A new method for monitoring machinery movement using an Unmanned Aerial Vehicle (UAV) system

Shihao Sun^a

^aDepartment of Construction Management and Engineering, University of Twente, Enschede, 7522 NB, Netherlands

Abstract

Hot Mixed Asphalt (HMA) is one of the most used materials in road construction. To receive a longer lifespan of asphalt and lower maintenance cost, quality control during asphalt paving is crucial. The temperature of the asphalt mixture is considered a key factor that impacts the final performance of the pavement. The temperature of the asphalt mix is changing continuously during the compaction. If the asphalt is not compacted within a suitable temperature window, it will reduce cracking toughness and raise the chances of crack propagation. Researchers are currently using thermal sensors to monitor the temperature both in paving and compaction process. Also, Global Positioning Systems (GPS) is used to monitor the movement of road construction equipment (i.e., pavers and rollers). In these systems, the geo-referenced temperature data are displayed to operators, so that they can develop their strategies during the paving process. However, the current monitoring system has several limitations: (1) the system requires a high initial investment, (2) instruments need to be mounted on every machine, setup and adjustment of these instruments are time-consuming, and (3) high density of buildings and tall trees disturb or even block GPS signals in some construction environments. On the other hand, Unmanned Aerial Vehicles (UAVs) offer a potential solution to the three limitations. UAVs are becoming increasingly more mature and available and they started to be used in the civil engineering industry as a data acquisition platform and an instrument for surveying purposes. This research aims to investigate the applicability of UAV-based monitoring systems for paving operations. To this end, a UAV-based method for monitoring the movement of the paving machinery during road construction is developed. In this method, markers are placed on (1) static known locations on the site, and (2) moving equipment. Computer vision and photogrammetry methods are used to localize moving equipment in each frame of the video based on location of known markers. The performance of the proposed method is tested in several case studies. The UAV based solution is found to be a promising method for tracking road construction machinery. The solution can reduce initial investment and at the same time ensure adequate accuracy of tracking targets. In addition, UAVs allow adding or replacing several components based on demand. For example, onboard computer for real-time data processing and thermal camera for temperature monitoring. Thus, the UAV based solution can be extended to both monitor paving operations and temperature during road construction in the future.

Keywords: Global positioning system (GPS), Unmanned aerial vehicle (UAV), Asphalt paving, Monitoring, Quality

1 Introduction

The growth of the economy is accompanied by increasing travel demand. As a vital component of any transportation system, there is a need for continuous construction and maintenance of road infrastructure. Stability and durability are two features of asphalt pavement that made it one of the most used material in paved road infrastructure [1]. But many efforts are still put in maintaining the asphalt pavement every year. To receive a longer lifespan of asphalt and lower maintenance cost, quality control is important during asphalt paving process.

In a paving process, the asphalt is first transported from the plant to the construction site. Then, two kinds of construction machinery are used. A paver spread the asphalt to a certain layer thickness and finally rollers compact the asphalt mixture to a certain density and quality level. In the past, asphalt paving process was heavily relied on the skills and experiences of the asphalt team working on the construction site and often without any instruments to monitor the crucial parameters during construction [2]. This is not ineffective because the number of parameters that need to be considered for a quality paving operation is too overwhelming to be left to the intuition or expertise of an operator. One such key parameter is the temperature of the asphalt. Temperature segregation occurs because of differential cooling of portions of the mixture on the surface of the mixture in the haul truck, along with the side of the truck box, and in the wings of the paver [4,5]. Temperature segregation of asphalt mixture can result in density differentials in asphalt layer, which will impact the lifespan of the pavement [5,6]. For example, operational discontinuities, which occurs when the paver stops during the paving process, can cause extensive variability in temperature homogeneity and can directly affect the final quality of the pavement [7]. In addition, temperature of the asphalt mixture is changing continuously during the compaction [8]. The heat affects not only the difficulty of compaction but also the time available for the compaction. The asphalt mixture should be compacted before the temperature of asphalt falls below the lowest bound of ideal compaction temperature, otherwise density progression is hardly reached. Also, the temperature of the pavement cannot be too high. Otherwise, it will damage the asphalt binder. If compaction is outside the compaction window, it will reduce cracking toughness and raise the chances of increasing the crack propagation, although the target density is reached [8]. Therefore, it can be stated that, overall, if asphalt is not compacted within a suitable temperature window, there is a high risk of getting a lower quality pavement.

To enhance process control, researchers has focused on using new technologies to monitor the key parameters and presenting that information to the asphalt team [3]. Researchers are using a set of technologies to professionalize process control on the construction site. To monitor the temperature/density differential and to visualize the collected data systematically, a methodology called Process Quality Improvement (PQi) is developed [3]. Three technologies are used in this framework: (1) Global Positioning System (GPS) receivers mounted on construction equipment to track the movements of construction machinery (paver and roller), (2) laser linescanners mounted at the back of the asphalt paver, (3) infrared cameras placed at fixed positions along the site, (4) thermocouples placed in the middle of asphalt layer to monitor surface and core temperature of the asphalt, and (5) a density gauge to monitor density differential by measuring density after every roller pass. An example of the GPS and linescanner set up on a paver is shown in Figure 1. Data derived from different equipment are downloaded to a processing center and analyzed in real-time. The data provide the asphalt team with the relevant information via appropriate visualizations.

The GPS coordinates of machinery can be integrated with surface/core temperature data from thermal sensors to draw the temperature contour plot, as shown in Figure 2. This georeferenced temperature information can be used both in real time, i.e., to help compactor operators develop compaction strategies, or post-mortem, i.e., to identify the potential areas of early defect in the asphalt layer.



Figure 1: Linescanner mounted behind the paver and a GPS receiver equipped to the paver[9]



Figure 2: Typical Temperature Contour Plot [9]

GPS data can also be used to determine and visualize the amount of compaction force applied to different parts of the asphalt in form of compaction contour plots, as shown Figure 3. The compaction contour plot shows the number of passes on different areas of the

paved layer and the compaction coverage of each roller. It helps to conduct a more detailed analysis of the compaction process. With GPS, the continuity of the asphalt paving process can also be analyzed by calculating the speed of paver. Companies can use the PQI measurements to analyze the strategies of the different operators and the consistency of their strategies.



