction. The performed tests do not represent %100 realistic field situations of complex construction sites. For future works, investigation the weather and visibility limitations are recommended to be taken into account.

6. Conclusion

This paper introduced a proof of concept solution for automating the vertical transportation procedure in construction sites. A platform is developed that could automatically manage material/instrument handling requests via an autopilot drone. UAVs are chosen for doing the mission due to their ability to fly and reach hard-to-access areas at a reasonable cost. The platform consists of a mobile application that is designed for the mission definition and a light-weight micro UAV that is in charge of handling the hardware side. To overcome the limitations caused by the erroneous GPS signal which is normally occurred in urban areas, a machine vision algorithm is implemented to calibrate the position of the aircraft. The flight path will be calculated in the mobile application based on the home position and the target position specified by the user. The aircraft will update its spatial position with the help of two visual markers during a flight.

Different real-world tests are performed to evaluate the introduced system. The results show that the entire process from (mission definition, *Threshold*, *Depth Estimation*, *Calibration*, and *Collision Avoidance*) could be handled by the platform within promising results. This platform could assist construction sites to more easily respond to delivery requests to reduce the workload of the *Vertical Transportation Equipment* and workers.

To the best of our knowledge, this is among the first attempts in using UAVs to automate the vertical transportation procedure in construction sites utilizing UAVs. The proposed approach can remarkably reduce the time that is wasted for lifting materials/instruments in the construction sites and make a great enhancement in the construction progress.

CRediT authorship contribution statement

Zeiab Imani: Software, Investigation, Formal analysis, Data curation. **Perry Forsythe:** Funding acquisition. **Alireza Ahmadian Fard Fini:** Writing – review & editing, Supervision, Formal analysis, Conceptualization. **Mojtaba Maghrebi:** Supervision. **Travis S. Waller:** Supervision, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Y. Shin, H. Cho, K.-I. Kang, Simulation model incorporating genetic algorithms for optimal temporary hoist planning in high-rise building construction, Autom. ConStruct. 20 (5) (2011) 550–558, https://doi.org/10.1016/j.autcon.2010.11.021.
[2] C.-Y. Cho, Y. Lee, M.-Y. Cho, S. Kwon, Y. Shin, J. Lee, An optimal algorithm of the multi-lifting operating simulation for super-tall building construction, Autom. ConStruct. 35 (2013) 595–607, https://doi.org/10.1016/j.autcon.2013.01.003.
[3] M. Park, S. Ha, H.-S. Lee, Y.-k. Choi, H. Kim, S. Han, Lifting demand-based zoning for minimizing worker vertical transportation time in high-rise building construction, Autom. ConStruct. 32 (2013) 88–95, https://doi.org/10.1016/j.autcon.2013.01.010.
[4] M. Jung, J. Moon, M. Park, H.-S. Lee, S.U. Joo, K.-P. Lee, Construction worker hoisting simulation for sky-lobby lifting system, Autom. ConStruct. 73 (2017) 166–174, https://doi.org/10.1016/j.autcon.2016.10.002.
[5] M. Maghrebi, S.T. Waller, C. Sammut, Optimality gap of experts' decisions in concrete delivery dispatching, J. Build. Eng. 2 (2015) 17–23, https://doi.org/10.1016/j.jobe.2015.04.001.
[6] R. Sacks, M. Radosavljevic, R. Barak, Requirements for building information modeling based lean production management systems for construction, Autom. ConStruct. 19 (5) (2010) 641–655, https://doi.org/10.1016/j.autcon.2010.02.010.
[7] P.M. Asaro, The labor of surveillance and bureaucratized killing: new subjectivities of military drone operators, Soc. Semiotic. 23 (2) (2013) 196–224, https://doi.org/10.1080/10350330.2013.777591.
[8] M.A. Ma'Sum, et al., Simulation of intelligent unmanned aerial vehicle (uav) for military surveillance, in: 2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS), IEEE, 2013, pp. 161–166, https://doi.org/10.1109/ICACSIS.2013.6761569.