Figure 3: Typical Compaction Contour Plot derived from the GPS data [9]

As shown above, GPS is a central instrument in PQI methodology to localize and track rollers and pavers. Differential GPS (DGPS) is an enhancement to GPS that can improve the positioning accuracy up to 10 cm [10]. DGPS uses base reference station; i.e., station placed on a known location, to determine the systematic errors of GPS in its vicinity and calculate/transmit the required corrections to the surrounding rovers. In recent years, researchers also started to use more accurate variant of GPS, i.e., Real-time Kinematic (RTK), that works based on carrier phase measurements [11]. RTK GPS can localize objects with the accuracy of 5 cm [11]. Nevertheless, regardless of the variant of GPS used for localization, GPS-based PQI measurements have a number of limitations: (1) the price of GPS instruments is high. As mentioned before, a DGPS configuration setup requires a base station and several rovers. On large projects that many pavers and rollers work together, the initial investment of the system will be high since every equipment on the field should be equipped with a rover; (2) installation, setup, and adjustment of the PQI instruments are timeconsuming; (3) GPS requires a clear line of view to the sky. However, in highly urbanized settings with a high density of buildings and tall trees, GPS signals are blocked. Although signal processing methods, e.g., Kalman filter, can be applied to improve the performance of GPS [12], the accuracy of GPS in urban settings remains significantly low, rendering it impractical for application in PQI measurements.

In recent decades, and with the advent of remote sensing technologies, Unmanned Arial Vehicles (UAVs) has found its way in monitoring civil engineering operations. Two characteristics of UAVs are particularly striking: (1) UAVs are capable of reaching areas or spaces that are difficult to reach otherwise, (2) on top of normal high-resolution cameras, UAVs can be installed with different imagery technologies, e.g., infrared cameras and Light Detection and Ranging (LiDAR), to collect different types of data all through the same platform [13]. This makes UAV a highly versatile platform for monitoring construction activities. In the past few years, the application of UAV for such domains as terrain mapping [14], 3D building reconstruction [15], bridge monitoring [16] and structural health monitoring [17], has been studied. For example, an UAV-based method is developed to build a 3D model of road surface and evaluate its condition [18]. Also, UAVs are used to autonomously monitor linear structures (such as pipelines, roads, bridges, canals) [19].

The researchers found that there are several advantages of UAV monitoring. First, UAVs are flexible and capable of reaching the area or space that a man cannot reach. Second, the data are suitable for post-flight analysis since they are always equipped with high-resolution cameras and capable of doing stationary flight above the target. Third, UAVs are easy to operate and there is a vast number of low cost UAVs available. Although UAVs have proven very efficient for monitoring purposes, there are relatively rare examples of applying UAVs for monitoring of paving operations. Therefore, given the above-mentioned limitations with the current GPS-based monitoring, it will be worthy to research if UAVs can be used in monitoring of paving operations.

1.1 Research Objective

The main objective of this research is to investigate the applicability of UAV-based monitoring systems for paving operations, in particular for PQI measurements. To this end, a marker-based tracking method will be developed, tested, and analyzed to identify the potentials, advantages and limitations of UAVs as a substitute for GPS in the current PQI measurements. Also, by applying the proposed method in several case studies, the functional requirements of UAV-based system for PQI measurements will be elucidated. This research will contribute to the body of knowledge by providing an insight into the possibilities and requirements of UAV-based system for PQI measurements.

1.2 Research Methodology

To reach the research objective, a research methodology (Figure 4) is formulated and designed. The methodology has 4 phases, namely literature review, method development, implementation and prototyping, and case study and validation. In the first phase, the research problem and potential solution is identified through a literature review, as presented above. In addition, the research objective is formulated, as presented in Section 1.1.

According to the research objective, a new UAV-based monitoring method is proposed in phase 2 of the research. The method is developed by building on the camera pose estimation and marker-based tracking of the object of interest. The method has three main stages: (1) camera calibration, (2) marker detection, (3) position estimation. Based on the findings from the literature review and traits of the UAV-based monitoring, the key factors of the monitoring method are identified. These factors are later used in phase 4 to validate the proposed method.

Once the proposed method is developed, it is implemented in a prototype in phase 3 of the research. To investigate the feasibility of the proposed method, the prototype is applied in an indoor test. The result of the test is presented to the researcher to verify if the prototype meets the basic functional requirement.

In the phase 4 of the research, the method is then applied in a case study for validation. The researcher conducted a UAV field test to research the impact of the identified key factors. During the validation phase, the result derived from the current monitoring method, which is elucidated in Section 1, is compared to the result derived from the new method utilizing UAV. Base on the result of the validation, the optimization of the new method is undertaken, if needed. In addition, the data derived from the UAV field test is recorded and archived. In the future, the data will help the researchers and UAV pilots to accumulate knowledge of flying strategy and plan a flight before start monitoring.



Figure 4: Research Methodology

2 Proposed Method

Figure 5 schematically presents the concept of marker-based tracking of paving equipment. In this concept, there are two types of markers: (1) target markers that are placed on pavers/rollers; and (2) feature markers that are placed on several known locations on the construction site. In a nutshell, the goal is to use UAVs, to capture aerial videos of the operations and then localize each target markers in each frame of the video based on the location of feature markers. More precisely, each pixel in the frame can be translated to the coordinate on the Earth based on transformation matrices. For any given frame, the goal is to find these transformation matrices based on the known coordinates of the feature

markers and then apply the matrices to the target markers. A main advantage of this concept is that a monitoring device from the sky has less possibility of either disturbing paving and compaction activities or to be obstructed by the asphalt team or the surrounding objects.



Figure 5: Schematic Representation of the Proposed Concept

2.1 Overview of the proposed method

Figure 6 shows the flowchart of the proposed method. As shown in this figure, the proposed method has three main stages, namely, camera calibration, marker detection, and position estimation. While the calibration is done once for the entire operation, latter two stages are applied on every frame of the video captured from the paving operation.

2.1.1 Camera Calibration

If the focal length of camera is too small, it will introduce a degree of distortion to images, it is necessary to first calibrate the camera. The camera calibration method is explained in the literature [20]. According to the literature, there are two major distortions, namely radial distortion and tangential distortion. Radial distortion makes straight lines appear curved, while tangential distortion makes some areas of the image appear closer than expected. In OpenCV, radial distortions are characterized by coefficients k1, k2, k3 and tangential distortion is characterized by coefficients p1, p2. These coefficients are typically bundled into a distortion coefficients vector, which is a 5×1 matrix containing k1, k2, p1, p2, and k3 [21]. If the images are not distorted significantly, the distortion coefficients can be set as zero. Beside distortion coefficients, intrinsic matrix [K] of the camera can also be

calculated by camera calibration. It is shown that the classic black and white chessboard, see Figure 7, can produce accurate calibration results due to its graphics [20]. By inputting checkerboard images from different angles and finding their intersections, the distortion coefficients and intrinsic matrix [K] can be calculated. It is assumed that these parameters are fixed. Thus, the camera zoom is not allowed during the recording of the camera in order to preserve the same parameters during all the procedure. In addition, to ensure that results are close to the real value, the checkerboard should fill the whole frame during the calibration.



Figure 6: Design of the algorithm



2.1.2 Marker detection

Different visual markers are used in many situations for tracking the object of interest. Given the context of PQI measurement, the 2D tracking of the target object is sufficient. In other words, in the current state of the PQI measurements, the elevation (i.e., Z value) of the paving equipment is not of interest. There are several 2D squared fiducial marker systems such as AR Tag [22] and ARToolkit [23], which were primarily developed for augmented reality (AR) applications. These square markers consist of an external pixel thick black border and an internal area that encodes a binary pattern. The binary pattern is unique for each marker, which means the unique binary pattern encoding the marker identifier will not be created by the rotation of other patterns in the same environment. An example of the 2D squared fiducial marker is shown in Figure 8.



Figure 8: Pattern of a 2D marker

In marker detection process, the goal is to identify the markers and then analyze the binary code inside them. For the code extracted, identification of the internal code is done to check if the type of markers, i.e., target or feature markers. In general, the 2D marker detection process includes 4 steps: (1) applying threshold for image segmentation: in this step, as shown in Figure 9(a), an adaptive threshold is applied to make the detection of markers easier; (2) finding contour of the marker; (3) identifying the type of marker: at the end of the previous step, many unwanted contours that do not belong to markers are also identified. Therefore, it is important to remove these noises to identify the markers. To this end, unwanted contours (too small or too large, too close to each other, etc.) are removed, as shown in Figure 9(b). Next, as shown in Figure 9(c), polygonal approximation is applied to identify the four corners of the rectangular contours. Then, the corners are sorted counter-clockwise and rectangles that are close to one other are removed. This is necessary because adaptive thresholds typically detect the internal and external parts of the marker boundaries. At this stage, as shown in Figure 9(d), the external borders of the marker are retained. Finally, as shown in Figure 9(e), homography is applied to the detected contours to remove the projection perspective from the detected marker. At this point, the marker is detected and the code inside the marker can be retrieved; and (4) finding the pixel coordinates of markers.



Figure 9: Marker detection process

2.1.3 Position Estimation

After the markers are successfully detected, it comes to the next function, which is the position estimation. The proposed method aims to locate pixel point of the center of the target marker in the 2D image and then calculate the corresponding coordinate on the Earth. Therefore, it is vital to get the corresponding relationship between 2D and 3D (Figure 10), which is represented in certain matrices. To better explain the remainder of the method, it is crucial to establish some of the basic terms in coordinate project calculation:

World coordinate (x_w, y_w, z_w) : It is also known as the measurement coordinate system. This is a three-dimensional rectangular coordinate system, which is used to describe the spatial position of the camera and the object to be measured. The position of the world coordinate system can be determined freely according to the actual situation.

Camera coordinate (x_c, y_c, z_c) : It is also a three-dimensional rectangular coordinate system. The origin is located at the optical center of the lens. The x and y axes are parallel to both sides of the phase plane respectively, z axis is the optical axis of the lens, perpendicular to the image plane.

The pixel coordinate system (u, o, v): It is a two-dimensional rectangular coordinate system, which reflects the arrangement of pixels in the camera CCD/CMOS chip. The origin o is located in the upper left corner of the image, and the u and v axes are parallel to both sides of the image plane. Pixel coordinates are in pixels (integers).

Imaging plane coordinate system (x, y): the pixel coordinate system is not conducive to coordinate transformation, so it is necessary to establish an image coordinate system XOY. The unit of its coordinate axis is usually mm, and the origin is the intersection point of camera optical axis and phase plane (called the main point), that is, the center point of the image. X axis and Y axis are parallel to u axis and v axis, respectively. Therefore, the two coordinate systems have actually translation relations. Imaging plane coordinate is ideal (distortion-free) image coordinate. If the image is distorted (although not observed significantly in this research), the distortion can be solved by applying distortion coefficients that derived from camera calibration [20].



Figure 10: Transforming between different coordinate systems (f: focal length) [24]

As mentioned above, the problem of the relationship between 2D and 3D is actually finding the relationship between those different coordinates (Figure 9). Each pixel in the frame can be translated to the coordinate on the Earth by using several matrices. The first step is to transform from the imaging plane coordinate system (pixel coordinate system) to the camera coordinate system, using parameters of physical characteristics of the camera. These parameters include information such as focal length (f_x , f_y) and distance to the optical center (c_x , c_y). These parameters can also be expressed in the camera Intrinsic Matrix [K] that derived from camera calibration. [K] is a 3x3 matrix as shown in Equation 1.

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$
 Equation 1

Where:

K: Intrinsic Matrix

 f_x : The camera focal length in terms of pixel dimension in the x direction

 f_{y} : The camera focal length in terms of pixel dimension in the y direction

 c_x : Distance to the optical center in terms of pixel dimension in the x direction

 c_{y} : Distance to the optical center in terms of pixel dimension in the y direction

The second step is to transform from the camera coordinate system to the world coordinate system. This step is the Euclidean transformation from 3D point to 3D point, using [R] (the rotation matrix that determine orientation of the camera in a 3D space) and t (the translation vector that determine the position of the camera in a 3D space) [25]. As presented in Equation 2, [R|t] is a 3x4 matrix which is a horizontal concatenation of [R] and t, it is also called the extrinsic matrix.

$$\begin{bmatrix} R|t \end{bmatrix} = \begin{bmatrix} R11 & R12 & R13 & t1 \\ R21 & R22 & R23 & t2 \\ R31 & R32 & R33 & t3 \end{bmatrix}$$
Equation 2

Where:

Ri,j: The elements contain Euler angles roll, pitch and yaw, which define a rotation ti: Distance from camera coordinate origin to world coordinate origin in x,y,z direction

If the intrinsic matrix [K] and extrinsic matrix [R|t] are known, the overall transformation matrix (T) can be defined according to Equation 3.

$$[T] = [K] \times [R|t]$$
Equation

3

Where:

T: Transformation matrix

With the transformation matrix known, the corresponding world 3D coordinate of any 2D points in the image can be calculated using Equation 4.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = [T] \times \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$
 Equation 4

Where:

x: Position of point in terms of pixel dimension in the x direction y: Position of point in terms of pixel dimension in the y direction x_w : Position of point in terms of real world dimension in the x direction y_w : Position of point in terms of real world dimension in the y direction z_w : Position of point in terms of real world dimension in the z direction

As stated before, Intrinsic Matrix [K] is based on parameters of physical characteristics of the camera, which is easy to retrieve by doing camera calibration. Given at least four feature points (which are geometric center of feature markers in this research) represented in a real-world reference frame and their corresponding 2D projection points on the image, [R|t] can be retrieved. Real coordinate of any detected target marker can be easily estimated using these matrices.

2.2 Key factors and parameters:

There are many factors that affect a successful UAV-based tracking of paving machinery. These factors include: (1) applicability of the UAV as a monitoring platform, (2) Update rate and accuracy of the estimated position data.