[9] A. Goodchild, J. Toy, Delivery by drone: an evaluation of unmanned aerial vehicle technology in reducing CO2 emissions in the delivery service industry, Transport. Res. Transport Environ. 61 (2018) 58–67, https://doi.org/10.1016/j.trd.2017.02.017.
[10] C.C. Murray, A.G. Chu, The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery, Transport. Res. C Emerg. Technol. 54 (2015) 86–109, https://doi.org/10.1016/j.trc.2015.03.005.
[11] V. Puri, A. Nayyar, L. Raja, Agriculture drones: a modern breakthrough in precision agriculture, J. Stat. Manag. Syst. 20 (4) (2017) 507–518, https://doi.org/10.1080/09720510.2017.1395171.
[12] C. Zhang, J.M. Kovacs, The application of small unmanned aerial systems for precision agriculture: a review, Precis. Agric. 13 (6) (2012) 693–712, https://doi.org/10.1007/s11119-012-9274-5.
[13] S. Chowdhury, A. Emelogu, M. Marufuzzaman, S.G. Nurre, L. Bian, Drones for disaster response and relief operations: a continuous approximation model, Int. J. Prod. Econ. 188 (2017) 167–184, https://doi.org/10.1016/j.ijpe.2017.03.024.
[14] M. Quaritsch, K. Kruggl, D. Wischounig-Strucl, S. Bhattacharya, M. Shah, B. Rinner, Networked UAVs as aerial sensor network for disaster management applications, E I Elektrotechnik Inf. 127 (3) (2010) 56–63, https://doi.org/10.1007/s00502-010-0717-2.
[15] B. Rabta, C. Wankmüller, G. Reiner, A drone fleet model for last-mile distribution in disaster relief operations, Int. J. Disaster Risk Reduc. 28 (2018) 107–112, https://doi.org/10.1016/j.ijdrr.2018.02.020.
[16] J.C. Hodgson, S.M. Baylis, R. Mott, A. Herrod, R.H. Clarke, Precision wildlife monitoring using unmanned aerial vehicles, Sci. Rep. 6 (1) (2016) 1–7, https://doi.org/10.1038/srep22574.
[17] S. Lee, Y. Choi, Reviews of unmanned aerial vehicle (drone) technology trends and its applications in the mining industry, Geosystem Engineering 19 (4) (2016) 197–204, https://doi.org/10.1080/12269328.2016.1162115.
[18] J. Fleureau, Q. Galvane, F.-L. Tariolle, P. Guillotel, Generic drone control platform for autonomous capture of cinema scenes, in: Proceedings of the 2nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use, 2016, pp. 35–40, https://doi.org/10.1145/2935620.2935622.
[19] M.C. Tatum, J. Liu, Unmanned aircraft system applications in construction, Procedia Eng. 196 (2017) 167–175, https://doi.org/10.1016/j.proeng.2017.07.187.
[20] S. Bang, H. Kim, H. Kim, UAV-based automatic generation of high-resolution panorama at a construction site with a focus on preprocessing for image stitching, Autom. ConStruct. 84 (2017) 70–80, https://doi.org/10.1016/j.autcon.2017.08.031.
[21] J.A. Besada, et al., Drone mission definition and implementation for automated infrastructure inspection using airborne sensors, Sensors 18 (4) (2018) 1170, https://doi.org/10.3390/s18041170.

[22] H. Freimuth, J. Müller, M. König, Simulating and executing UAV-assisted inspections on construction sites, in: 34th International Symposium on Automation and Robotics in Construction (ISARC 2017), 2017, https://doi.org/10.22260/ISARC2017/0090.
[23] S. Siebert, J. Teizer, Mobile 3D mapping for surveying earthwork projects using an Unmanned Aerial Vehicle (UAV) system, Autom. ConStruct. 41 (2014) 1–14, https://doi.org/10.1016/j.autcon.2014.01.004.
[24] N. Anwar, M.A. Izhar, F.A. Najam, Construction monitoring and reporting using drones and unmanned aerial vehicles (UAVs), in: The Tenth International Conference on Construction in the 21st Century (CITC-10), 2018, pp. 2–4.
[25] J.J. Lin, K.K. Han, M. Golparvar-Fard, "A framework for model-driven acquisition and analytics of visual data using UAVs for automated construction progress monitoring,", Computing in Civil Engineering (2015) 156–164, 2015.
[26] R.R.S. De Melo, D.B. Costa, J.S. Álvares, J. Irizarry, Applicability of unmanned aerial system (UAS) for safety inspection on construction sites, Saf. Sci. 98 (2017) 174–185, https://doi.org/10.1016/j.ssci.2017.06.008.
[27] J. Irizarry, M. Gheisari, B.N. Walker, Usability assessment of drone technology as safety inspection tools, J. Inf. Technol. Construct. 17 (12) (2012) 194–212.
[28] R. Ashour, et al., Site inspection drone: a solution for inspecting and regulating construction sites, in: 2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS), IEEE, 2016, pp. 1–4, https://doi.org/10.1109/MWSCAS.2016.7870116.
[29] E. Belyaev, S. Forchhammer, An efficient storage of infrared video of drone inspections via iterative aerial map construction, IEEE Signal Process. Lett. 26 (8) (2019) 1157–1161, https://doi.org/10.1109/LSP.2019.2921250.
[30] D. Moon, S. Chung, S. Kwon, J. Seo, J. Shin, Comparison and utilization of point cloud generated from photogrammetry and laser scanning: 3D world model for smart heavy equipment planning, Autom. ConStruct. 98 (2019) 322–331, https://doi.org/10.1016/j.autcon.2018.07.020.
[31] S. Goessens, C. Mueller, P. Latteur, Feasibility study for drone-based masonry construction of real-scale structures, Autom. ConStruct. 94 (2018) 458–480, https://doi.org/10.1016/j.autcon.2018.06.015.
[32] J. Paek, J. Kim, R. Govindan, Energy-efficient rate-adaptive GPS-based positioning for smartphones, in: Proceedings of the 8th International Conference on Mobile Systems, applications, 2010, pp. 299–314, https://doi.org/10.1145/1814433.1814463, *and services*.
[33] C. Godard, O. Mac Aodha, M. Firman, G.J. Brostow, Digging into self-supervised monocular depth estimation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3828–3838.
[34] A. Devos, E. Ebeid, P. Manoonpong, Development of autonomous drones for adaptive obstacle avoidance in real world environments, in: 2018 21st Euromicro Conference on Digital System Design (DSD), IEEE, 2018, pp. 707–710, https://doi.org/10.1109/DSD.2018.00009.
[35] G. Brunner, B. Szebedy, S. Tanner, R. Wattenhofer, The urban last mile problem: autonomous drone delivery to your balcony, in: 2019 International Conference on Unmanned Aircraft Systems (Icuas), IEEE, 2019, pp. 1005–1012, https://doi.org/10.1109/ICUAS.2019.8798337.
[36] V. Usenko, L. Von Stumberg, A. Pangercic, D. Cremers, Real-time trajectory replanning for MAVs using uniform B-splines and a 3D circular buffer, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2017, pp. 215–222, https://doi.org/10.1109/IROS.2017.8202160.
[37] D. Wang, W. Li, X. Liu, N. Li, C. Zhang, UAV environmental perception and autonomous obstacle avoidance: a deep learning and depth camera combined solution, Comput. Electron. Agric. 175 (2020) 105523, https://doi.org/10.1016/j.compag.2020.105523.
[38] A. Al-Kaff, F. García, D. Martín, A. De La Escalera, J.M. Armingol, Obstacle detection and avoidance system based on monocular camera and size expansion algorithm for UAVs, Sensors 17 (5) (2017) 1061, https://doi.org/10.3390/s17051061.
[39] S. Saha, A. Natraj, S. Waharte, A real-time monocular vision-based frontal obstacle detection and avoidance for low cost UAVs in GPS denied environment, in: 2014 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology, IEEE, 2014, pp. 189–195, https://doi.org/10.1109/ICARES.2014.7024382.