2.2.1 Applicability of the UAV

The selection of a UAV system must be based on the traits characteristics of asphalt pavement construction. The UAV needs to be able to hover in the sky stably and be robust to withstand harsh ambient weather environment, e.g., strong winds. Performance of UAV-based solution will be impacted if the data derived from UAV are not stable. Thus, parameter like max wind resistance, number of rotors, weight and max payload need to be considered. Flight height, flight range and max flight time are also crucial parameters in UAV-based tracking.

2.2.2 Update rate and accuracy

As described above, the proposed method consists of two functions, marker detection and position estimation. Real coordinate of the target marker will be estimated as long as the target marker and at least four feature makers are detected. Tracking update rate of the target position is directly correlated to marker detection ratio. The update rate of UAV-based solution is defined by Equation 5.

$$U = \left(\frac{C}{T_f}\right) \times R$$
 Equation 5

Where:

U: the tracking update rate

- C: the total amount of calculated position of target within the given time
- T_f: the total frames of the recorded video within the given time

R: the frame rate of camera

To ensure a better update rate of the data, it is needed to increase the chance of detecting the markers. In addition, the main concern of the proposed method is the accuracy. The following introduce the factors that might impact marker detection ratio and accuracy of the UAV-based solution:

- Type of the markers
- Margin size of markers
- Focal length of UAV camera
- Height of UAV
- Marker size
- Resolution of image
- Flight strategy of the UAV
- Amount of feature markers

Depending on different types, there are markers with more or fewer bits for analyzing the binary code inside. The more bits, the smaller the chance of a marker being recognized as another marker. However, markers with more bits means that more camera resolution is required for correct detection. It will also take a longer time to extract the inner code and increase the computational load.

As shown in Figure 11, markers are always printed on a paper or another material. Therefore, a wider margin outside the pattern of the marker can make it more distinctive from the ambient environment. Thus, contour extraction is enhanced and marker detection ratio is increased.



Figure 11: Margin of a 2D marker

Focal length of the UAV camera, flight height and marker size together determine the size of a marker in the given image, which is S_m . S_m is the main parameter that determines if the marker can be detected or not. S_m can be determined by Equation 6.

$$S_m = \frac{A_m}{A_t} \times 100\%$$
 Equation 6

Where:

 A_m : the area of pixels that the marker occupied A_t : total pixel area of the given image

Resolution of image is directly determined by the resolution of the camera, higher resolution helps to find the outline and geometry center of marker accurately and thus increase the accuracy of result. However, higher resolution increases the computational load and process time of the designed algorithm, which will impact the performance of UAV-based tracking in real time.

Flight strategy can also be correlated with the accuracy because the UAV hover stationary in the sky or follow the movement of moving target might lead to different quality of the captured video.

Placing more markers means higher marker density in a given area. Although it might not increase the detection ratio of each single marker, adding more feature markers means more redundancy for the method. In other words, it is easier to detect at least four feature markers from a frame to get the extrinsic matrix [R|t] and achieve higher update rate of the target

position data. However, it is unknown if accuracy will be increased when more feature markers are placed in a given area. Also, the disadvantage of adding more feature markers is that it will take longer preparation before UAV starts monitoring.

3 Implementation and Case Studies

A prototype is developed and implemented in a number case studies to investigate the feasibility of the proposed method.

3.1 Prototype development

The prototype is developed using Open Source Computer Vision Library (OpenCV) [25]. A camera with resolution of 1080x720 pixels is used to test the prototype. ArUco is an Open Source library for detecting squared fiducial markers in images [26]. In this research, ArUco markers are chosen because they are proved to be very robust and easily detectable for a very wide range of applications [27]. The ArUco markers used in the case study are 7x7 squares array and yield up to 1024 different patterns. An Example of ArUco markers used in the research is shown in Figure 12.



Figure 12: Patterns of ArUco marker

Geometry center of ArUco markers is easily detectable. However, in some situations, there are several parameters that refer to detection ratio and detection speed can be fine-tuned (See Appendix A).

In OpenCV, the extrinsic matrix [R|t] is calculated by solving the Perspective N-Point problem. The Perspective N-Point problem refers to the problem of obtaining the camera or object posture by calculating the projection relation between several feature points in the real world and several pixel points in the image. Function solvePnP() was used in this research [28]. According to Moreno-Noguer et al. [29], the geometry center of four or more detected feature markers with known coordinates in both the real world and camera pixel plane are required to find the extrinsic matrix [R|t]. Also, the distortion coefficients and intrinsic matrix [K] that derived from camera calibration are used as inputs for solvePnP(). However, the distortion coefficients were negligible. As shown in Equation 3, if the image is distortion-free, only [K] and [R|t] are needed for position estimation. The real coordinate of any detected target marker can be easily estimated using these matrices as long as geometry center of the target marker in pixel coordinate system is known.

3.1.1 Case study 1: Indoor test

In the first case study, a pre-defined ArUco library that contains 1024 patterns is used. Five markers from this library are chosen and printed in the size of 7x7 cm. Four of the markers are placed in four corners with known positions (measured in millimeters) of the test plane, see Figure 13, as feature markers. The position of the left top corner is set at (0,0,0), left bottom corner at (297,0,0), right top corner at (0,420,0) and so on. A target marker was placed in the middle of the two top feature markers. The program was run in real-time, the fixed web-camera, which is mounted on a tripod one meter from the test plane, is used to capture image stream and detect markers. After all five markers on the plane are detected, the contour and unique ID of markers are visualized. Output of the program is the estimated position of the target marker.



Figure 13: Prototype test plane

The position of the target marker is recalculated in every frame. Therefore, the value of the position is calculated and then compared to the known position (0,210,0) of the target marker. To testify the accuracy of algorithm, the procedure of test is repeated 5 times. The result of the accuracy measurements is presented in Figures 14. After the prototype test, it was concluded that the developed program was able to estimate the positions of target marker based on four feature markers at certain accuracy.



Figure 14: Errors of the calculated position of target marker

3.2 Case study 2: UAV field test

UAV field test is designed to gain more insight into how the influential practical factors affect the performance of the proposed method in terms of the detection ratio, update rate, and accuracy. The considered factors are: (1) size of a marker in the given image, i.e., S_m , (2) flying altitude of the UAV, (3) density of feature markers in a given area.

3.2.1 Preparation

Before the case study, several 50×50 cm ArUco markers are prepared. Then, an eight-meter wide road is chosen as the test field and a feature marker is placed every five meters along the road, in total ten feature markers are placed, as shown in Figure 15. GPS coordinate system is used to find the coordinates of the feature markers. The GPS coordinate of the center of each marker is surveyed by a D-GPS equipment (Trimble SPS 851). A trolley is used as a target to simulate the movement of a roller during a paving operation. A target marker is placed on the top of the trolley along with a D-GPS rover antenna, whose data is used as ground truth for accuracy measurements. DJI PHANTOM 4, Figure 16, with a gimbal and a built-in high-resolution camera is used to record the video of the simulated operation. The specifications of the built-in camera are 1) resolution: 4096×2160 pixels, 2) focal length: 20 mm and 3) frame rate: 25p. In addition, distortion coefficients and intrinsic matrix [K] of the UAV camera are derived from calibration. In the test, the data derived from UAV camera are processed offline by the program.