[40] A. Kouris, C.-S. Bouganis, Learning to Fly by Myself: A Self-Supervised Cnn-Based Approach for Autonomous Navigation, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 1–9, https://doi.org/10.1109/IROS.2018.8594204, 2018: IEEE.
[41] R.P. Padhy, S. Verma, S. Ahmad, S.K. Choudhury, P.K. Sa, Deep neural network for autonomous uav navigation in indoor corridor environments, Procedia Comput. Sci. 133 (2018) 643–650, https://doi.org/10.1016/j.procs.2018.07.099.
[42] M.A. Anwar, A. Raychowdhury, NavREn-Rl: learning to fly in real environment via end-to-end deep reinforcement learning using monocular images, in: 2018 25th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), IEEE, 2018, pp. 1–6, https://doi.org/10.1109/M2VIP.2018.8600838.
[43] S.-Y. Shin, Y.-W. Kang, Y.-G. Kim, Reward-driven u-net training for obstacle avoidance drone, Expert Syst. Appl. 143 (2020) 113064, https://doi.org/10.1016/j.eswa.2019.113064.
[44] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, N. Navab, Deeper depth prediction with fully convolutional residual networks, in: 2016 Fourth International Conference on 3D Vision (3DV), IEEE, 2016, pp. 239–248, https://doi.org/10.1109/3DV.2016.32.
[45] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, A. Kolb, Real-time 3d reconstruction in dynamic scenes using point-based fusion, in: 2013 International Conference on 3D Vision-3DV 2013, IEEE, 2013, pp. 1–8, https://doi.org/10.1109/3DV.2013.9.
[46] B. Li, C. Shen, Y. Dai, A. Van Den Hengel, M. He, Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1119–1127.
[47] F. Liu, C. Shen, G. Lin, Deep convolutional neural fields for depth estimation from a single image, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5162–5170.
[48] M. Liu, M. Salzmann, X. He, Discrete-continuous depth estimation from a single image, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 716–723.
[49] A. Singla, S. Padakandla, S. Bhatnagar, Memory-based deep reinforcement learning for obstacle avoidance in UAV with limited environment knowledge, IEEE Trans. Intell. Transport. Syst. (2019), https://doi.org/10.1109/TITS.2019.2954952.
[50] T. Baca, et al., Autonomous landing on a moving vehicle with an unmanned aerial vehicle, J. Field Robot. 36 (5) (2019) 874–891, https://doi.org/10.1002/rob.21858.
[51] D. Falanga, A. Zanchettin, A. Simovic, J. Delmerico, D. Scaramuzza, Vision-based autonomous quadrotor landing on a moving platform, in: 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), IEEE, 2017, pp. 200–207, https://doi.org/10.1109/SSRR.2017.8088164.
[52] J. Garcia-Pulido, G. Pajares, S. Dormido, J.M. de la Cruz, Recognition of a landing platform for unmanned aerial vehicles by using computer vision-based techniques, Expert Syst. Appl. 76 (2017) 152–165, https://doi.org/10.1016/j.eswa.2017.01.017.
[53] P.H. Nguyen, M. Arsalan, J.H. Koo, R.A. Naqvi, N.Q. Truong, K.R. Park, LightDenseYOLO: a fast and accurate marker tracker for autonomous UAV landing by visible light camera sensor on drone, Sensors 18 (6) (2018) 1703, https://doi.org/10.3390/s18061703.
[54] N.Q. Truong, P.H. Nguyen, S.H. Nam, K.R. Park, Deep learning-based super-resolution reconstruction and marker detection for drone landing, IEEE Access 7 (2019) 61639–61655, https://doi.org/10.1109/ACCESS.2019.2915944.
[55] O. Araar, N. Aouf, I. Vitanov, Vision based autonomous landing of multirotor UAV on moving platform, J. Intell. Rob. Syst. 85 (2) (2017) 369–384, https://doi.org/10.1007/s10846-016-0399-z.
[56] A. Borowczyk, D.-T. Nguyen, A. Phu-Van Nguyen, D.Q. Nguyen, D. Saussié, J. Le Ny, Autonomous landing of a multirotor micro air vehicle on a high velocity ground vehicle, IFAC-PapersOnLine 50 (1) (2017) 10488–10494, https://doi.org/10.1016/j.ifacol.2017.08.1980.