Figure 15: Test field and feature markers placement



Figure 16: DJI PHANTOM 4 with an in-build camera

3.2.2 Impact of Flying Altitude

Since the real size of markers is fixed, the impact of the marker size (S_m) on the performance of the proposed method is simulated by modify the flying altitude of UAV. However, the field of view is unknown. The UAV pilot first conducted a trial by hovering the UAV over test field at different height. It is discovered that all ten feature markers are visible in image at the altitude of 13 meters, so the test starts from this altitude. The trolley is pushed from one side of the test field to the other while the UAV hovering at a certain altitude and monitoring the movement of target. This process is repeated at 8 different altitudes, namely 13 m, 15 m, 17m, 19 m, 21 m, 23m, 25m and 27m.

Although the solvePnP() function could output [R|t] in two different scenarios: 1) stationary camera and feature markers and 2) moving camera and feature markers, the impact of UAV movements on the performance is not fully known. It is assumed that the movement of UAV camera might reduce the quality of recorded data and thus impact the detection ratio. In order to investigate the impact of UAV movements, the same process is repeated for the scenario where the UAV follows the movement of target marker on the trolley and keep the target in the center of camera's view instead. This scenario is also tested at 8 different altitudes.

3.2.3 Impact of Marker Density

Although the developed method only requires four feature markers to calculate [R|t], placing more feature markers is expected to enhance the detection rate by providing redundancy. In other words, it is envisioned to be easier to detect at least four feature markers, and thus achieve higher update rate, when more than 4 feature markers are available in the image. However, derived [R|t] might be different if more than four feature markers are detected and used as inputs of the program. To investigate the impact of marker density on the performance of the method, the test where UAV was hovering at 27m is used. When processing the data, it is possible to modify the algorithm and controlled the number of detected feature markers manually. Because all ten feature markers are visible at altitude of 27m, it is possible to experiment various scenarios where the number of detectable feature markers is ranged between 10 to 4. In doing so, the relationship between the number of feature markers and the accuracy is investigated.

3.2.4 Results

The data derived from UAV are processed offline using a PC. Since the real marker size was fixed, the S_m at each flight height is calculated based on Equation 5. Table 1 lists the flight altitude and the corresponding S_m.

Table 1: Flight height and correspondent S_m												
Height	13m	15m	17m	19m	21m	23m	25m					
S_m	0.075%	0.068%	0.061%	0.054%	0.047%	0.040%	0.033%					

Table 1:	Flight	height	and	corres	pondent	S_m
----------	--------	--------	-----	--------	---------	-------

Coordinates from the D-GPS rover is used as ground truth. To estimate the error, the distance between (1) the coordinates registered by the D-GPS rover and (2) the coordinates estimated by the proposed method is calculated. First the distribution of errors is studied. Then, a cumulative distribution of errors is used to estimate confidence intervals of errors. In this research, confidence intervals of 95%, 75% and 50% are considered. For example, when the error at confidence interval of 95% is 1 meter, it means 95% of all the calculated errors are smaller than 1 meter. The tracking update rate of D-GPS rover is 1 Hz. The tracking update rate of UAV based solution is calculated based on Equation 4.

The result of the accuracy measurements is presented in Figures 17 to 19. Figure 17 shows the impact of marker size (S_m) on the accuracy when the camera was fixed. From the figure, it is clear that the accuracy of the UAV-based tracking is not noticeably impacted by S_m . Figure 18 shows the errors when different number of feature markers are used. Again, no obvious pattern in the errors is observed. It is evident from the figure that, against expectations, the higher marker density does not necessarily result in a higher accuracy. For instance, at 95% error confidence interval, the errors of 9, 7, and 5 markers are higher than 4 markers. This is because, GPS coordinate of feature marker was surveyed by a D-GPS equipment manually, which means the accuracy of GPS coordinates of each feature markers were different. In the cases of five or more detected feature markers, all the detected feature markers with certain survey error are used as input of the program, which will impact the accuracy.

Accuracy of two different flight strategy was also compared. Figure 19 shows the accuracy at different marker size (S_m) when the camera was moving (UAV followed the movement of target). Again, from this experiment, the accuracy of the UAV based solution is not impacted by the S_m . However, when compared to Figure 17, the errors for the moving camera are generally higher than the errors for the fixed camera.



Figure 17: Errors at various S_m of 4 feature markers (fixed camera)



Figure 18: Errors at various marker density with the flying altitude of 27 m



Figure 19: Errors at various Sm of 4 feature markers (moving camera)

Although from the experiments, it can be concluded that S_m and the density of feature markers have no observable influence on the accuracy of the proposed method, they effect the tracking update rate. The results of the update rate experiments are presented in Figures 20 and 21. Figure 20 shows the influence of the marker density on the update rate of the monitoring system. In this experiment, UAV hovered at 27m, so the S_m was fixed at 0.026%. From the figure, it is clear that the update rate for the UAV solution decreases when the number of markers is reduced from 10 to four. However, the update rate at four markers is more than 5 Hz, which is still higher than the update rate of D-GPS rover. Figure 21 shows the relationship between the update rate and S_m . By conducting a regression analysis on the data, the two variables seem to have a linear relationship. Based on the regression analysis, two more points are predicted. As shown in the figure, the update rate is 0.91 Hz when S_m is 0.012%.



Figure 20: Update rate at various amount of feature markers on 27m (fixed camera)



Figure 21: Update rate at various S_m of 4 feature markers (fixed camera)

4 Discussion

From the UAV field test, it can be concluded that the proposed method can monitor the movement of a target equipment on the road by tracking the position accurately. The method is shown to have a higher update rate than the current monitoring method in general. Base on the findings from the result, to ensure the accuracy of derived data, it is recommended that the UAV hover in the sky during monitoring instead of following the movements of targets. There are two kinds of target equipment in asphalt paving projects, the paver and roller. The paver can move up to 1m/s and the roller can move up to 2.5m/s. There are several ways to increase the update rate of the propose method. It can be realized by increasing the amount of feature markers or increase size of a marker in the given image, which means using bigger markers or flying the UAV at a lower height.

There are several steps that need to be taken to ensure a better performance of the proposed method. First, camera calibration is necessary, every camera has unique camera calibration parameters because of the accuracy deviation of the manufacturing process. In the proposed method, camera calibration parameters are fixed as long as the same camera is used. Second, the feature markers are required to be placed in the field before monitoring and have a good variation on at least 2 axis. Thus, it is not idea to place the feature markers on a line as reference points. Third, after placing the feature markers, some errors might be introduced when surveying the GPS coordinate of marker center, it is hard to ensure that the position of D-GPS antenna and the center of markers are overlapping. These errors are considered as random errors, they are always present and cannot be eliminated, however, their impact on the result of the measurements can be minimized by conducting redundant surveys. The proposed method requires a low initial investment. For example, the DJI Phantom 4 type cost 800 euros. Although flight time is limited to approximately 30 minutes, it only takes few minutes to replace the batteries on the UAV and it is possible to charge spare batteries on-site.

There are several limitations of the UAV based monitoring method. (1) The worldwide use of UAVs is regulated by the national government and specific local administrations. The use of UAV for research and companies is considered as business use. For safety and security reasons, UAV for business use needs to be registered in the aircraft register and operated by a registered pilot. In addition, in some countries, UAVs are forbidden in some regions. (2) The proposed method has systematic errors that cannot be avoided. The GPS coordinates of the feature markers are used as input for the method. All the coordinates of the feature markers are surveyed by the D-GPS equipment to about 5-10 cm accuracy for the time being. This means that the coordinates always deviate from the true value slightly. (3) Similar to the current method, the proposed method is also constrained by obstructions. In real asphalt construction environment, the UAV is flexible enough to avoid the obstruction of tall trees. However, the monitoring of the UAV will be interrupted if equipment makes too much smoke and block the view of camera. In addition, the UAV cannot work in some construction situation like in tunnels.

5 Conclusions and future works

A novel UAV based approach for monitoring of paving equipment was presented in this research. The proposed method uses a UAV as a platform for the detection of square fiducial marker. The method is found to be promising for the purpose of equipment tracking, mainly because it can reduce initial investment without compromising the accuracy of tracking equipment significantly. The paper explained the development of a computer vision algorithm as well as implementation of the algorithm. The proposed method was evaluated in a test and its performance was assessed. Factors that influence the performance and the relationship between different key variables are investigated.

Future work may address the limitations of this novel UAV based solution. In the next step of this research, there are many functions to be optimized; (1) In the UAV field test, only one target marker was placed and tracked. The algorithm can be modified to support tracking multi targets in a given area. (2) All the data was computed offline for the time being. However, by mounting an onboard-PC on the UAV, real-time data processing can be realized. It is worthy to mention that the UAV platform should have enough payload. (3) Kalman filter is always used to avoid noise in GPS signal tracking. It can also be used in the solvePnP() function to optimize the developed algorithm. (4) Many types of UAV now support carrying various types of additional sensors, for example, range or thermal sensors. UAV can be used in the PQi measurements for both asphalt temperature and equipment tracking. By combing temperature and location data, surface temperature at any given position can be estimated.

6 Acknowledgements

The author wish to thank drone expert Tim, and practitioners from Space 53.

Appendix A

Main parameters for ArUco marker detection

In some cases, it is needed to tune several parameters refer to the detection rate and detection speed.

- Aruco-minSize: ArUco marker developed and defines a way to increase detection speed by using smaller images. However, the accuracy has not been affected by this fact, only the computation time has been reduced. To be generic and adaptable to any image size, the minimum marker size is expressed as a standardized value (0,1), which represents the minimum area that the tag must occupy in the image to be considered valid. A value of 0 indicates that all marker is considered valid. When the minimum size is modified to 1, only larger markers are detected.
- Aruco-detectMode:

DM_NORMAL: this is a case where you need to detect the markers in the image and don't care much about the computation time. This is usually the case in batch processing where computation time is not an issue. In this case, we adopt a very robust local adaptive threshold method.

DM_FAST: in this case, you care about speed. Then, the global threshold method is used to search the best threshold randomly. It works in most cases.

DM_VIDEO_FAST: this is specifically designed to work with video sequences. In this mode, each frame automatically determines a global threshold and automatically determines the minimum markers size to achieve the maximum speed. If the markers you see in one image is very large, then search only for similar-sized markers in the next frame.

• Aruco-borderDistThres: It is defined as: Markers with corners at a distance from image boundary nearer than (0,1) % of image are ignored.

To fine tune the parameters for detection, it is allowed to modify all the parameters and save the configuration in a YML file, an example configuration is displayed in the figure below.

%YAML:1.0

- - -

```
aruco-dictionary: ARUCO
 aruco-detectMode: DM NORMAL
 aruco-cornerRefinementM: CORNER_SUBPIX
 aruco-thresMethod: THRES_ADAPTIVE
 aruco-maxThreads: 1
 aruco-borderDistThres: 1.4999999664723873e-02
 aruco-lowResMarkerSize: 20
 aruco-minSize: 9.9999997764825821e-03
 aruco-minSize_pix: -1
 aruco-enclosedMarker: 0
 aruco-NAttemptsAutoThresFix: 3
 aruco-AdaptiveThresWindowSize: -1
 aruco-ThresHold: 40
 aruco-AdaptiveThresWindowSize_range: 0
 aruco-markerWarpPixSize: 5
 aruco-autoSize: 0
 aruco-ts: 2.5000000000000000e-01
 aruco-pyrfactor: 2.
 aruco-error_correction_rate: 2.0000000000000000e-01
Figure 22: ArUco Detection parameter configuration
```