[57] Y. Feng, C. Zhang, S. Baek, S. Rawashdeh, A. Mohammadi, Autonomous landing of a UAV on a moving platform using model predictive control, Drones 2 (4) (2018) 34, https://doi.org/10.3390/drones2040034.
[58] M.F. Sani, G. Karimian, Automatic navigation and landing of an indoor AR. drone quadrotor using ArUco marker and inertial sensors, in: 2017 International Conference on Computer and Drone Applications (IConDA), IEEE, 2017, pp. 102–107, https://doi.org/10.1109/ICONDA.2017.8270408.
[59] J. Wubben, et al., Accurate landing of unmanned aerial vehicles using ground pattern recognition, Electronics 8 (12) (2019) 1532, https://doi.org/10.3390/electronics8121532.
[60] F.J. Romero-Ramirez, R. Muñoz-Salinas, R. Medina-Carnicer, Speeded up detection of squared fiducial markers, Image Vis Comput. 76 (2018) 38–47, https://doi.org/10.1016/j.imavis.2018.05.004.
[61] C. Forster, M. Faessler, F. Fontana, M. Werlberger, D. Scaramuzza, Continuous on-board monocular-vision-based elevation mapping applied to autonomous landing of micro aerial vehicles, in: 2015 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2015, pp. 111–118, https://doi.org/10.1109/ICRA.2015.7138988.
[62] A.Y. Chung, J.Y. Lee, H. Kim, Autonomous mission completion system for disconnected delivery drones in urban area, in: 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), IEEE, 2017, pp. 56–61, https://doi.org/10.1109/ROBIO.2017.8324394.
[63] T. Yang, P. Li, H. Zhang, J. Li, Z. Li, Monocular vision SLAM-based UAV autonomous landing in emergencies and unknown environments, Electronics 7 (5) (2018) 73, https://doi.org/10.3390/electronics7050073.
[64] L. Feng, L. Xiaoyu, C. Yi, An efficient detection method for rare colored capsule based on RGB and HSV color space, in: 2014 IEEE International Conference on Granular Computing (GrC), IEEE, 2014, pp. 175–178, https://doi.org/10.1109/GRC.2014.6982830.
[65] J. Heikkila, O. Silvén, A four-step camera calibration procedure with implicit image correction, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 1997, pp. 1106–1112, https://doi.org/10.1109/CVPR.1997.609468.
[66] C. Luo, et al., Every pixel counts++: joint learning of geometry and motion with 3d holistic understanding, IEEE Trans. Pattern Anal. Mach. Intell. 42 (10) (2019) 2624–2641, https://doi.org/10.1109/TPAMI.2019.2930258.

[67] A. Ranjan, et al., Competitive collaboration: joint unsupervised learning of depth, camera motion, optical flow and motion segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 12240–12249.
[68] C. Wang, J.M. Buenaposada, R. Zhu, S. Lucey, Learning depth from monocular videos using direct methods, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2022–2030.
[69] H. Zhan, R. Garg, C. Saroj Weerasekera, K. Li, H. Agarwal, I. Reid, Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 340–349.
[70] T. Zhou, M. Brown, N. Snavely, D.G. Lowe, Unsupervised learning of depth and ego-motion from video, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1851–1858.
[71] N. Yang, L.v. Stumberg, R. Wang, D. Cremers, D3vo: deep depth, deep pose and deep uncertainty for monocular visual odometry, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1281–1292.
[72] Shenzhen DJI Sciences and Technologies Ltd [Online]. Available: https://www.dji.com/no.
[73] *Dji Mobile SDK* [Online]. Available: https://developer.dji.com/mobile-sdk/.
[74] OpenCV (Open Source Computer Vision Library) [Online]. Available: https://opencv.org/.
[75] *Chaquopy 8.0* [Online]. Available: https://chaquo.com/chaquopy/doc/8.0/android.html.
[76] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 3354–3361.