Source code of the program

```
1.
        #include <stdio.h>
    #include <conio.h>
2.
3.
        #include <iostream>
4.
      #include <vector>
        #include <string>
5.
6.
      #include <math.h>
7.
       #include <fstream>
8.
      #include "opencv2/opencv.hpp"
        #include "aruco.h"
9.
10. #include "opencv2/core.hpp"
11. #include "opencv2/highgui.hpp"
12. #include "opencv2/imgproc.hpp"
       #include "opencv2/calib3d/calib3d.hpp"
13.
14.
15.
        using namespace std;
16.
       using namespace aruco;
17.
       using namespace cv;
18.
19.
        struct MARKERS
20.
       {
        public:
21.
22.
        int id;
            cv::Point3f location;
23.
24.
25.
26.
            MARKERS(cv::Point3f location, int id)
27.
            {
28.
                this->id = id;
29.
                this->location = location;
30.
            3
31.
        };
32.
33.
        int main()
34.
       {
            cout << "Start detecting..." << endl;</pre>
35.
36.
            double camD[9] = { 2.3612e+03 , 0, 2.3589e+03 ,
37.
            0, 1.9696e+03 , 1.4421e+03,
38.
            0, 0, 1};
            double distCoeffD[5] = { 0,0,0,0, 0};
39.
```

```
40.
41.
            Mat camera matrix = Mat(3, 3, CV 64FC1, camD);
42.
            Mat rvec = Mat::zeros(3, 1, CV_64FC1);
43.
            Mat tvec = Mat::zeros(3, 1, CV_64FC1);
44.
45.
            Mat distortion_coefficients = Mat(5, 1, CV_64FC1, distCoeffD);
46.
47.
            CameraParameters LAPTOPCAMParam;
48.
            LAPTOPCAMParam.CameraMatrix = camera_matrix.clone();
49.
            LAPTOPCAMParam.Distorsion = distortion_coefficients.clone();
50.
51.
            Mat frame, frameCopy;
            VideoCapture cap(CAP DSHOW)
52.
53.
            cap.open("C:\\20.mov");
54.
            if (!cap.isOpened())
55.
                return -1;
            cap.set(CAP_PROP_FRAME_WIDTH, 1280);
56.
57.
            cap.set(CAP_PROP_FRAME_HEIGHT, 720);
58.
            cap.set(CAP_PROP_FPS, 30);
59.
            cout << cap.get(CAP_PROP_FRAME_COUNT) << endl;</pre>
            cout << "Frame Width: " << cap.get(CAP_PROP_FRAME_WIDTH) << "\tFrameHeigh</pre>
60.
t: " << cap.get(CAP_PROP_FRAME_HEIGHT) << "\tFPS: " << cap.get(CAP_PROP_FPS) << endl</pre>
;
61.
62.
            vector<MARKERS> knownMarkers;
            knownMarkers.push back(MARKERS(cv::Point3f(52.23631, 6.86102, 0), 101));
63.
            knownMarkers.push back(MARKERS(cv::Point3f(52.23636709, 6.861087939, 0),
64.
102));
            knownMarkers.push back(MARKERS(cv::Point3f(52.23622427, 6.860790617, 0),
65.
103));
            knownMarkers.push_back(MARKERS(cv::Point3f(52.23635059, 6.860912042, 0),
66.
104));
67.
            knownMarkers.push back(MARKERS(cv::Point3f(52.23649196, 6.861065725, 0),
105));
            knownMarkers.push_back(MARKERS(cv::Point3f(52.23618791, 6.860913103, 0),
68.
229));
69.
            knownMarkers.push_back(MARKERS(cv::Point3f(52.2364, 6.86097, 0), 86));
70.
            knownMarkers.push_back(MARKERS(cv::Point3f(52.23623802, 6.860952194, 0),
93));
            knownMarkers.push back(MARKERS(cv::Point3f(52.23627799, 6.860855067, 0),
71.
177));
72.
73.
74.
            MarkerDetector mDetector;
75.
            vector<Marker> mMarkers;
            mDetector.loadParamsFromFile("C:\\para.yml");
76.
77.
            //mDetector.setDictionary("ARUCO", 0.2f); // sets the dictionary to be em
ployed (ARUCO, APRILTAGS, ARTOOLKIT, etc)
78.
            //mDetector.setDetectionMode(DM FAST, 0.01f);//(15m,0.02f)
79.
            while (true)
80.
81.
                cap >> frame; // get a new frame from camera
82.
83.
                frame.copyTo(frameCopy);
84.
85.
                mMarkers = mDetector.detect(frame, LAPTOPCAMParam, 0.5f);
86.
                namedWindow("ThresholdedImage", WINDOW NORMAL);
                imshow("ThresholdedImage", mDetector.getThresholdedImage());
87.
88.
                vector<int> markerIndexFound;
                vector<cv::Point3f> Points3D;
89.
90.
                int markerMovableFound = -1;
91.
                for (unsigned int i = 0; i < mMarkers.size(); i++)</pre>
92.
                {
93.
                    mMarkers[i].draw(frameCopy, Scalar(0, 0, 255), 2, true);
```

```
94.
95.
                    if (mMarkers[i].id == 100) //moveable
96.
97.
                        markerMovableFound = i;
98.
                    }
99.
                    else
100.
                    {
101.
                        for (int j = 0; j < knownMarkers.size(); j++)</pre>
102.
                        {
103.
                             if (mMarkers[i].id == knownMarkers[j].id)
104.
                             {
105.
                                 markerIndexFound.push_back(i);
106.
                                 Points3D.push_back(knownMarkers[j].location);
107.
                                 break;
108.
                            }
109.
                        }
110.
                    }
111.
                }
112.
                if (markerIndexFound.size() >= 4 && markerMovableFound != -1)
113.
                {
114.
                    Point2f x = mMarkers[markerMovableFound].getCenter();//use movabl
e marker100 as input;
                    vector<Point2f> markerinimage;
115.
116.
117.
                    for (int i = 0; i < markerIndexFound.size(); i++)</pre>
118.
                    {
119.
                        markerinimage.push back(mMarkers[markerIndexFound[i]].getCent
er());
120.
                        //cout << markerinimage << endl;</pre>
121.
                    }
122.
                    solvePnP(Points3D, markerinimage, camera_matrix, distortion_coeff
icients, rvec, tvec, false, SOLVEPNP_EPNP);//also try solvePnPRansac. if the camera
is moving, choose false, SOLVEPnP EPNP and SolvePNP P3P have best result
123.
124.
                    double rm[9];
                    Mat rotM(3, 3, CV_64FC1, rm);
125.
126.
                    Rodrigues(rvec, rotM);
127.
                    rotM.ptr<double>(0)[2] = tvec.at<double>(0, 0);
128.
                    rotM.ptr<double>(1)[2] = tvec.at<double>(1, 0);
129.
                    rotM.ptr<double>(2)[2] = tvec.at<double>(2, 0);
130.
                    Mat hu;
                    hu = camera_matrix * rotM;
131.
132.
                    Mat hu2 = hu.inv();
133.
                    double a1, a2, a3, a4, a5, a6, a7, a8, a9;
134.
                    a1 = hu2.at<double>(0, 0);
135.
                    a2 = hu2.at<double>(0, 1);
136.
                    a3 = hu2.at<double>(0, 2);
                    a4 = hu2.at<double>(1, 0);
137.
                    a5 = hu2.at<double>(1, 1);
138.
139.
                    a6 = hu2.at<double>(1, 2);
140.
                    a7 = hu2.at<double>(2, 0);
141.
                    a8 = hu2.at<double>(2, 1);
142.
                    a9 = hu2.at<double>(2, 2);
143.
                    Point2f key;
144.
                    vector<Point2f>realgps1;
145.
                    int xe = x.x;
146.
                    int ye = x.y;
147.
                    key.x = (a1 * xe + a2 * ye + a3) / (a7 * xe + a8 * ye + a9);//
                    key.y = (a4 * xe + a5 * ye + a6) / (a7 * xe + a8 * ye + a9);//
148.
149.
                    if (key.x < 54, key.y < 8)
150.
                    {
151.
                        realgps1.push_back(key);
152.
                    }
153.
                    else { continue; }
154.
                    cout << realgps1 << endl;</pre>
```

```
155.
                     ofstream fout1("./DataFiles/Output.csv", ios::app);
                    fout1 << realgps1 << "," << endl;</pre>
156.
157.
                }
                namedWindow("Marker Detector", WINDOW NORMAL);
158.
159.
                imshow("Marker Detector", frameCopy);
160.
                if (waitKey(30) >= 0) break;
161.
162.
163.
        }
```

References

[1]S. E. Hmrawey, A. Maaty, A. I. Elmohr, Durability of Hot Asphalt Mixtures Containing Reclaimed Asphalt Pave- ments, OALib 02 (2015) 1–18. doi:10.4236/oalib.1101508.

[2]A. Vasenev, F. Bijleveld, T. Hartmann, A. G. Doree, et al., Visualization workflow and its implementation at asphalt paving construction site, in: M. Bourdeau, J. Beetz (Eds.), Proceedings of the CIB W78-W102 2011: international conference, computer knowledge building, 26-28 October 2011, Sophia Antipolis, France, CIB, 2011, pp. 1–10.

[3]S. R. Miller, Hot mix asphalt construction, towards a more professional approach (12 2010).

[4]NCAT (1991). Hot Mix Asphalt Materials, Mixture Design and Construction. NAPA Research and Education Foundation.

[5]J. Mahoney, S. Muench, L. Pierce, S. Read, H. Jakob, R. Moore, et al., Construction-Related Temperature Differ- entials in Asphalt Concrete Pavement: Identification and Assessment, Transportation Research Record 1712 (2000) 93–100. doi:10.3141/1712-12.

[6]K. A. Willoughby, J. P. Mahoney, L. M. Pierce, J. S. Uhlmeyer, K. W. Anderson, et al., Construction-Related Asphalt Concrete Pavement Temperature and Density Differentials, Transportation Research Record 1813 (1) (2002) 68–76. doi:10.3141/1813-09.

[7]H. L. ter Huerne, S. R. Miller, A. G. Doree, et al., Monitoring hot mix asphalt temperature to improve homogeneity and pavement quality, in: E. Santagata (Ed.), The sixth international conference on maintenance and rehabilitation of pavements and technological control, 8-10 July 2009, Torino, Italy, politechnico di Torino, 2009, pp. 556–565.

[8]Bijleveld, Frank & Miller, Seirgei & De Bondt, Arian & Dorée, André. (2012). Too hot to handle, too cold to control -influence of compaction temperature on the mechanical properties of asphalt.

[9]Makarov, D. (2017). Developing a real-time process control system for asphalt paving and compaction. Enschede: University of Twente.

[10]C.-C. Li, A. A. Oloufa, H. R. Thomas, et al., A GIS-based system for tracking pavement compaction, Automation in Construction 5 (1996) 51–59. doi:10.1016/0926-5805(95)00019-4.

[11]S. Zhu, X. Li, H. Wang, D. Yu, et al., Development of an Automated Remote Asphalt Paving Quality Control System, Transportation Research Record 2672 (26) (2018) 28–39. doi:10.1177/0361198118758690.

[12]Why you should be using the Trimble SPS850 series GNSS receivers, (Accessed July 15,2019).

URL https://www.positioningsolutions.com/sps85x.php

[13]S. Rathinam, Z. W. Kim, R. Sengupta, et al., Vision-Based Monitoring of Locally Linear Structures Using an Unmanned Aerial Vehicle1, Journal of Infrastructure Systems 14 (1) (2008) 52–63. doi:10.1061/(ASCE)1076-0342(2008)14:1(52).

[14]B. Hudzietz, S. Saripalli, et al., An experimental evaluation of 3D terrain mapping with an autonomous helicopter, ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVIII- 1/C22 (2012). doi:10.5194/isprsarchives-XXXVIII-1-C22-137-2011.

[15]R. Hänsch, C. Wefelscheid, O. Hellwich, et al., Three-Dimensional Building Reconstruction Using Images Obtained By Unmanned Aerial Vehicles, in: ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXVIII-1/C22, 2011. doi:10.5194/isprsarchives-XXXVIII-1-C22-183-2011.

[16]N. Metni, T. Hamel, et al., A UAV for bridge inspection: Visual servoing control law with orientation limits, Automation in Construction 17 (2007) 3–10. doi:10.1016/j.autcon.2006.12.010.

[17]N. Hallermann, G. Morgenthal, V. Rodehorst, et al., Vision-based deformation monitoring of large scale structures using Unmanned Aerial Systems, in: IABSE Symposium Report, Vol. 102, 2014. doi:10.2749/222137814814070343.

[18]C. Zhang, A. Elaksher, et al., An Unmanned Aerial Vehicle-Based Imaging System for 3D Measurement of Unpaved Road Surface Distresses1, Computer-Aided Civil and Infrastructure Engineering 27 (2) (2012) 118–129. doi:10.1111/j.1467-8667.2011.00727.x. [19]S. Rathinam, Z. W. Kim, R. Sengupta, et al., Vision-Based Monitoring of Locally Linear Structures Using an Unmanned Aerial Vehicle1, Journal of Infrastructure Systems 14 (1) (2008) 52–63. doi:10.1061/(ASCE)1076-0342(2008)14:1(52).

[20]Z. Zhang, A flexible new technique for camera calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (11) (2000) 1330–1334. doi:10.1109/34.888718.

[21]G. Bradski, A. Kaehler, Learning OpenCV 3, in: Learning OpenCV 3, O'Reilly Media, Inc, 2016, pp. 645–647.

[22]M. Fiala, ARTag, a fiducial marker system using digital techniques, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 2, 2005. doi:10.1109/CVPR.2005.74.

[23]H. Kato, M. Billinghurst, et al., Marker tracking and HMD calibration for a videobased augmented reality con- ferencing system, in: Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99), 1999, pp. 85–94. doi:10.1109/IWAR.1999.803809.

[24] Pinhole camera model, (Accessed June 15,2019).

URL https://openmvg.readthedocs.io/en/latest/openMVG/cameras/cameras/

[25] OpenCV modules, (Accessed May 1,2019).

URL https://docs.opencv.org/4.1.1/

[26]F. R. Ramirez, R. Muñoz-Salinas, R. Medina-Carnicer, et al., Speeded Up Detection of Squared Fiducial Markers, Image and Vision Computing 76 (2018). doi:10.1016/j.imavis.2018.05.004.

[27]A. M. Singh, Q. P. Ha, D. K. Wood, M. Bishop, et al., Low-latency Vision-based Fiducial Detection and Localization for Object Tracking, 2017.

[28]G. Bradski, A. Kaehler, Learning OpenCV 3, in: Learning OpenCV 3, O'Reilly Media, Inc, 2016, pp. 700–702.

[29]F. Moreno-Noguer, V. Lepetit, P. Fua, Accurate Non-Iterative O(n) Solution to the PnP Problem, IEEE Interna- tional Conference on Computer Vision (01 2007). doi:10.1109/ICCV.2007.4409116